



Daffodil
International
University

INTERNSHIP

Submitted By

SHAHRIAR SIDDIQUE

181-35-286

Supervised By

MR. MD RAJIB MIA

Lecturer

Department of Software Engineering

Daffodil International University

A document submitted in partial fulfillment of the requirement for the
degree of Bachelor of Science in Software Engineering

Department of Software Engineering
DAFFODIL INTERNATIONAL UNIVERSITY

Spring – 2022

APPROVAL

This Internship titled on "**Internship**", submitted by **Shahriar Siddique** (ID: **181-35-286**) to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS



Dr. Imran Mahmud
Head and Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Chairman



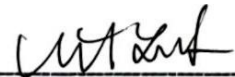
Kaushik Sarker
Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1



Dr. Md. Fazla Elahe
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2




Mohammad Abu Yousuf, PhD.
Professor
Institute of Information Technology
Jahangirnagar University

External Examiner

DECLARATION

I am **Shahriar Siddique**, ID: 181-35-286 student of Daffodil International University. I hereby declare that I have successfully completed my internship at **BugsBD Limited**, where **Mr. Md. Rajib Mia**, Department of Software Engineering, appointed as my supervisor. Additionally, I am stating that I did not prepare or submit this report prior for any other reason, incentive, or presentation by someone other than myself. Also mentioned is the absence of any plagiarism or data manipulation in the materials consulted for this report.



Student Name: Shahriar Siddique

ID: 181-35-286

Batch: 25th

Department of Software Engineering

Daffodil International University

Certified By:



Supervised By:

Mr. Md Rajib Mia

Lecturer

Department of Software Engineering

Daffodil International University

ACKNOWLEDGEMENT

I would first like to express my thankfulness to Almighty Allah for his kindness in granting me the opportunity to submit my internship report on time. I would like to express my sincere gratitude to the Faculty of Science and Information Technology for maintaining the internship credit in the graduation academic program and giving me the chance to work in industry in my area of competence.

I want to thank Mr. Md Rajib Mia from the Department of Software Engineering, who is my supervisor. I owe him a great debt of gratitude for the wise, helpful advice, direction, and inspiration he gave me. I want to express my sincere gratitude to Dr. Imran Mahmud, professor and department head of software engineering, for his unceasing support. I'd like to express my gratitude to everyone who helped me with my internship by offering insightful advice. I am really happy and proud to express my appreciation and deep admiration to the esteemed teachers of the Department of Software Engineering for giving me this chance.

Finally, I want to express my gratitude to my parents for always being an inspiration to me. Without their assistance, perhaps I wouldn't be here.

Table of Contents

| | |
|---|-----|
| APPROVAL | i |
| DECLARATION | ii |
| ACKNOWLEDGEMENT | iii |
| EXECUTIVE SUMMARY | x |
| CHAPTER 1: INTRODUCTION | 1 |
| 1.1 BACKGROUND | 1 |
| 1.2 MOTIVATION | 1 |
| 1.3 OBJECTIVES | 2 |
| 1.4 SCOPE | 2 |
| CHAPTER 2: COMPANY OVERVIEW | 2 |
| 2.1.1 Mission And Vision | 3 |
| 2.1.2 History | 3 |
| 2.1.3 Location | 4 |
| 2.1.4 Management | 5 |
| 2.1.5. Partner | 5 |
| 2.2 ORGAN-GRAM | 6 |
| 2.3 SERVICES | 6 |
| 2.3.1 Penetration Testing | 6 |
| 2.3.2 Mobile App & Device Security | 7 |
| 2.3.3 Network Pentesting | 7 |
| 2.3.3 Vulnerability Assessment | 7 |
| 2.3.4 Web Server Pentesting | 7 |
| 2.3.5 Source Code Audit | 8 |
| 2.3.6 Web Application Pentesting | 8 |
| 2.3.7 Mobile Application Pentesting | 8 |
| 2.4 CLIENTS | 9 |

| | |
|---|----|
| CHAPTER 3: COMPANY CULTURE AND CARRYING OUT | 9 |
| 3.1 DEPARTMENT/SECTION OVERVIEW | 9 |
| 3.2 WORKING TEAM (NAME OF YOUR WORKING TEAM) | 10 |
| 3.3 WORKING ENVIRONMENTS & PROTOCOLS | 10 |
| 3.3.1 RULES & REGULATIONS | 10 |
| 3.3.2 Motto of the Organization..... | 10 |
| 3.3.3 HANDLING CLIENTS | 10 |
| 3.3.4 Facilities | 10 |
| 3.4 INTERNEE LIFE CYCLE | 11 |
| 3.4.1 Getting Started | 11 |
| 3.4.2 Recruiting Policies | 11 |
| 3.5 FIRST DAY AT OFFICE..... | 11 |
| CHAPTER 4: VULNERABILITY ASSESSMENT AND PENETRATION TESTING | 12 |
| 4.1 TECHNICAL SUMMARY | 12 |
| 4.1.1 EXECUTIVE SUMMARY | 12 |
| 4.1.2 SCOPE | 12 |
| 4.1.3 THREAT MODELLING | 13 |
| 4.1.4 RISK RATING | 13 |
| 4.1.5 FINDINGS OVERVIEW | 14 |
| 4.1.6 PORT SCANNING:..... | 14 |
| 4.1.7 SYSTEM INFORMATION:..... | 14 |
| 4.1.8 VULNERABILITIES IN BRIEF:..... | 14 |
| 4.2 TECHNICAL DETAILS | 15 |
| 4.2.1 CROSS SITE SCRIPTING (STORED)..... | 15 |
| Summary:..... | 15 |
| Description:..... | 15 |

| | |
|---|-----------|
| Steps to reproduce:..... | 15 |
| Proof of Concept:..... | 15 |
| Impact: | 16 |
| Remediation: | 16 |
| 4.2.2 CROSS SITE SCRIPTING (REFLECTED) | 16 |
| Summary:..... | 16 |
| Description:..... | 16 |
| Steps to reproduce:..... | 16 |
| Impact: | 17 |
| Remediation: | 17 |
| 4.2.3 EXTERNAL REDIRECT | 18 |
| Summary:..... | 18 |
| Description:..... | 18 |
| Steps to reproduce:..... | 18 |
| Proof of Concept:..... | 19 |
| Remediation: | 19 |
| 4.2.4 REMOTE FILE INCLUSION | 19 |
| Summary:..... | 19 |
| Description:..... | 20 |
| Steps to reproduce:..... | 20 |
| Proof of Concept:..... | 20 |
| Impact: | 20 |
| Remediation: | 21 |
| 4.2.5 REMOTE OS COMMAND INJECTION | 21 |
| Summary:..... | 21 |
| Description:..... | 21 |
| Steps to reproduce:..... | 21 |

| | |
|--|----|
| Proof of Concept: | 22 |
| Impact: | 22 |
| Remediation: | 22 |
| 4.2.6 SQL INJECTION..... | 23 |
| Summary: | 23 |
| Description: | 23 |
| Steps to reproduce:..... | 23 |
| Proof of Concept: | 23 |
| Impact: | 24 |
| Remediation: | 24 |
| 4.2.7 NO RATE LIMIT | 24 |
| SUMMARY:..... | 24 |
| Description: | 24 |
| Steps to reproduce:..... | 24 |
| Proof of Concept: | 25 |
| Impact: | 25 |
| Remediation: | 25 |
| 4.2.8 NO SSL/TLS..... | 25 |
| Summary: | 25 |
| Description: | 25 |
| Impact: | 26 |
| Steps to reproduce:..... | 26 |
| Proof of Concept: | 26 |
| Remediation: | 26 |
| 4.2.9 APPLICATION ERROR DISCLOSURE | 26 |
| Summary: | 26 |
| Description:..... | 26 |

