“Dodgeball” Web Application
Security Assessment Findings Report

Date: August 31, 2023
Conducted by: Yujie Yamamoto

Table of Contents

Confidentiality Statement

This document is exclusive property of NCC Group. This document contains confidential information about the web application that is used for interview assessment for web app penetration tester role.

Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment. Due to time limited assessment, the attacker prioritized to identify the critical security issues.

Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization’s attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Scope

Assessment	Details
Web Application Penetration Test	10.0.8.10:5000

Scope Exclusion

As per client request, the attacker did not perform any of the following during penetration test:

- Automated tool for scanning web application vulnerabilities

All the other attacks are permitted by the client.

Executive Summary

Synopsis

On August 31, 2023, the “Dodgeball” website from NCC was assessed through a web app penetration test. All the activities conducted simulates a real penetration test event with the goal of:

- Identifying the vulnerabilities that can be exploited and potentially compromise the website.
- Determining the level of risk and impact of security weaknesses.
- Recommend possible mitigation to prevent such attacks.

All the security issues discovered during the assessment are achieved and verified through manual exploitation.

Findings Overview

During the assessment, there were several critical vulnerabilities that was found in “Dodgeball” website. The attacker was able to gain administrative privilege to the website. This was possible due to the weak password policy of the website and allowing the user and administrator to use default credentials.

Target: Dodgeball – Cross Site Scripting (XSS) was found in the landing page of the website by injecting XSS payload to the “Team Lead” input field. Administrative account was obtained by running brute force attack and using default credentials login. It was also found that the search bar in admin page is susceptible to SQL injection attack.

Recommendations

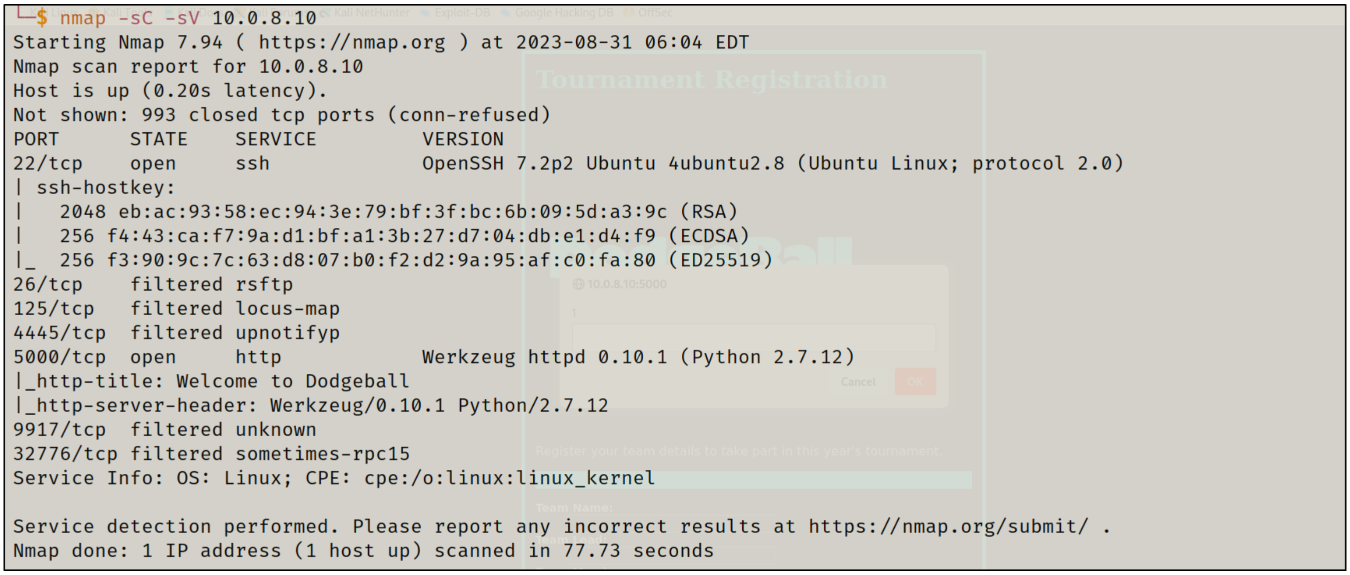
In the Dodgeball web application, there are multiple input field and search bar present. It is recommended to have a proper input sanitization to prevent injection attack. Filtering is a must to filter out common XSS payload and unique characters combination that might lead to injection.

Another one is to implement prepared statements with parameterized queries. It was found that the search bar in admin page is susceptible to SQL injection that might lead to extracting, tampering, and deleting valuable credentials in SQLITE Database.

During testing, weak password policy and allowing default credentials led to the compromise of administrator account. It is recommended to re-evaluate the current password policy of the website and consider a policy of minimum of 8 characters with the combination of number, capital letter, and unique symbols. It is also recommended to have password blacklisting and rate limiting to prevent brute force attack in any sort of login pages.

