Pen-testing report

Date: [21 December 2023 - 10:24:21 AM]

From Pen-tester	To Target
[CSTAD]	Foodie shop

Table of Contents

Legal

Confidentiality

This document contains sensitive and confidential information, it should not be shared with any other 3rd parties without written permission.

GDPR

This document may contain personal data subject to the General Data Protection Regulation (GDPR). Handle any personal data in accordance with applicable data protection laws.

Disclaimers

The information provided in this document is for general informational purposes only and should not be construed as professional advice. The author(s) disclaim any liability for damages arising from the use of this information.

Change

The author(s) reserve the right to modify these terms. Review this document periodically for updates.

Contact

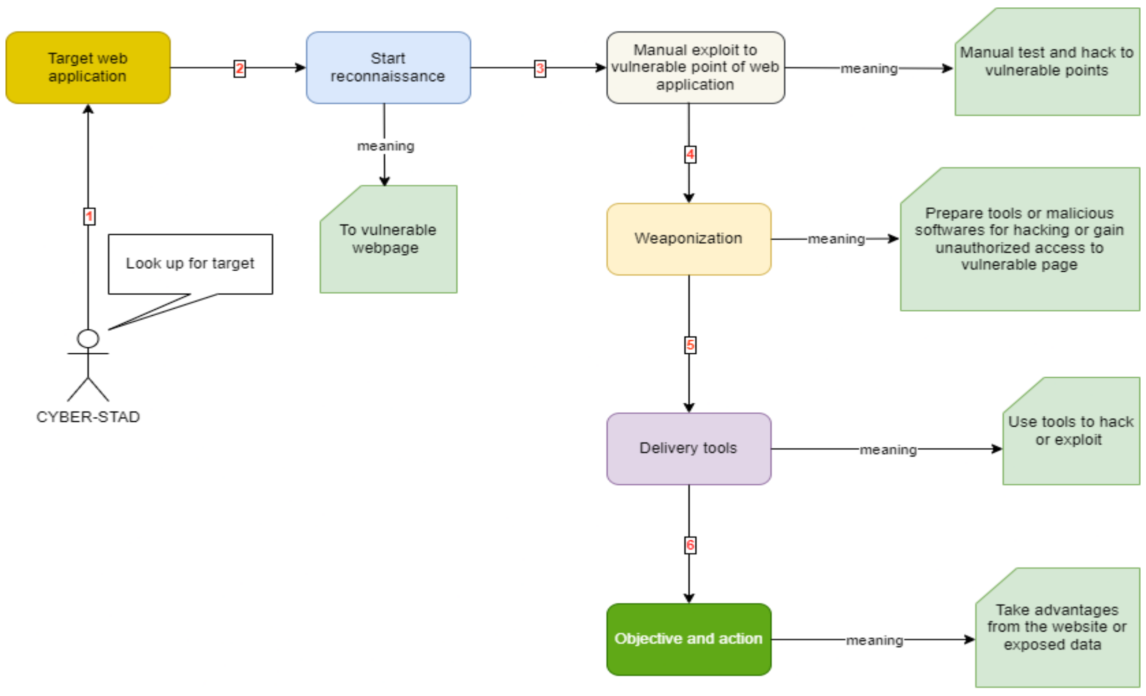
For inquiries, contact:(+855) 875-248-05

Change Log

Date	Version	Comments
1/12/2023	0.1	Initial Report
2/12/2023	0.2	Recon Stage
20/12/2023	0.3	Finalizing Stage