| | |
|---|----|
| Steps to reproduce:..... | 27 |
| Proof of Concept:..... | 27 |
| Impact: | 27 |
| Remediation: | 27 |
| 4.2.10 MISSING ANTI-CLICKJACKING HEADER | 28 |
| Summary:..... | 28 |
| Description:..... | 28 |
| Steps to reproduce:..... | 28 |
| Proof of Concept:..... | 28 |
| Impact: | 28 |
| Remediation: | 29 |
| 4.2.11 VULNERABLE JS LIBRARY | 29 |
| Summary:..... | 29 |
| Description:..... | 29 |
| Steps to reproduce:..... | 29 |
| Proof of Concept:..... | 30 |
| Impact: | 30 |
| Remediation: | 30 |
| 4.2.12 X-FRAME OPTIONS HEADER NOT SET | 30 |
| Summary:..... | 30 |
| Description:..... | 30 |
| Steps to reproduce:..... | 30 |
| Proof of Concept:..... | 31 |
| Impact: | 31 |
| Remediation: | 31 |
| 4.2.13 ABSENCE OF ANTI-CSRF TOKEN | 31 |
| Summary:..... | 31 |

| | |
|--|----|
| Description:..... | 31 |
| Steps to reproduce:..... | 32 |
| Proof of Concept:..... | 32 |
| Impact: | 32 |
| Remediation:..... | 32 |
| 4.2.14 COOKIE NO HTTPONLY FLAG | 33 |
| Summary:..... | 33 |
| Description:..... | 33 |
| Steps to reproduce:..... | 33 |
| Proof of Concept:..... | 34 |
| Impact: | 34 |
| Remediation:..... | 34 |
| 4.2.15 COOKIE WITHOUT SAME SITE ATTRIBUTE..... | 34 |
| Summary:..... | 34 |
| Description:..... | 34 |
| Steps to reproduce:..... | 35 |
| Proof of Concept:..... | 35 |
| Impact: | 35 |
| Remediation:..... | 35 |
| 4.3 PENETRATION TESTING METHODOLOGIES | 35 |
| CHAPTER 5: CONCLUSIONS | 38 |
| Reference | 39 |
| PLAGIARISM REPORT..... | 45 |
| ACCOUNT CLEARANCE | 45 |

EXECUTIVE SUMMARY

The internship report was written in response to the supervisor's recommendation. Mr. Md. Rajib Mia, a lecturer in the department of software engineering at Daffodil International University, requested that it include information on vulnerability assessment and penetration testing at BugsBD Limited: A Look from BugsBD Limited. Studying a system's vulnerability and identifying it are the study's goals. This study was created using both primary and secondary data.

There are five chapters in this report. The aims, methodology, and scope of the study are presented in the first chapter of this study's introduction. The second chapter of the research discusses some theoretical ideas on the Profile of BugsBD Limited in order to help the reader comprehend the Cyber Security Career on which the study is based. The third chapter discusses BugsBD Limited's corporate culture. The fourth chapter discusses several forms of analysis and discoveries about various types of web application vulnerabilities. The entirety of a VAPT report is also detailed in the fourth chapter. The fifth chapter discusses the results that were reached after analyzing the entire investigation.

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

In addition to learning and applying academic ideas, students who participate in internship programs also pick up new abilities that they may use to their undergraduate bachelor's degree programs. An internship gives students the chance to observe and learn from working experts. They would be motivated to learn more about developing their professional skills and establishing future goals.

I have always been interested in software technologies and how they function, as well as the security that supports them, as a student in the Daffodil International University's department of software engineering. My interest in software testing and cyber security especially has been aroused by this. I had the opportunity to work as a vulnerability Assessment Penetration Testing (VAPT) intern at the BugsBD LTD's cyber security branch in order to obtain real-world experience. Based on the knowledge I picked up throughout my internship, I have written this report.

1.2 MOTIVATION

Throughout my internship, I've gained knowledge about how various companies run as well as professional office etiquette. It teaches me how to effectively manage my time and the task at hand. I have to get knowledgeable about potential security issues and system vulnerabilities as a VAPT intern. It could lead to a system compromise, a data leak unintentionally, or a cyberattack. I needed to learn how software works, where to look for vulnerabilities, how to approach them, how to solve problems, and what elements may be responsible for such issues. VAPT stands for vulnerability assessment and penetration testing.

It's also important to safeguard your system against hackers who aim to ruin it by committing crimes or taking advantage of security holes. This internship ignited my interest in the topic, and the hands-on work experience I gained there enabled me to clarify most of my doubts and get me interested in more studies in the area of cyber security.

1.3 OBJECTIVES

Here is a summary of the main objectives of the internship in VAPT (vulnerability assessment and penetration testing) in cyber security: -

- Discover how to solve issues in the real world.
- to increase my testing and assessment expertise.
- to get work-related experience.
- help boost confidence and enhance communication.
- get knowledge about how to apply new ideas.
- the chance to hear from experts about current issues and potential solutions.

I've attempted to recognize all of my flaws by adhering to these goals and have made every effort to fix them. My ability to produce more effective work has considerably increased as a result of pursuing these goals.

1.4 SCOPE

While describing the information and perspective gained throughout the company's internship program, this internship report's primary goal is to fulfill the requirements of the undergraduate B.Sc. program. I've gained knowledge about my duties as an employee throughout my time here as an intern, including how to complete my work swiftly and competently. Additionally, develop the ability to control yourself when under pressure at work, give it your all under such conditions, and approach the task in a professional manner. It also showed me that sometimes thinking is necessary while resolving issues at work. Out of the box to solve the problem. It also inspired me to learn about the cyber security more and more. Because it brings new challenge every time. The problem isn't same so I have to think different to solve every problem. It brings me more passionate about cyber security when I feel challenge in this work.

CHAPTER 2: COMPANY OVERVIEW

2.1 ABOUT

Our team currently includes several brilliant, youthful, and knowledgeable software engineers and security experts that have previously demonstrated their abilities in this field. We also have a solid network of people with a range of skill sets to enable us provide our valued customer with the finest solution available. This acts as a venue for the deliberate transmission of information that enhances security competence as well as a public platform for entrepreneurs and clients.

2.1.1 MISSION AND VISION

Mission: The group helps organizations, governments, and countries protect them from cybercrime, reduce their risk in a connected world, follow rules, and modernize their operations. In addition to having a big influence on staff monitoring, data security, and user behavior analytics, we wish to become a global leader in these areas. Our goal to offer a secure online environment is built on our cutting-edge goods and reliable service. The objective is to establish a reputation as a leader in the ICT sector for reliable, high-quality solutions and services. Our goal is to improve the business operations of our clients by developing and/or implementing top-notch IT products and services. Our goal is to help our clients expand their companies. We want to go above and beyond your expectations so that we can establish a long-lasting, mutually beneficial partnership.

Vision: The vulnerabilities of the digital environment are reduced, and security is enhanced, through the implementation of integrated cyber-security and cyber-defense systems that address contemporary threats. Our primary priority is strengthening our relationships with our customers. By providing top-notch security products and solutions, we go above and beyond what our customers expect from us. We provide first-rate training services and employ top talent to help any organization maintain high levels of workmanship. We strive to be recognized as a technical innovation on a global scale by addressing cyber security challenges.

2.1.2 HISTORY

BugsBD Limited's focus is to enhance enterprises' ability to construct complex business systems. We create, market, and support enterprise software. Global system operational teams, developers, and architects The company was

established in 2015. the previous few years in terms of income and staff, the firm has developed fast and strategically. We presently have more than 35 staff. We are completely distributed, much like our program, which is one of our most distinguishing features. We served clients with both domestic and international services. We are spread across numerous time zones and six different countries, and we work when it is most convenient for us. the preceding six years.

Our products help businesses optimize their operations, manage their costs, and invest in innovation. The solutions are appropriate for the organizations' requirements. They simplify current technology problems. We've coupled our intelligence with modernization technologies. This combo is useful against cyber-attacks. As we progress through the modernization cycle, our services will be the most useful cyber protection platform. Our company's major goal is to make our customers happy.