Enumeration

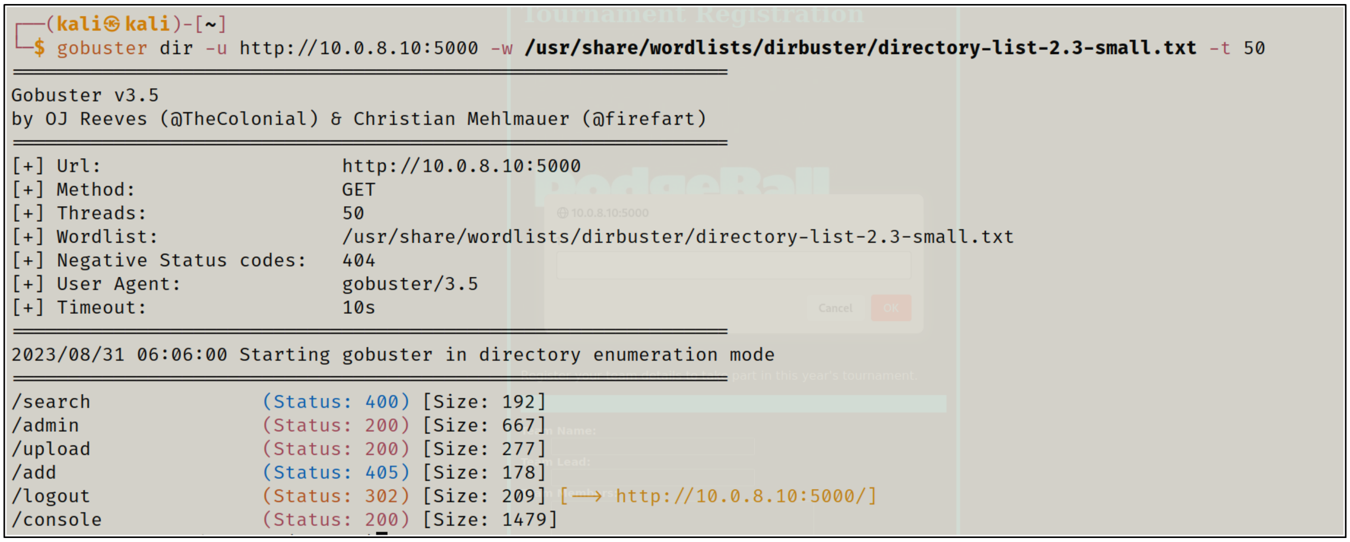
The attacker performed port scanning and service enumeration to gather information about the services running in Dodgeball web application. The initial Nmap scan discovered that TCP port 5000 is where the website is running.



```
└─$ nmap -sC -sV 10.0.8.10
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-31 06:04 EDT
Nmap scan report for 10.0.8.10
Host is up (0.20s latency).
Not shown: 993 closed tcp ports (conn-refused)
PORT      STATE    SERVICE         VERSION
22/tcp    open     ssh             OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 eb:ac:93:58:ec:94:3e:79:bf:3f:bc:6b:09:5d:a3:9c (RSA)
|   256 f4:43:ca:f7:9a:d1:bf:a1:3b:27:d7:04:db:e1:d4:f9 (ECDSA)
|_  256 f3:90:9c:7c:63:d8:07:b0:f2:d2:9a:95:af:c0:fa:80 (ED25519)
26/tcp    filtered rsftp
125/tcp   filtered locus-map
4445/tcp  filtered upnotifyp
5000/tcp  open     http            Werkzeug httpd 0.10.1 (Python 2.7.12)
|_http-title: Welcome to Dodgeball
|_http-server-header: Werkzeug/0.10.1 Python/2.7.12
9917/tcp  filtered unknown
32776/tcp filtered sometimes-rpc15
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 77.73 seconds
```