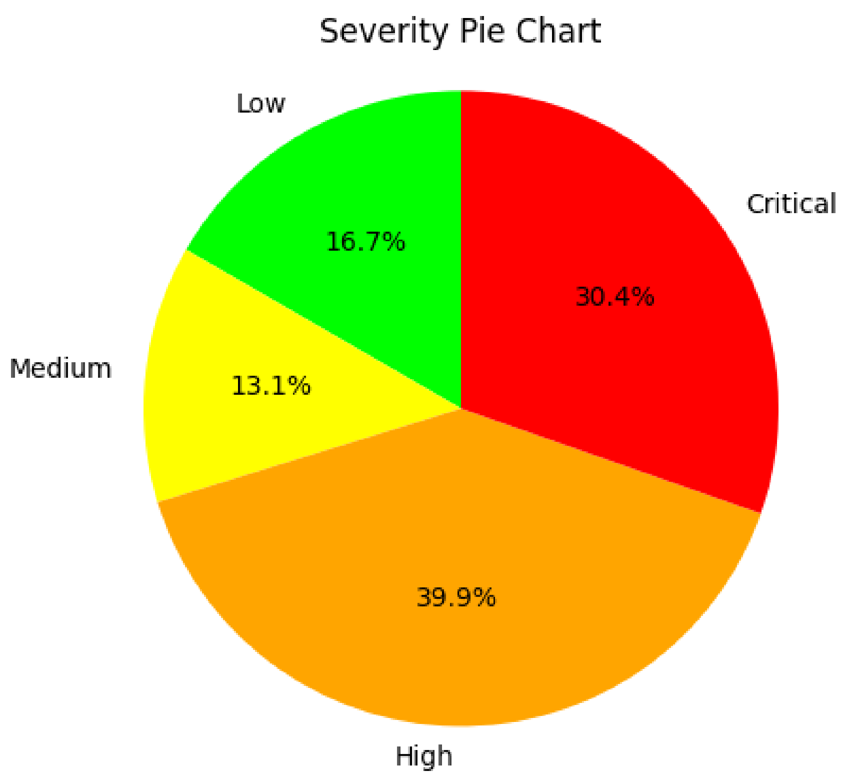
Executive Summary

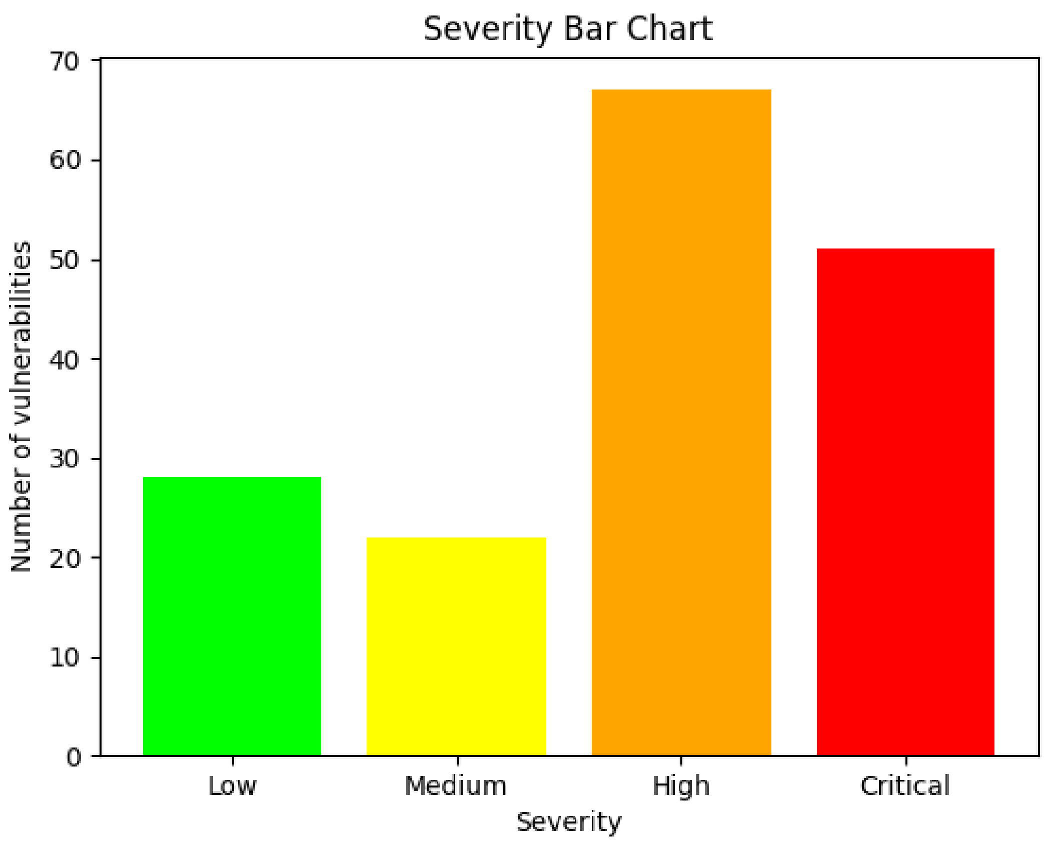
CSTAD engaged CYBER-STAD to conduct a security assessment and penetration testing against a website. The main goal of the engagement was to evaluate the security of the platform and identify possible threats and vulnerabilities. This report details the scope of the engagement, detailed information about all of the findings and some recommendations. The summary below is intended for non-technical audiences to give an idea of the overall results of the engagement and the key findings. The second section of this report is intended for a technical audience as it lists all of our findings in detail, along with reproduction steps, analysis, and recommendations. Based on the security assessment we carried for [platform] and based on our findings, the current risk rating is high. The vulnerabilities discovered can be used by malicious actors to cause breaches and even gain unauthorized access to some management pages. The methodology followed is detailed in the following diagram:


Target web application
2
Start reconnaissance
3
Manual exploit to vulnerable point of web application
meaning
Manual test and hack to vulnerable points
meaning
To vulnerable webpage
1
Look up for target
CYBER-STAD
4
Weaponization
meaning
Prepare tools or malicious softwares for hacking or gain unauthorized access to vulnerable page
5
Delivery tools
meaning
Use tools to hack or exploit
6
Objective and action
meaning
Take advantages from the website or exposed data

The following charts summarize the findings grouped by severity of the threat:

Vulnerabilities Breakdown





1 Engagement Summary

1.1 Scope

As requested, the security assessment was only carried out on the following targets:

IP Address: 54.221.11.310
cstad.shop
food.cstad.shop
coffee.cstad.shop
koko.cstad.shop
mama.cstad.shop

1.2 Risk Ratings

The vulnerability risk was calculated based on the Common Vulnerability Scoring System (CVSS v3.0) which is the industry standard for assessing the severity of security vulnerabilities.
The table below gives a key to the risk naming and colours used throughout this report to provide a clear and concise risk scoring system.

Risk	CVSS v3.0 Score	Recommendation
None	0.0	N/A
Low	0.1 - 3.10	Fix at the next update cycle.
Medium	4.0 - 6.10	Fix immediately if there are 0 high risk vulnerabilities.
High	7.0 - 8.10	Fix immediately if there are 0 critical vulnerabilities.
Critical	10.0 - 10.0	Fix immediately.

1.3 Findings Overview

Below is a list of all the issues found during the engagement along with a brief description, its impact and the risk rating associated with it. Please refer to the 'Risk Ratings' section for more information on how this is calculated.

ID	Risk	Description
1	Hard	SQL Injection leading to unauthorized database access.
2	medium	CSRF - Clients can be forced to submit certain non-critical requests.

3	low	PHP version disclosure - Can help develop attacks for this specific version.

2 Technical Details

2.1 SQL Injection CRITICAL ID: 1

We discovered that using specially crafted requests a malicious actor can communicate with the database and query it to retrieve stored data including data stored in the users tables.

URL	https://food.cstad.shop
Parameter	id
References	https://owasp.org/www-community/attacks/SQL_Injection
Request	POST rest/user/login HTTP/1.1 Host: domain.shop Accept: application/json, text/plain, */*
Response	HTTP/1.1 200 OK Content-Type: application/json; charset=utf-8 Vary: Accept-Encoding

[+] How to prevent SQL Injection CRITICAL ID: 1

Preventing injection requires keeping data separate from commands and queries:

1. The preferred option is to use a safe API, which avoids using the interpreter entirely,
provides a parameterized interface, or migrates to Object Relational Mapping Tools (ORMs).
[+] Note: Even when parameterized, stored procedures can still introduce SQL injection if
PL/SQL or T-SQL concatenates queries and data or executes hostile data with EXECUTE
IMMEDIATE or exec().
2. Use positive server-side input validation. This is not a complete defense as many applications require special
characters, such as text areas or APIs for mobile applications.
3. For any residual dynamic queries, escape special characters using the specific escape syntax for that interpreter.
[+] Note: SQL structures such as table names, column names, and so on cannot be escaped, and thus user-supplied
structure names are dangerous. This is a common issue in report-writing software.
4. Use LIMIT and other SQL controls within queries to prevent mass disclosure of records in case of SQL injection.