2.1.3 LOCATION

The Location of BugsBD Limited is Shyamoli, Dhaka, Bangladesh. Full Address below here:

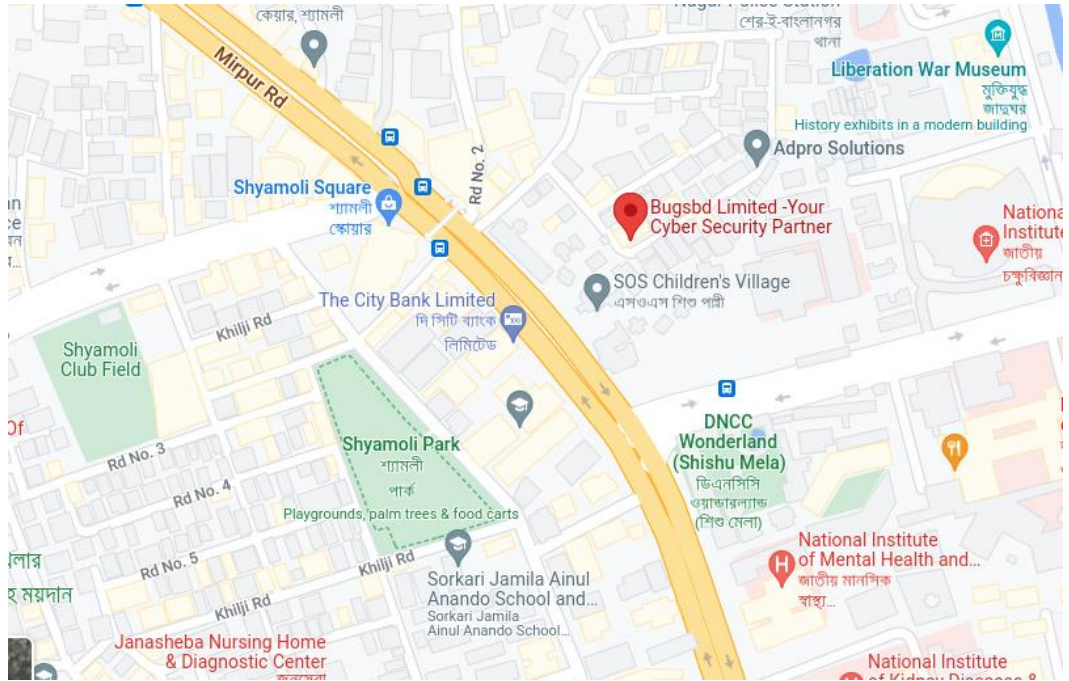
Address: - 1/C (5th Floor) Road #1, Shyamoli.

Contacts: +8801761616261

Email: info@bugsgbd.com / hr@bugsgbd.com

Official website: <https://bugsgbd.com>

<https://g.page/cybersecurityservicesandsolution?share>



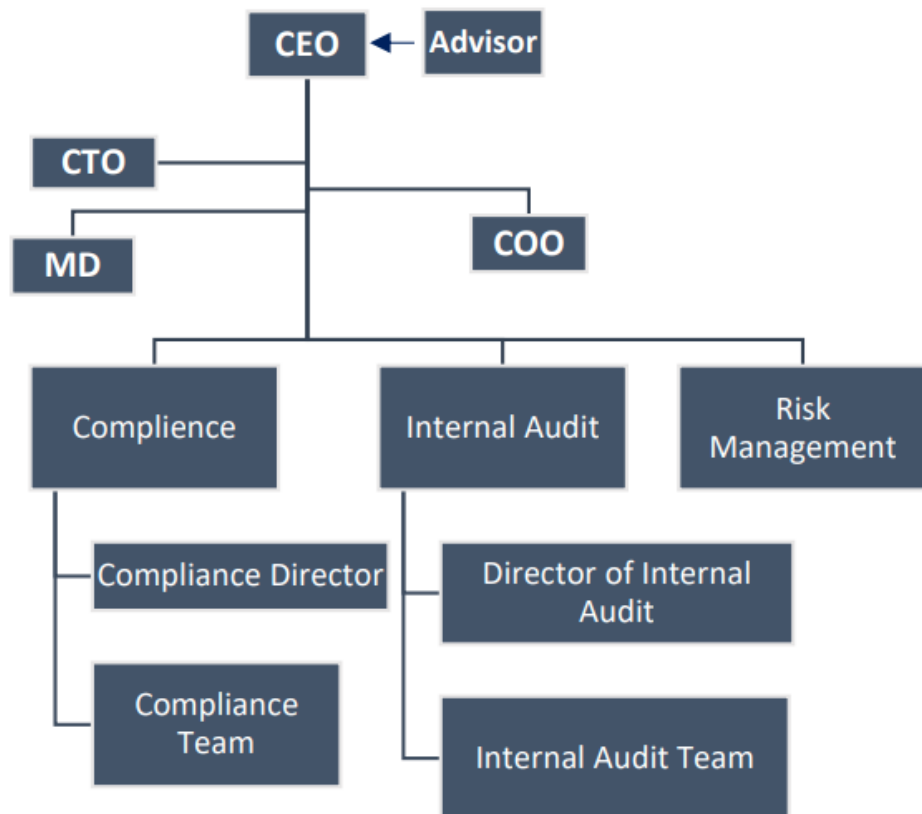
2.1.4 MANAGEMENT

BugsBD Limited has Managing Director then come's Chief Executive Officer. Under CEO Comes Chief Technology Officer then Chief Operation Officer then Others Department.

2.1.5. PARTNER

BugsBD Limited has no organization partner. BugsBD has BASIS Membership.

2.2 ORGAN-GRAM



We allocate dedicated teams to each of our projects, with distinct profiles based on the needs.

2.3 SERVICES

2.3.1 PENETRATION TESTING

Penetration testing evaluates if unauthorized access or other destructive behavior is possible given any system's present flaws. It identifies and ranks security hazards. A rigorous penetration testing technique is a good defense against any flaws in any system.

- immediately determine the attack surface of a target organization
- Bypass network restrictions and conduct a remote IP address scan.
- Creating Proof of concept(POC) to provide vulnerability risk.

2.3.2 MOBILE APP & DEVICE SECURITY

Conducting security assessments of mobile apps that operate on iOS and Android platforms is our area of expertise. These operating systems are used by the vast majority of mobile devices, including tablets and smartphones. We utilize these devices now, and their use in business has substantially increased. We utilize them for mobile banking, mobile payment, and financial services. In reality, an entire industry has been created as a result of the app economy. Due to their widespread usage, hackers now have a significant chance to access these devices and present a serious threat.

2.3.3 NETWORK PENTESTING

We find exploitable flaws in networks, systems, hosts, and network devices via network penetration testing (ie: routers, switches). Hackers can use these flaws to take advantage of the network. Hackers have the chance to infiltrate networks and systems thanks to network penetration testing. They can gain access to private information in this way. System for preventing intrusions (IPS) For the purpose of preventing sophisticated malware attacks and zero-day vulnerabilities, IPS collect network traffic and analyse behaviour.

- Identify any network security weaknesses that exist.
- Determine the degree of risk to which your business is exposed.
- Correct any lingering network security issues.

2.3.3 VULNERABILITY ASSESSMENT

Software flaws that might be utilized to cause harm or lapses in internal security protocols are referred to as software vulnerabilities. Vulnerability assessment is the process of detecting, classifying, and ranking vulnerabilities in a system. We are aware of and understand susceptibility threats, and we act appropriately.

2.3.4 WEB SERVER PENTESTING

We assess your company's resistance to numerous simulated online social engineering attacks. The three major areas of attention for web server pentesting

are identity, analysis, and reporting vulnerabilities, including those pertaining to authentication, configuration, and protocol linkages.

- Examine web server directories to glean vital details about site features, login screens, etc.

2.3.5 SOURCE CODE AUDIT

The foundation of source code audit is a combined investigation of coding convention breaches, bug identification, and security flaws. It makes sure the application is free of errors before release. Specialists manually inspect the source code during the source code audit to find security problems. Our experts painstakingly concentrate on every significant component and complete the assessment. Through their efforts, both high-risk and low-risk vulnerabilities are addressed.

2.3.6 WEB APPLICATION PENTESTING

Our skilled website penetration testers will review every component of your online application to help you identify security problems. As a consequence, a secure software development lifecycle is advanced and organizational risks are better detected and given higher priority.