Figure 1. Nmap Scan Result

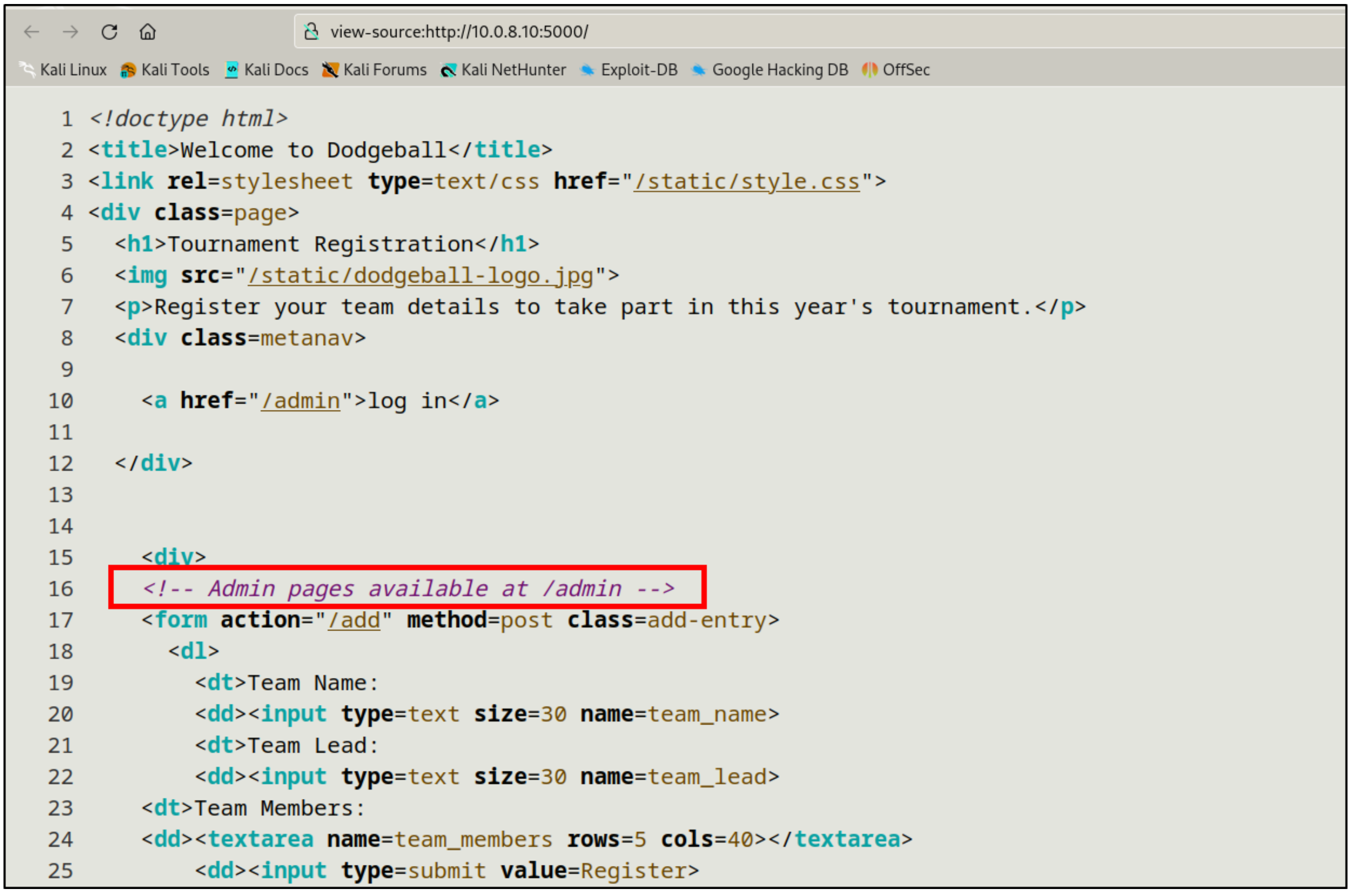
For further enumeration, directory busting of http://10.0.8.10:5000/ using **GoBuster** was achieved. It was revealed that there are directories that have critical security issues such as **/admin** page.



```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://10.0.8.10:5000 -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -t 50
===============================================================
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.0.8.10:5000
[+] Method:                  GET
[+] Threads:                 50
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.5
[+] Timeout:                 10s
===============================================================
2023/08/31 06:06:00 Starting gobuster in directory enumeration mode
===============================================================
/search               (Status: 400) [Size: 192]
/admin                (Status: 200) [Size: 667]
/upload               (Status: 200) [Size: 277]
/add                  (Status: 405) [Size: 178]
/logout               (Status: 302) [Size: 209] [--> http://10.0.8.10:5000/]
/console              (Status: 200) [Size: 1479]
```

Figure 2. Directory Busting

Viewing the page source code reveal that there are comments that disclose information about the directory of admin page.



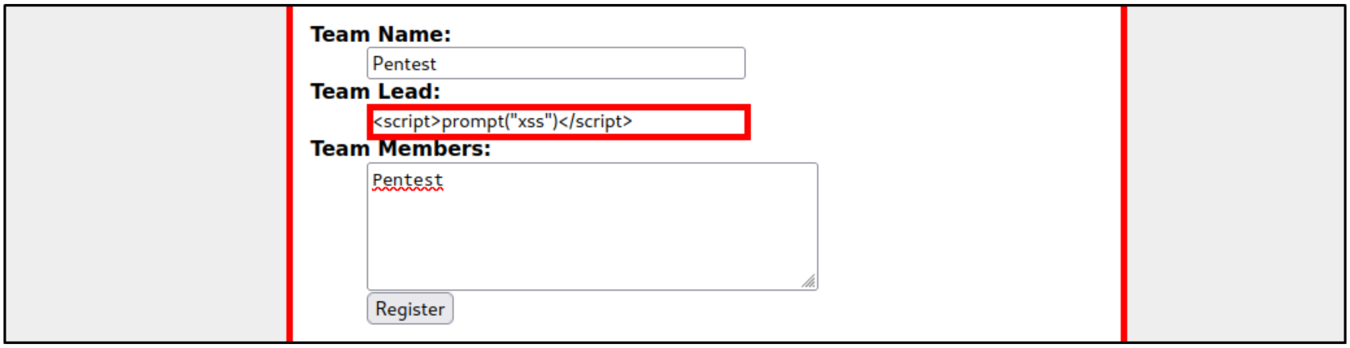
```
<!doctype html>
<title>Welcome to Dodgeball</title>
<link rel=stylesheet type=text/css href="/static/style.css">
<div class=page>
  <h1>Tournament Registration</h1>
  <img src="/static/dodgeball-logo.jpg">
  <p>Register your team details to take part in this year's tournament.</p>
  <div class=metanav>

    <a href="/admin">log in</a>

  </div>


    <div>
    <!-- Admin pages available at /admin -->
    <form action="/add" method=post class=add-entry>
      <dl>
        <dt>Team Name:
        <dd><input type=text size=30 name=team_name>
        <dt>Team Lead:
        <dd><input type=text size=30 name=team_lead>
    <dt>Team Members:
    <dd><textarea name=team_members rows=5 cols=40></textarea>
        <dd><input type=submit value=Register>
```

Figure 3. View page source in homepage

Exploitation

Stored Cross-Site Scripting (XSS)

The homepage of Dodgeball is composed of multiple input field for registration purposes. It was tested to observe if it is vulnerable to injection attack. The **Team Lead input field** was found to be vulnerable to Cross-Site Scripting.



Team Name: Pentest

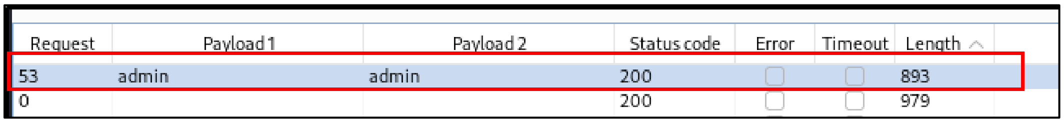
Team Lead: <script>prompt("xss")</script>

Team Members: Pentest

Figure 4. Cross-Site Scripting in Team Lead input field

Brute Force Attack

The **/admin page** was found to be vulnerable to brute force attack as the website is not rate limiting the login attempt. It was also found that admin account is using default and common credentials which are identified using Burp Suite.

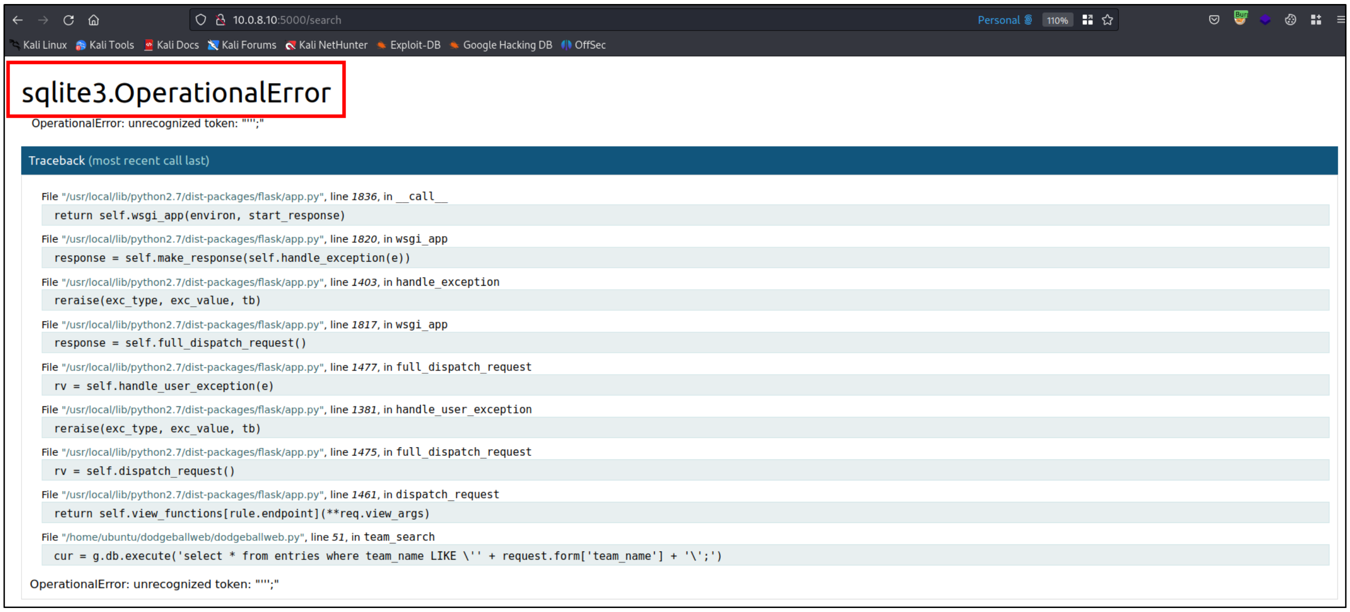


Request	Payload 1	Payload 2	Status code	Error	Timeout	Length
53	admin	admin	200	□	□	893
0			200	□	□	979