- Look for security gaps and insufficient security measures.
- Utilize security flaws in web applications.
- Make the insecure features of your program obvious.
- Make security design issues a priority and catch them early.

2.3.7 MOBILE APPLICATION PENTESTING

Smartphones are becoming a highly in demand platform for the development of business apps across several industries. These mobile applications provide several advantages over conventional online services, including portability and simplicity of usage. New attack vectors and threats to mobile devices, on the other hand, are usually neglected. Due to this new danger situation, BugsBD offers specialized mobile application penetration test services.

2.4 CLIENTS

Client of BugsBD Limited are:



CHAPTER 3: COMPANY CULTURE AND CARRYING OUT

3.1 DEPARTMENT/SECTION OVERVIEW

BugsBD Limited has Only One Department which is Cyber Security Department. But in Cyber Security Department there are many sections present. Like Vulnerability

Assessment, Penetration Testing Section, Red Team Assessments, Mobile Security, Source Code Audit, Network Security.

3.2 WORKING TEAM (NAME OF YOUR WORKING TEAM)

The assignments that the corporation gives to our group or team. Where we collaborate on duties provided by the organization depends on how significant those jobs are, which is typically the case. The assignment must be finished by the deadline since it is imperative that we do so. They can evaluate both our degree of individual effort and our ability to work as a team by doing this.

3.3 WORKING ENVIRONMENTS & PROTOCOLS

The workplace is essential for building a successful career. The working environment here is excellent. My employees are all nice and friendly. Everyone has their own personal area. In addition, the necessities are provided by our company.

3.3.1 RULES & REGULATIONS

Each employee has been given responsibilities to do. The company includes a timer with each job it gives out. Thus, they must finish and turn in the report by the due date. There are, nevertheless, some rules that we must follow:

- Personal projects or forbidden topics should not be discussed during working hours.
- Cannot interfere with a coworker's equipment or carry out any illegal conduct without his consent.
- if you require a vacation. You must tell your team's leader.
- If you require a leave of absence, you must notify Human Resources in writing.
- cannot physically damage any technology that the company has provided. In that case, you must alert the relevant authority.

3.3.2 MOTTO OF THE ORGANIZATION

Motto of BugsBD Limited is “Your Cyber Security Partner”.

3.3.3 HANDLING CLIENTS

BugsBD Authority is responsible to Handling Client.

3.3.4 FACILITIES

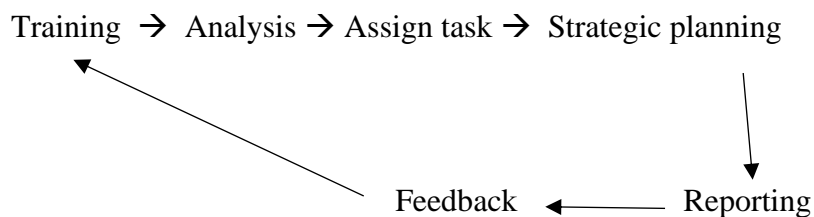
We Got Several facilities from BugsBD Limited. Like:

- Provide Snacks
- Hybrid Work Facilitates.

3.4 INTERNEE LIFE CYCLE

3.4.1 GETTING STARTED

The life cycle of the intern was calculated using a few variables, and the internship is based on those results.



3.4.2 RECRUITING POLICIES

BugsBD Limited recruit for internship is only two times at a year. They post circular in online job portal, or social media like their official Facebook page, LinkedIn jobs etc. Based on candidate cv and background they shortlist and call for interview. The selection candidate is then selected for internship.

3.5 FIRST DAY AT OFFICE

First day at office was little bit weird. Office personal supported us a lot. Which make me relief and comfortable. At first we meet each other team mate to familiar ourselves. They were very friendly. We Discuss our University Life actives, Our Skill about programming, Cyber security career etc. Some of us were highly professional at other side but in cyber security career they are new. I told them about my programming skill that how good programmer I am, how I participate programming contest and how I achieve those position in programming contest. They were very surprise to hear my programming achievement. After our personal discussion Our office supervisor came and provide guidelines and rules regulations that we should follow. They give us some tasks to complete. We complete our task in time and submit it. That was my first day at office.

CHAPTER 4: VULNERABILITY ASSESSMENT AND PENETRATION TESTING

4.1 TECHNICAL SUMMARY

4.1.1 EXECUTIVE SUMMARY

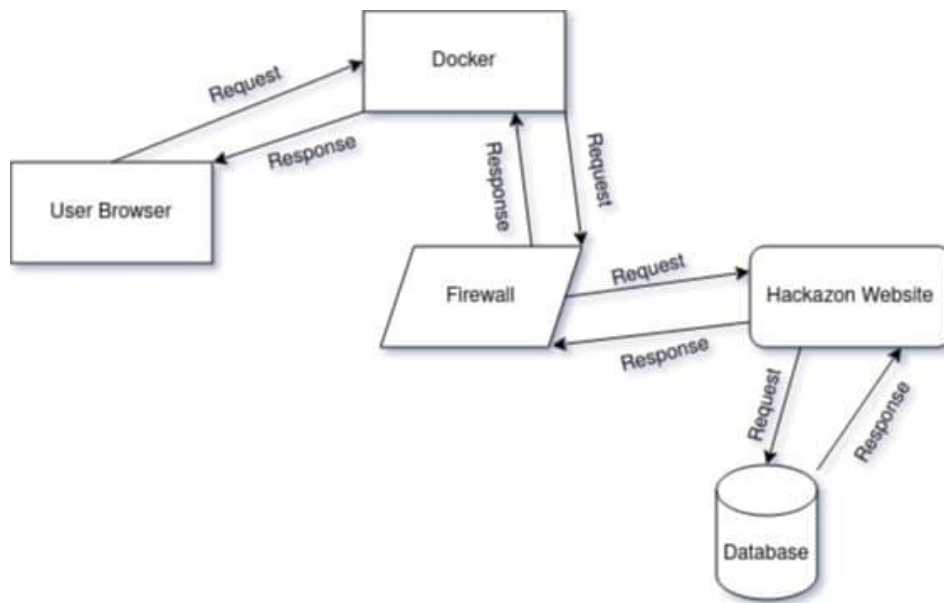
xyz is an web application project that incorporates a e-commerce web applications. It is built with php with Laravel framework technologies that used in modern web applications including AJAX and RESTful API's.

4.1.2 SCOPE

The scope of the web application testing of the engagement located at <http://10.6.6.16:80/>. The web application is developed by modern technology so finding security issue our main target.

Testing was performed using industry-standard penetration testing tools and frameworks, including: Nmap, DirBuster, Metasploit, Acunetix, Nessus Zap, OpenVAS, and Burp Suite.

4.1.3 THREAT MODELLING



4.1.4 RISK RATING

The table below gives a key to the risk nomenclature and colors used throughout this report in order to provide a clear and straightforward risk grading system.

We must keep in mind that no matter what issue is discovered during any inspection, we cannot fully evaluate the market risk it poses. This means that even though these risks may be significant from a technological perspective, they can be acceptable to humans due to hidden safety measures.

| SL | CVSS Score | v3. Description | Risk Rate |
|----|------------|--|-----------------|
| 1 | 9.0 – 10 | A critical vulnerability is identified and assessed. This needs to be resolved as soon as feasible. | CRITICAL |
| 2 | 7.0 - 8.9 | There was found to be a high-rated vulnerability. This needs to be resolved quickly. | HIGH |
| 3 | 4.0 – 6.9 | There was found to be a medium-level vulnerability. This ought to be fixed as part of routine maintenance. | MEDIUM |
| 4 | 1.0 – 3.9 | There was identified a low-level vulnerability. This has to be taken care of as part of regular maintenance. | LOW |

4.1.5 FINDINGS OVERVIEW

Below is a list of all the concerns discovered throughout the evaluation, along with a brief description and risk score for each issue. The Risk Ratings Section defines the risk ratings that were utilized in this study.