Figure 5. Brute Force attack using Burp Suite

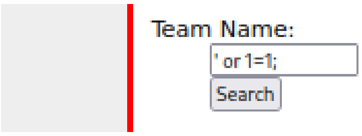
SQL Injection

In the admin page, it was found that the Team Name search bar is vulnerable to SQL Injection. After giving some input apostrophe (‘) which is a basic SQL statement, it throw some error and redirect to a page disclosing that the SQLITE3 is the type of database that is running in the website.



```
sqlite3.OperationalError
OperationalError: unrecognized token: "'";"

Traceback (most recent call last)

  File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1836, in __call__
    return self.wsgi_app(environ, start_response)
  File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1820, in wsgi_app
    response = self.make_response(self.handle_exception(e))
  File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1403, in handle_exception
    reraise(exc_type, exc_value, tb)
  File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1817, in wsgi_app
    response = self.full_dispatch_request()
  File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1477, in full_dispatch_request
    rv = self.handle_user_exception(e)
  File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1381, in handle_user_exception
    reraise(exc_type, exc_value, tb)
  File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1475, in full_dispatch_request
    rv = self.dispatch_request()
  File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1461, in dispatch_request
    return self.view_functions[rule.endpoint](**req.view_args)
  File "/home/ubuntu/dodgeballweb/dodgeballweb.py", line 51, in team_search
    cur = g.db.execute('select * from entries where team_name LIKE \'' + request.form['team_name'] + '\';')
OperationalError: unrecognized token: "'";"
```

Figure 6. Sqlite3 information disclosure

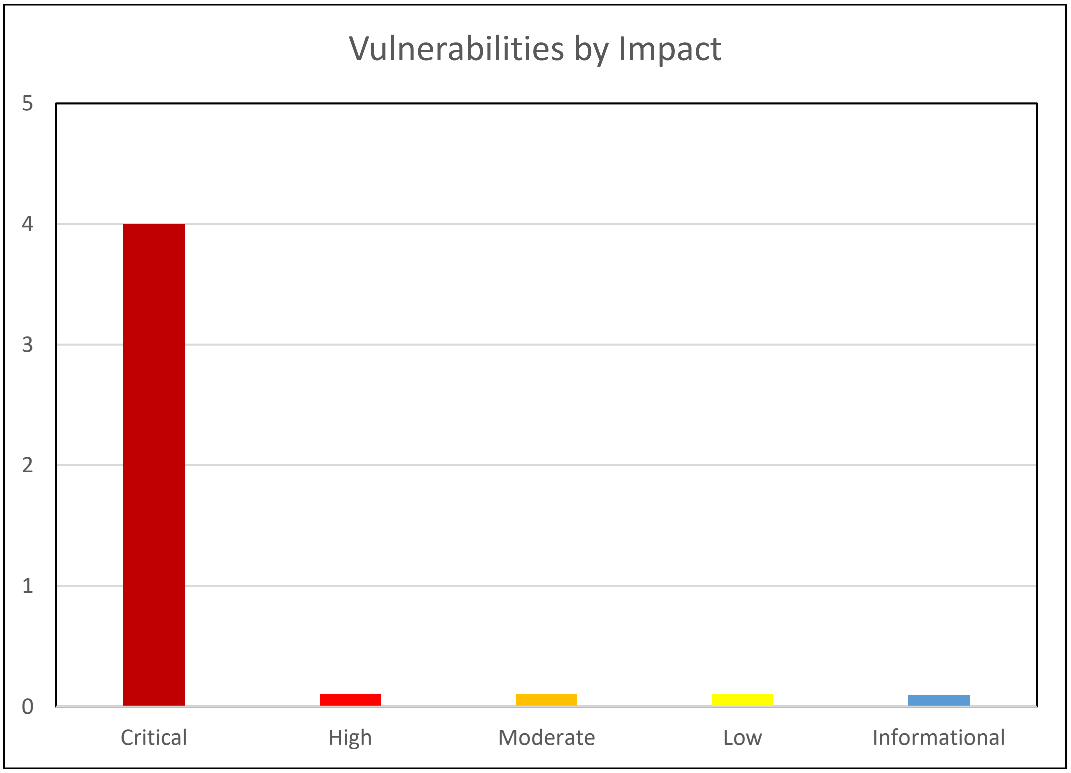


Team Name: ' or 1=1;

Figure 7. Sample SQL Injection Payload

Vulnerabilities by Impact

The following chart illustrates the vulnerabilities found by impact:

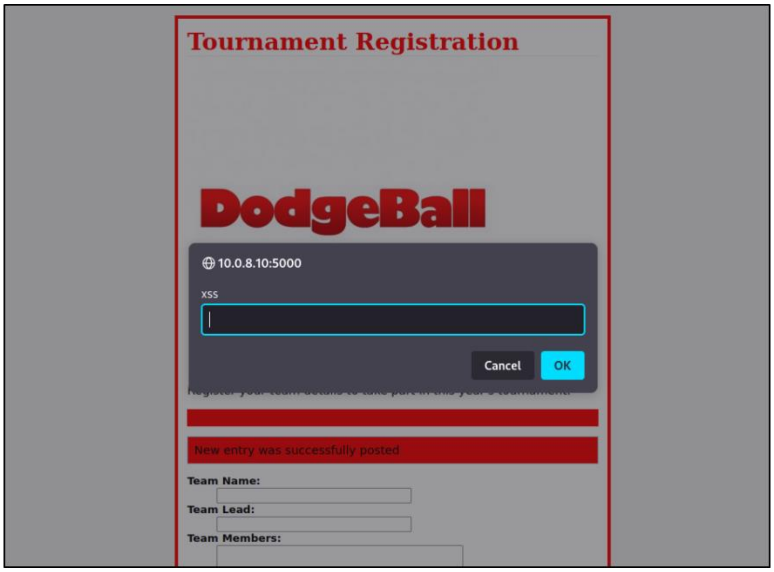


Penetration Test Findings

Finding PT-001: Homepage – Stored Cross-Site Scripting

Description:	"Team Name" Input field in homepage is vulnerable to stored cross site scripting. Common XSS payload <script>prompt(1)</script> was used and it is response is reflected back to the homepage and admin page verifying that it is stored in the SQLITE3 database.
Impact:	**CRITICAL**
Tools:	Manual Testing
Reference:	https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html

Exploitation Proof of Concept



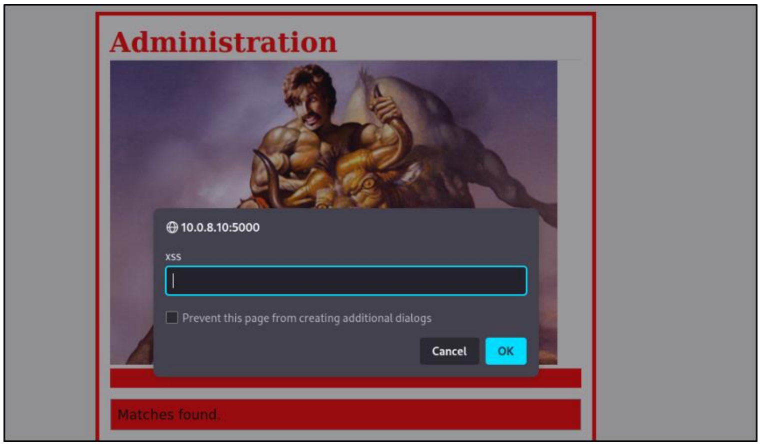
New entry was successfully posted

Team Name:

Team Lead:

Team Members:

Figure 8. XSS prompt in homepage of the website.



☐ Prevent this page from creating additional dialogs

Matches found.

Figure 9 XSS prompt in admin page

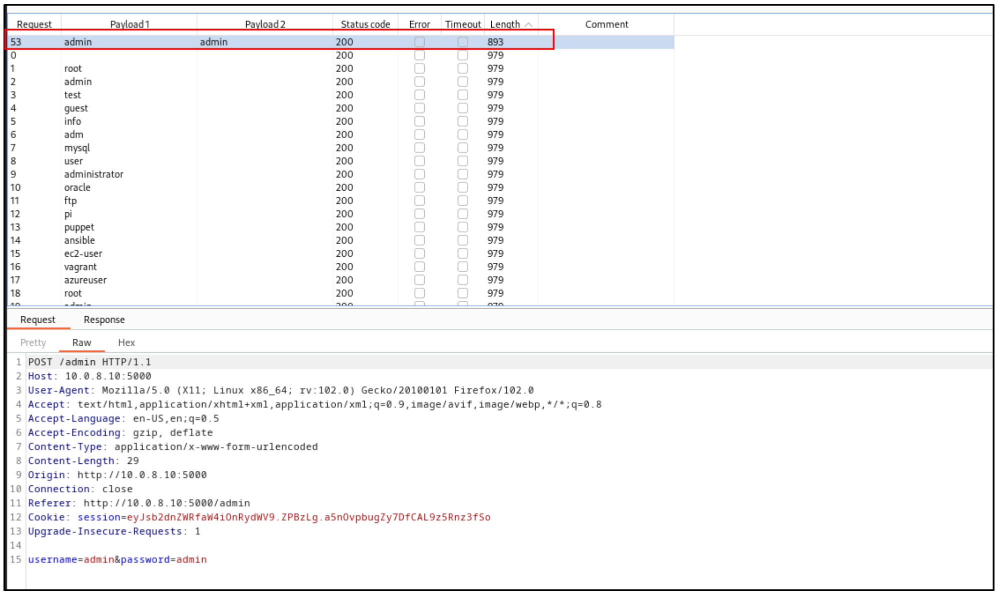
Remediation

Ensure input sanitization. Use Output Encoding to convert untrusted input into a safe form where the input is treated as data and not an executable parameter or code.