4.1.6 PORT SCANNING:

```
Host is up (0.000082s latency).
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http   Apache httpd 2.4.7 ((Ubuntu))
```

4.1.7 SYSTEM INFORMATION:

```
MAC Address: 02:42:0A:06:06:10 (Unknown)
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5
OS details: Linux 5.3 - 5.4
Uptime guess: 18.950 days (since Sat Jan 29 01:40:27 2022)
```

4.1.8 VULNERABILITIES IN BRIEF:

| SL No. | Description | Risk Type |
|--------|-----------------------------------|-----------|
| 1 | Cross Site Scripting (Stored) | HIGH |
| 2 | Cross Site Scripting (Reflected) | HIGH |
| 3 | External Redirect | HIGH |
| 4 | Remote File Inclusion | HIGH |
| 5 | Remote OS Command Injection | HIGH |
| 6 | SQL Injection | HIGH |
| 7 | No Rate Limit | MEDIUM |
| 8 | SSL/TLS Vulnerability | MEDIUM |
| 9 | Application Error Disclosure | MEDIUM |
| 10 | Missing Anti-clickjacking Header | MEDIUM |
| 11 | Vulnerable JS Library | MEDIUM |
| 12 | X-Frame-Options Header Not Set | MEDIUM |
| 13 | Absence of Anti-CSRF Tokens | LOW |
| 14 | Cookie No HttpOnly Flag | LOW |
| 15 | Cookie without SameSite Attribute | LOW |

4.2 TECHNICAL DETAILS

4.2.1 CROSS SITE SCRIPTING (STORED)

Summary:

| | | |
|------------------|--------------------------------------|-----------------|
| Severity: | HIGH | SL No: 1 |
| Host: | http://10.6.6.16:80 | |
| Path: | /user/register | |
| Method: | _POST() | |
| Payload: | <script>alert("First Name")</script> | |

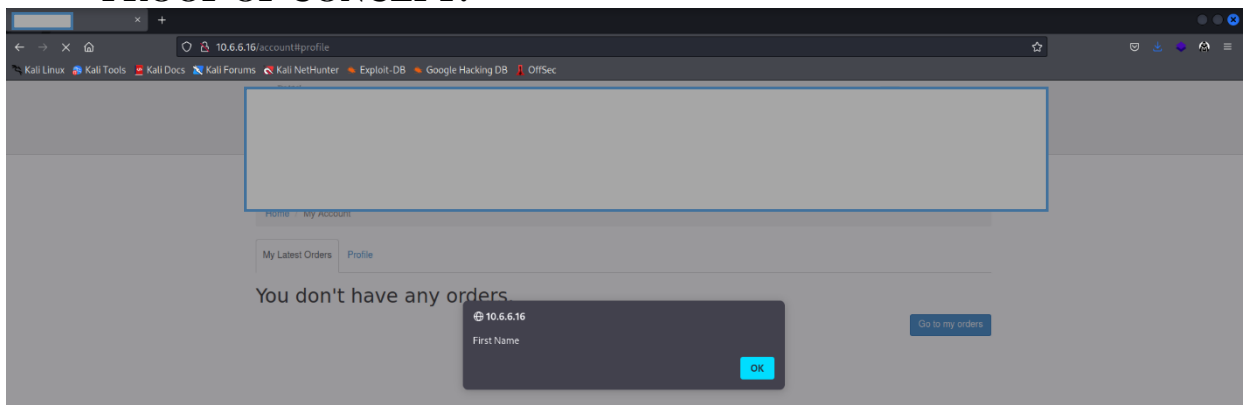
DESCRIPTION:

When user-supplied input is saved and subsequently rendered within a web page, stored XSS takes place. Message boards, blog comments, user profiles, and username fields are common entry sites for cached XSS. Attackers often take advantage of this flaw by injecting XSS payloads on popular website pages or by sending a victim a link that would force them to access the page with the stored XSS payload. When a victim accesses the website, the victim's web browser executes the payload client side. Persistent cross-site scripting, or persistent XSS, are other names for stored XSS.

STEPS TO REPRODUCE:

1. go to <http://10.6.6.16:80> and go to login page
2. Register new user and input "<script>alert("First Name")</script>" "first name field.
3. Login with your credential.
4. Pop up will be shown.

PROOF OF CONCEPT:



IMPACT:

An attacker may often completely compromise a victim if they have control over a script that is running in their browser. The attacker is capable of doing any operations that can be used to exploit reflected XSS vulnerabilities.

REMEDIATION:

An XSS version that relies on the application's capacity to persistently preserve user input on the target server. Then, unaware consumers acquire this data from the software without any sanitization or validation. Attackers can use HTML databases and contemporary apps that use HTML5 to permanently store the malicious payload on the browser.

4.2.2 CROSS SITE SCRIPTING (REFLECTED)

SUMMARY:

| | | |
|------------------|--|-----------------|
| Severity: | HIGH | SL No: 2 |
| Host: | http://10.6.6.16:80 | |
| Path: | /search?id=&searchString= | |
| Method: | _GET() | |
| Payload: | <script>alert("1")</script> | |

DESCRIPTION:

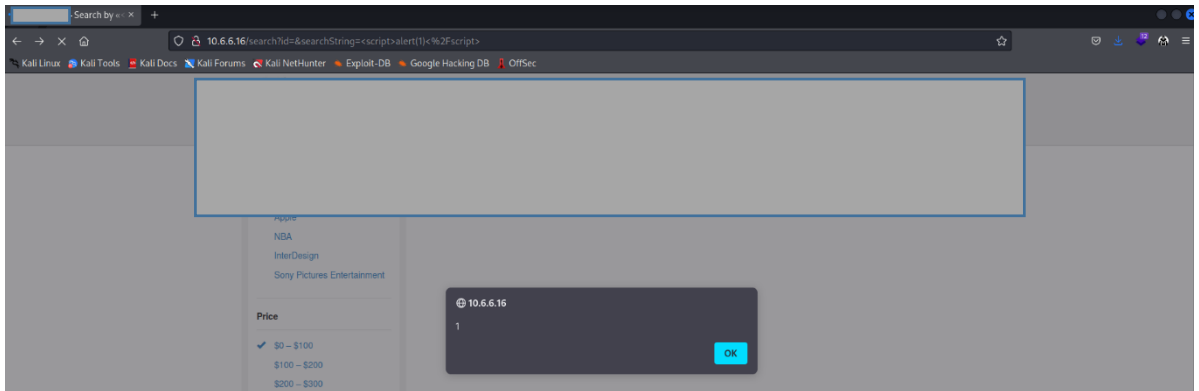
Users must either click on a malicious link or go to a fake website with a web form that when submitted to the target site would start the attack in order to be vulnerable to reflected cross-site scripting attacks. Using a malicious form is typically utilized when a site only accepts HTTP POST requests. In this case, the form might be sent automatically without the victim's knowledge (e.g. by using JavaScript). Once the user clicks on the malicious link or submits the infected form, the XSS payload will be echoed back, where the user's browser will interpret and execute it. Another method to send essentially random requests is by using an embedded client (GET and POST).

STEPS TO REPRODUCE:

1. go to <http://10.6.6.16:80>
2. Search “<script>alert(“1”)</script>”.

3. Pop up will be shown.

Proof of Concept:



IMPACT:

An attacker may often completely compromise a client if they have control over a script that is running in their browser. The attacker has access to, among other things:

- Every action a user can do within the program.
- Examine any data that the user is able to access.
- Change any data that the user has the ability to change.
- Start conversations with other app users and launch harmful assaults that seem to come from the original victim user.

REMEDIATION:

- Sanitizing inputs.
- Using the HTTPOnly cookie option;
- Implementing the content security policy.
- Using the X-XSS-Protection Header

4.2.3 EXTERNAL REDIRECT

SUMMARY:

| | | |
|------------------|---|-----------------|
| Severity: | HIGH | SL No: 3 |
| Host: | http://10.6.6.16:80 | |
| Path: | /user/login?return_url= | |
| Method: | _POST() | |
| Payload: | 8307007217615419775.owasp.org | |

DESCRIPTION:

URL redirectors are frequently used by websites to route incoming requests to alternate resources. There are several reasons why this may be done, but the most common ones are to allow resources to be moved throughout the directory structure and to avoid breaking functionality for users who request the resource in its previous location. It is also feasible to use URL redirectors for load balancing, keeping track of outbound links, or employing abbreviated URLs. Phishing efforts commonly make advantage of this last implementation, as can be seen in the sample below. Even though URL redirectors may not always directly compromise security, attackers can use them to deceive consumers into believing they are visiting a website other than the one they were really looking for.

STEPS TO REPRODUCE:

1. open burp suite and go to <http://10.6.6.16:80> login page with burp browser.
2. Login with your credential while intercept one and modify login url with http://10.6.6.16:80/user/login?return_url=8307007217615419775.owasp.org
3. You will see http 302 response from server which tell redirect possible.

PROOF OF CONCEPT:

```
Original request
1 POST /user/login HTTP/1.1
2 Host: 10.6.6.16
3 Content-Length: 32
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://10.6.6.16
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.82 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer: http://10.6.6.16/
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Cookie: visited_products=%2C72%2C202%2C202%2F2%2C1%2C; PHPSESSID=061jbeshigfjnrulnfpauphj62
14 Connection: close
15
16 username=test3&password=abcd1234

Response
1 HTTP/1.1 302 Found
2 Date: Thu, 17 Feb 2022 07:22:45 GMT
3 Server: Apache/2.4.7 (Ubuntu)
4 X-Powered-By: PHP/5.5.9-1ubuntu4.24
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
7 Pragma: no-cache
8 Location: 8307007217615419775.owasp.org
9 Content-Length: 0
10 Connection: close
11 Content-Type: text/html; charset=utf-8
12
13
```

IMPACT:

External redirection is the process of changing a URL externally such that the user agent is unaware of the change. The user and the browser are both aware that one URL has changed when an external redirection takes place.

REMEDIATION:

We advise at the very least filtering based on protocol handler for apps where whitelisting is just not possible. Making ensuring you just redirect to websites may be done with a regex like `https?://` (and not JavaScript: handlers).

It is advised that a "speed bump" be installed if your application is a secure authenticated application. A screen warning that users are departing for an external URL would appear as a result.

4.2.4 REMOTE FILE INCLUSION

SUMMARY:

| | | |
|-----------|-------------------------|----------|
| Severity: | HIGH | SL No: 4 |
| Host: | http://10.6.6.16:80 | |
| Path: | /user/login?return_url= | |
| Method: | _POST() | |
| Payload: | http://www.google.com/ | |

DESCRIPTION:

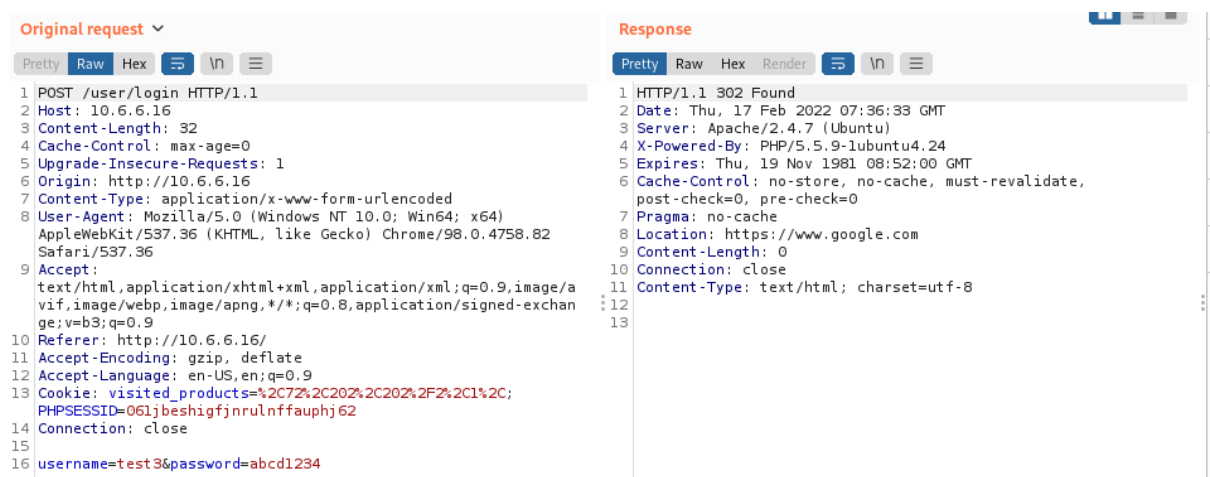
An attack method called Remote File Inclusion (RFI) is used to take advantage of "dynamic file include" capabilities in web applications. Web applications may be fooled into including remote files with malicious code when they receive user input (URL, parameter value, etc.) and send it into file include instructions.

The insertion of files is supported by almost all online application frameworks. File inclusion is mostly used to package shared code into independent files that are subsequently referred to by the main application modules. The code in an inclusion file may be run automatically or explicitly by executing particular functions when a web application accesses it. The web application may be vulnerable to RFI if the decision over which module to load is made based on information from the HTTP request.

STEPS TO REPRODUCE:

1. Open Burp suite and go to <http://10.6.6.16:80> login page with burp browser.
2. Login with your credential while intercept one and modify login URL with http://10.6.6.16:80/user/login?return_url=https://www.google.com
3. You will redirect to <http://www.google.com>.

PROOF OF CONCEPT:



The screenshot displays the Burp Suite interface with two panels: 'Original request' and 'Response'. The 'Original request' panel shows a POST request to /user/login with various headers and a body containing login credentials. The 'Response' panel shows an HTTP 302 Found status with headers indicating a redirect to https://www.google.com.

```
Original request
1 POST /user/login HTTP/1.1
2 Host: 10.6.6.16
3 Content-Length: 32
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://10.6.6.16
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.82
  Safari/537.36
9 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
  image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer: http://10.6.6.16/
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Cookie: visited_product=s%2C72%2C202%2C202%2F2%2C1%2C;
  PHPSESSID=061jbeshigfjnruInfauphj62
14 Connection: close
15
16 username=test3&password=abcd1234

Response
1 HTTP/1.1 302 Found
2 Date: Thu, 17 Feb 2022 07:36:33 GMT
3 Server: Apache/2.4.7 (Ubuntu)
4 X-Powered-By: PHP/5.5.9-lubuntu4.24
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate,
  post-check=0, pre-check=0
7 Pragma: no-cache
8 Location: https://www.google.com
9 Content-Length: 0
10 Connection: close
11 Content-Type: text/html; charset=utf-8
12
13
```

IMPACT:

RFI is a disruption that occurs in an electrical circuit as a result of electromagnetic radiation or conduction from an outside source. The disruption may hinder, impede, impair, or restrict the circuit's ability to operate effectively.

REMEDIATION:

The best way to get rid of file inclusion vulnerabilities is to never give any filesystem/framework API any user-provided input. There is no attack surface for malevolent users to modify the route since any requests with incorrect identifiers must be refused.

4.2.5 REMOTE OS COMMAND INJECTION

SUMMARY:

| | | |
|------------------|----------------------------------|-----------------|
| Severity: | HIGH | SL No: 5 |
| Host: | http://10.6.6.16:80 | |
| Path: | /accounts/documents?page= | |
| Method: | _GET() | |
| Payload: | cat /etc/passwd | |

DESCRIPTION:

An application server hosting problem known as OS command injection, sometimes known as shell injection, allows an attacker to execute arbitrary operating system (OS) instructions, thus compromising the program and all of its data. By taking advantage of trust links and redirecting the attack to various systems inside the business, a hacker may commonly utilize an OS command injection vulnerability to compromise other components of the hosting infrastructure.

STEPS TO REPRODUCE:

1. go to <http://10.6.6.16:80> and go to login page
2. Login with your credential.
3. go to <http://10.6.6.16:80/accounts/documents?page=cat> /etc/passwd
4. You will see webserver user information.