Finding PT-002: Admin page – Broken Authentication

Description:	Dodgeball is considered to have a broken authentication vulnerability because of the compromise high privilege user and weak password policy.
Impact:	**CRITICAL**
Tools:	Manual Testing, Burpsuite
Reference:	https://owasp.org/API-Security/editions/2023/en/0xa2-broken-authentication/

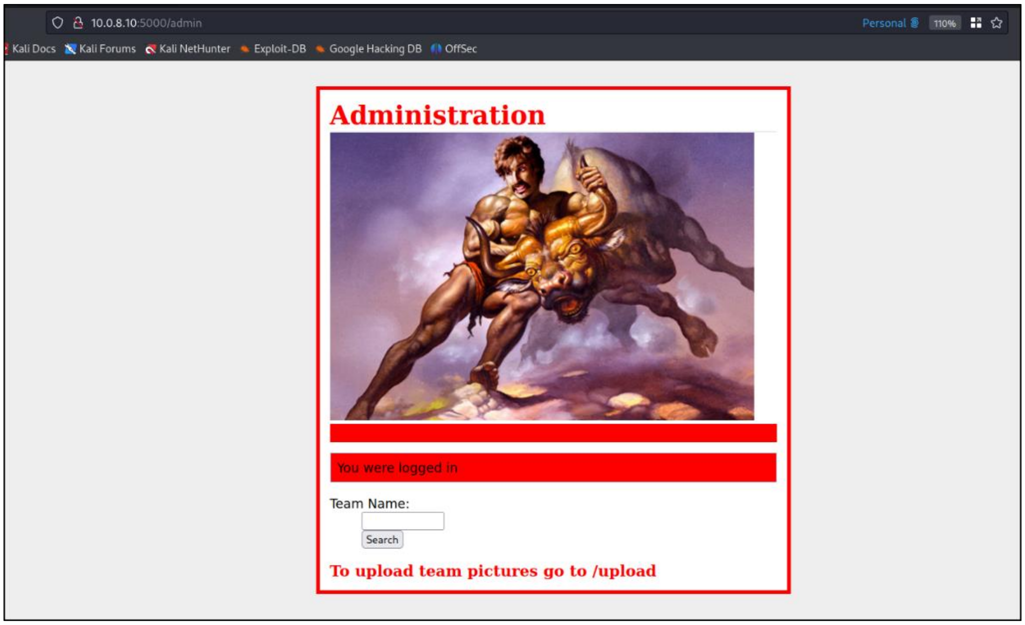
Exploitation Proof of Concept



Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
53	admin	admin	200	☐	☐	893	
0			200	☐	☐	979	
1	root		200	☐	☐	979	
2	admin		200	☐	☐	979	
3	test		200	☐	☐	979	
4	guest		200	☐	☐	979	
5	info		200	☐	☐	979	
6	adm		200	☐	☐	979	
7	mysql		200	☐	☐	979	
8	user		200	☐	☐	979	
9	administrator		200	☐	☐	979	
10	oracle		200	☐	☐	979	
11	ftp		200	☐	☐	979	
12	pi		200	☐	☐	979	
13	puppet		200	☐	☐	979	
14	ansible		200	☐	☐	979	
15	ec2-user		200	☐	☐	979	
16	vagrant		200	☐	☐	979	
17	azureuser		200	☐	☐	979	
18	root		200	☐	☐	979	

```
POST /admin HTTP/1.1
Host: 10.0.8.10:5000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 29
Origin: http://10.0.8.10:5000
Connection: close
Referer: http://10.0.8.10:5000/admin
Cookie: session=eyJsb2dnZWRfaW4iOnRydWV9.ZPBzLg.a5nOvpbugZy7DfCAL9z5Rnz3fSo
Upgrade-Insecure-Requests: 1

username=admin&password=admin
```

Figure 10. Brute Force Attack using Burp Suite



Administration

You were logged in

Team Name:

To upload team pictures go to /upload

Figure 11. Compromise Admin Page

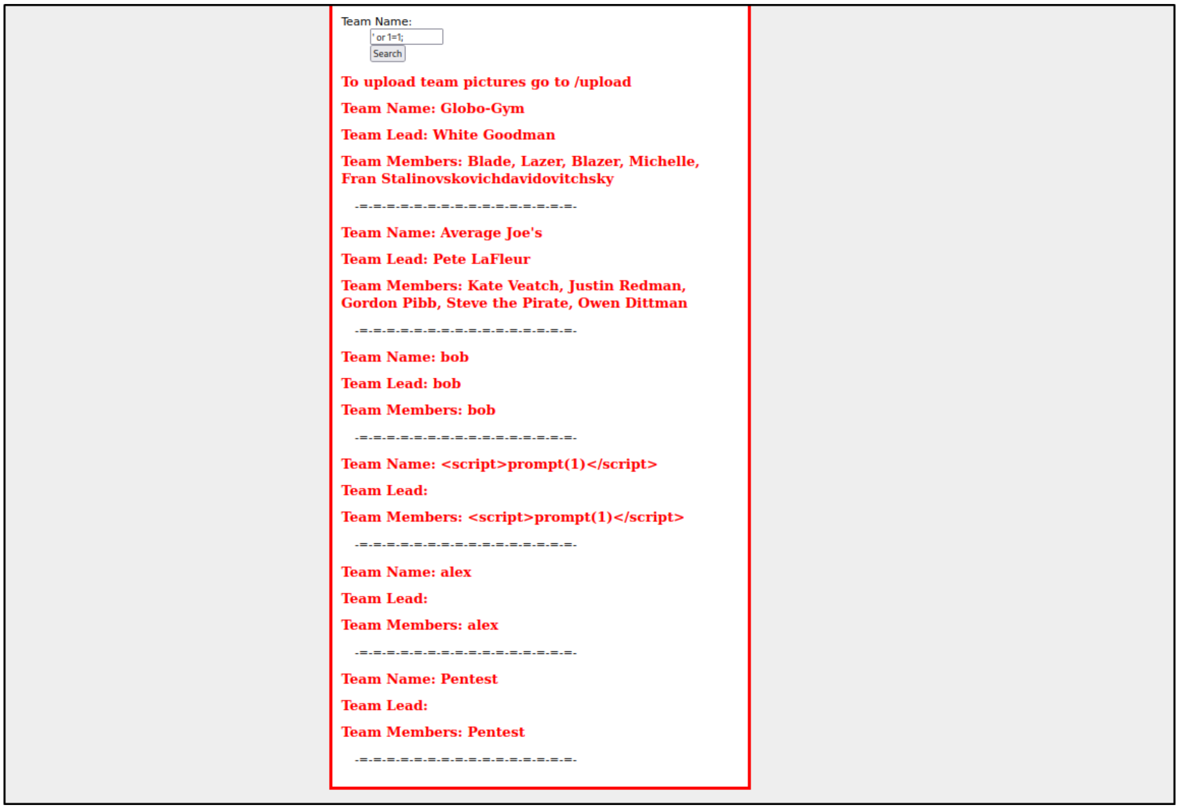
Remediation

Have a strong password policy with combination of number, capitals, and unique character. Require re-authentication for user login such as multi-factor authentication.

Finding PT-003: Admin page – SQL Injection

Description:	The attacker tried giving a malicious SQL statement “**’ or 1=1;**” to “Team Name” search bar in admin page and was found vulnerable to SQL injection. This action results in displaying all the information that was entered in the homepage.
Impact:	**CRITICAL**
Tools:	Manual Testing
Reference:	https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

Exploitation Proof of Concept



```
Team Name:
      ' or 1=1;
      Search

To upload team pictures go to /upload

Team Name: Globo-Gym

Team Lead: White Goodman

Team Members: Blade, Lazer, Blazer, Michelle,
Fran Stalinovskovichdavidovitchsky

   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

Team Name: Average Joe's

Team Lead: Pete LaFleur

Team Members: Kate Veatch, Justin Redman,
Gordon Pibb, Steve the Pirate, Owen Dittman

   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

Team Name: bob

Team Lead: bob

Team Members: bob

   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

Team Name: <script>prompt(1)</script>

Team Lead:

Team Members: <script>prompt(1)</script>

   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

Team Name: alex

Team Lead:

Team Members: alex

   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

Team Name: Pentest

Team Lead:

Team Members: Pentest

   -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

Figure 12. SQL Injection Attack in Admin page

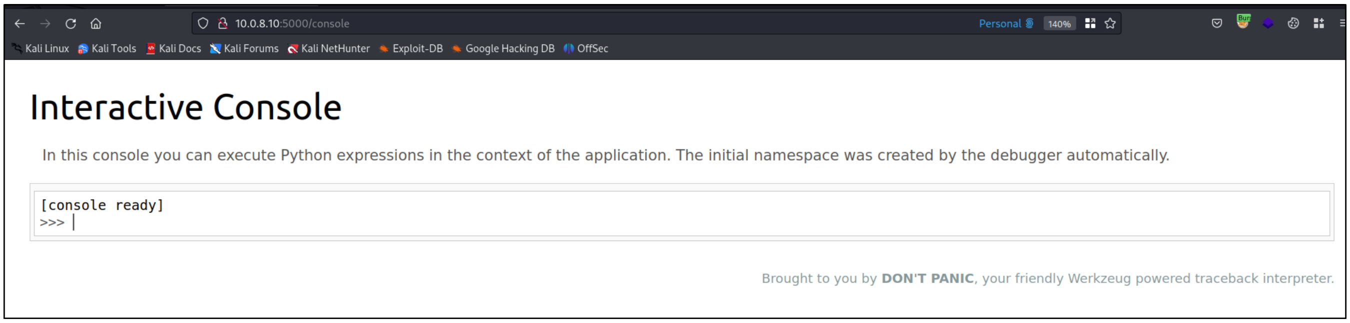
Remediation

Use prepared statements with parameterized queries. This will define all the SQL code, and then pass in each input to the query later. It allows the database to define the user input as data and not a syntax or code.

Finding PT-004: Console Page – Remote Code Execution

Description:	After running GoBuster to look for directories, it was found that **/console** directory has a working python console that is running. Having console in an open directory means anyone can run a script and perform Remote Code Execution
Impact:	**CRITICAL**
Tools:	Manual Testing
Reference:	https://www.crowdstrike.com/cybersecurity-101/remote-code-execution-rce/

Exploitation Proof of Concept



Interactive Console

In this console you can execute Python expressions in the context of the application. The initial namespace was created by the debugger automatically.

```
[console ready]
>>>
```

Brought to you by **DON'T PANIC**, your friendly Werkzeug powered traceback interpreter.

Figure 13. Working Python Console

Remediation

Disable any debugging and development features such as Python Console in web applications. Having those might lead to sensitive information disclosure or remote code execution. If it is necessary, make sure to have least privilege principles for other unauthenticated user to not access such features.