PROOF OF CONCEPT:



IMPACT:

Without having to introduce malicious code, the attacker increases a susceptible application's default functionality by making it pass instructions to the system shell. Command injection frequently offers the attacker more power over the target system.

REMEDIATION:

Strong input validation must be carried out if calling out to OS commands with user input is deemed to be an unavoidable need. Effective validation includes, for instance:

- Verifying against an approved set of values.
- Verifying that a number was entered.
- Checking for the presence of just alphanumeric characters and no other syntax or whitespace in the input.

Never attempt to escape shell metacharacters in order to sanitize input. In actuality, this is just too error-prone and open to being defeated by a cunning opponent.

IMPACT:

There are several effects that SQL injection may have on a corporation. A successful attack might lead to the illegal access of user lists, the deletion of whole tables, and, in some circumstances, the attacker obtaining administrator rights to a database—all very bad things for a company.

REMEDIATION:

The most secure approach to dealing with SQL injection is probably to switch from accepting direct input in a web app to utilizing a secure API. Typically, this will employ prepackaged calls rather than making a call to the SQL translator. The easiest way to prevent SQL injection attacks is to build a parameterized API.

4.2.7 NO RATE LIMIT

SUMMARY:

| | | |
|------------------|----------------------------|-----------------|
| Severity: | MEDIUM | SL No: 7 |
| Host: | http://10.6.6.16:80 | |
| Path: | /users/login | |
| Method: | _POST() | |
| Payload: | ' | |

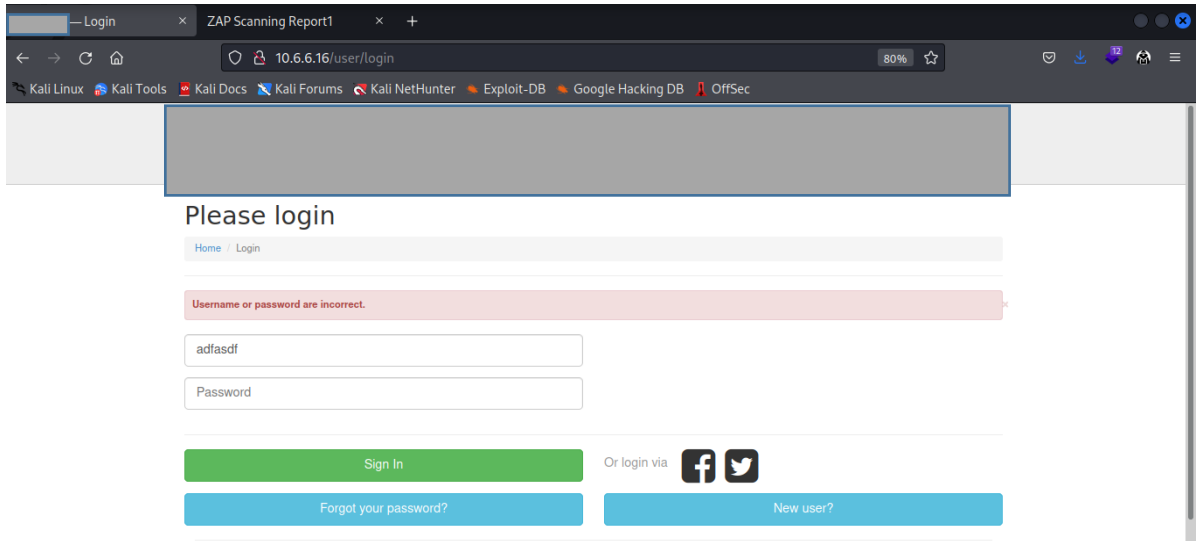
DESCRIPTION:

Forgotten passwords don't have a rate limit check, which might lead to mass mailings and employee spam to users. A quick synopsis of Rate Limit A rate limitation algorithm is used to assess if the user session (or IP address) needs to be limited based on the information in the session cache.

STEPS TO REPRODUCE:

1. go to <http://10.6.6.16:80/users/login>
2. try invalid credential several times.
3. You will see websites do not block you.

PROOF OF CONCEPT:



IMPACT:

This type of attack can cause you to lose money and can slow down your services if you use any email service software APIs or other tools that charge you for your emails. Although users who are impacted by this vulnerability may stop using your services, which might put your business at risk, it can take up a significant amount of storage in sent mail.

REMEDIATION:

If you receive a lot of requests, use CAPTCHA verification.

4.2.8 NO SSL/TLS

SUMMARY:

| | | |
|------------------|----------------------------|-----------------|
| Severity: | MEDIUM | SL No: 8 |
| Host: | http://10.6.6.16:80 | |
| Path: | / | |
| Method: | _GET() | |
| Payload: | ' | |

DESCRIPTION:

SSL/TLS is not implemented.

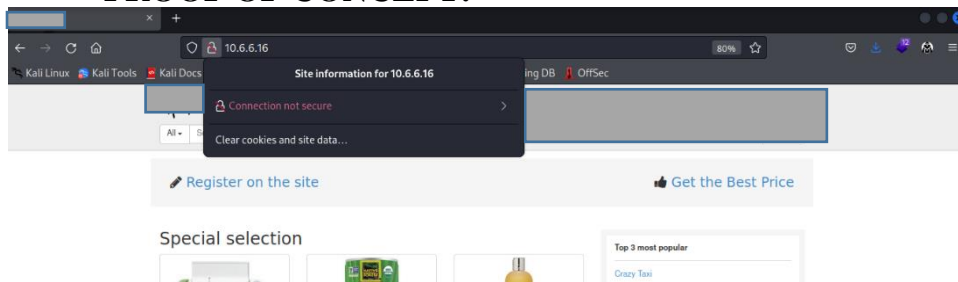
IMPACT:

Any communications sent between your server and users can be read and altered by an attacker who is able to intercept network traffic from either you or your users. This implies that a hacker might view passwords in plain text, change the way your website looks, reroute users to other websites, or collect session data. As a result, nothing you provide to the server is kept a secret.

STEPS TO REPRODUCE:

1. go to `http://10.6.6.16:80`
2. You will see No SSL/TLS certificate available.

PROOF OF CONCEPT:



REMEDIATION:

We advise you to correctly install SSL/TLS, maybe by utilizing the Certbot tool made available by the Let's Encrypt certificate authority. The majority of contemporary web servers, including Apache and Nginx, may have SSL/TLS automatically configured. The tool and the certificates are both free, and they may both be installed quickly.

4.2.9 APPLICATION ERROR DISCLOSURE

SUMMARY:

| | | |
|------------------|----------------------------|-----------------|
| Severity: | MEDIUM | SL No: 9 |
| Host: | http://10.6.6.16:80 | |
| Path: | /twitter | |
| Method: | _GET() | |
| Payload: | ' | |

DESCRIPTION:

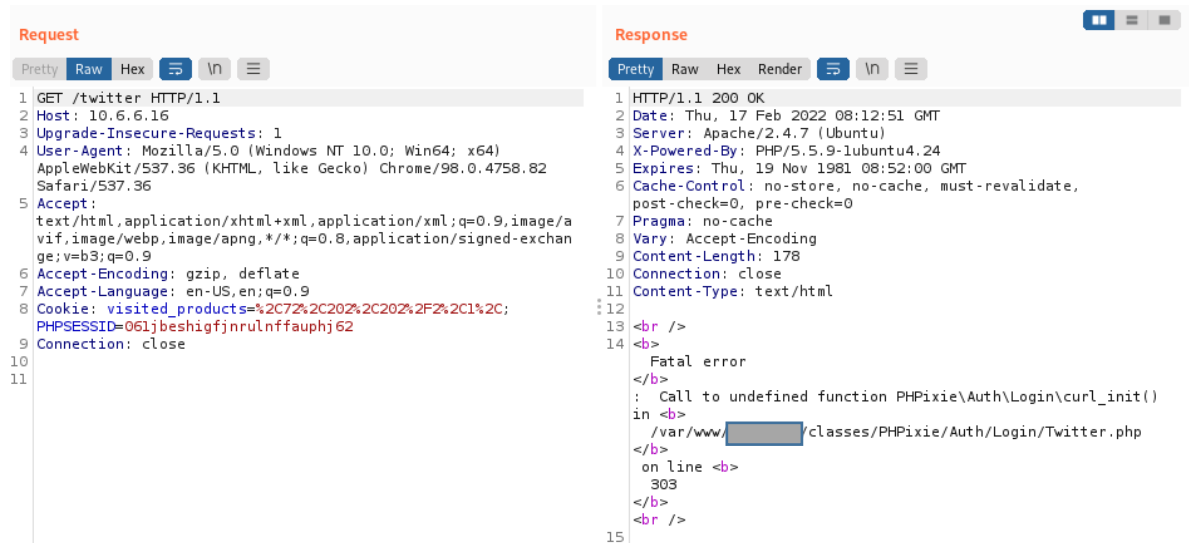
An error or warning message on this page may reveal private information, such as the name of the file that generated the unhandled exception. Additional attacks against the

web application may be launched using this information. If the problem notice is located inside a manual page, the warning can be a false positive.

STEPS TO REPRODUCE:

1. go to <http://10.6.6.16:80/twitter>
2. An Application Error will be shown.

PROOF OF CONCEPT:



```
Request
1 GET /twitter HTTP/1.1
2 Host: 10.6.6.16
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.82
Safari/537.36
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9
8 Cookie: visited_product=%2C7Z%2C202%2C202%2F%2C1%2C;
PHPSESSID=061jbeshigfjnruInffauphj62
9 Connection: close
10
11

Response
1 HTTP/1.1 200 OK
2 Date: Thu, 17 Feb 2022 08:12:51 GMT
3 Server: Apache/2.4.7 (Ubuntu)
4 X-Powered-By: PHP/5.5.9-1ubuntu4.24
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate,
post-check=0, pre-check=0
7 Pragma: no-cache
8 Vary: Accept-Encoding
9 Content-Length: 178
10 Connection: close
11 Content-Type: text/html
12
13 <br />
14 <b>
Fatal error
</b>
: Call to undefined function PHPixie\Auth\Login\curl_init()
in <b>
/var/www/
classes/PHPixie/Auth/Login/Twitter.php
</b>
on line <b>
303
</b>
<br />
15
```

IMPACT:

Depending on the website's goal and the information an attacker can access as a result, information disclosure vulnerabilities can have both a direct and an indirect effect. The simple act of exposing sensitive material may, in some circumstances, have a significant effect on the persons involved. Consider the serious repercussions that may result from an online store revealing the credit card information of its clients. On the other hand, technical information that is leaked, such as the directory structure or the third-party frameworks that are being utilized, may not have much of an immediate effect. This knowledge, however, may be essential in the wrong hands to build a variety of different attacks. Depending on what the attacker can do with this knowledge, the severity of this situation will vary.

REMEDIATION:

Check out this page's source code. Create unique error pages. It could be a good idea to build a system that gives the client (browser) a specific error reference or identification while logging the specifics on the server side and hiding them from the user.

4.2.10 MISSING ANTI-CLICKJACKING HEADER

SUMMARY:

| | | |
|------------------|----------------------------|------------------|
| Severity: | MEDIUM | SL No: 10 |
| Host: | http://10.6.6.16:80 | |
| Path: | / | |
| Method: | _GET() | |
| Payload: | ' | |

DESCRIPTION:

Neither the X-Frame-Options to guard against "Clickjacking" attacks nor the Content-Security-Policy with "frame-ancestors" directive are present in the answer.

STEPS TO REPRODUCE:

1. Open Burp suite and go to <http://10.6.6.16:80> with burp browser.
2. When Intercept on you will see Anti-clickjacking header missing/not available.

PROOF OF CONCEPT:

```
Request
1 GET / HTTP/1.1
2 Host: 10.6.6.16
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.82
  Safari/537.36
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
  image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9
8 Cookie: visited_products=%2C7%2C202%2C202%2F%2C1%2C;
  PHPSESSID=061jbesbigfjnrlnffauqh62
9 Connection: close
10
11

Response
1 HTTP/1.1 200 OK
2 Date: Thu, 17 Feb 2022 08:19:01 GMT
3 Server: Apache/2.4.7 (Ubuntu)
4 X-Powered-By: PHP/5.5.9-1ubuntu4.24
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate,
  post-check=0, pre-check=0
7 Pragma: no-cache
8 Vary: Accept-Encoding
9 Content-Length: 74185
10 Connection: close
11 Content-Type: text/html; charset=utf-8
12
13 <!DOCTYPE html>
14 <html lang="en">
15 <head>
16 <meta charset="utf-8">
17 <title>
18 </title>
19 <meta name="viewport" content="width=device-width,
  initial-scale=1.0">
20 <meta name="description" content="">
21 <meta name="author" content="">
22
```

IMPACT:

When a user clicks on a button or hyperlink on a frame page instead of the top-level page as intended, the attacker has tricked them using several transparent or opaque layers. This technique is known as clickjacking. By doing this, the attacker "hijacks" clicks intended for their page and diverts them to another page, most likely controlled by a different application, domain, or both.

Keystrokes can likewise be taken over using a similar method. A user can be tricked into thinking they are typing their email or bank account password into a text box when, in reality, they are typing into an invisible frame that is under the attacker's control thanks to a well designed combination of stylesheets, iframes, and stylesheets.

REMEDIATION:

The Content-Security-Policy and X-Frame-Options HTTP headers are supported by contemporary web browsers. Make sure at least one of them is active on every web page your site or app returns.

Use SAMEORIGIN if you anticipate that only pages hosted on your server will frame the page (for example, if it is a component of a FRAMESET), otherwise, use DENY if you never anticipate that the page will be framed. Alternately, think about using the "frame-ancestors" directive from Content Security Policy.

4.2.11 VULNERABLE JS LIBRARY

SUMMARY:

| | | |
|------------------|----------------------------|------------------|
| Severity: | MEDIUM | SL No: 11 |
| Host: | http://10.6.6.16:80 | |
| Path: | /js/ | |
| Method: | _GET() | |
| Payload: | jquery-1.10.2.js | |

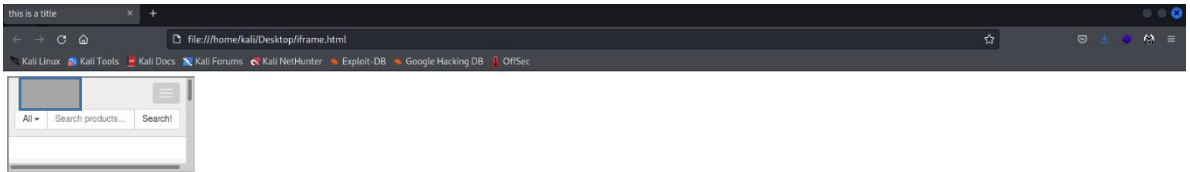
DESCRIPTION:

The identified library jQuery, version 1.10.2 is vulnerable.

STEPS TO REPRODUCE:

1. Go to <http://10.6.6.16:80/js/jquery-1.20.2.js> with browser.
2. You will see vulnerable JS library.

PROOF OF CONCEPT:



IMPACT:

When a user clicks on a button or hyperlink on a framed website when they intended to click on the top level page, the attacker has tricked them by using several transparent or opaque layers. This technique is known as clickjacking. As a result, the attacker is "hijacking" clicks intended for their page and diverting them to a different page, most likely owned by a different application, domain, or both. Keystrokes can likewise be hijacked with a similar method. A user can be duped into thinking they are entering in the password to their email or bank account when, in reality, they are typing into an invisible frame that is under the attacker's control using a skillfully constructed combination of stylesheets, iframes, and text boxes.

REMEDIATION:

Implement X-frame Header and CSP Header.

4.2.13 ABSENCE OF ANTI-CSRF TOKEN

Summary:

| | | |
|------------------|----------------------------|------------------|
| Severity: | LOW | SL No: 13 |
| Host: | http://10.6.6.16:80 | |
| Path: | / | |
| Method: | _GET() | |
| Parameter | ' | |
| : | | |

DESCRIPTION:

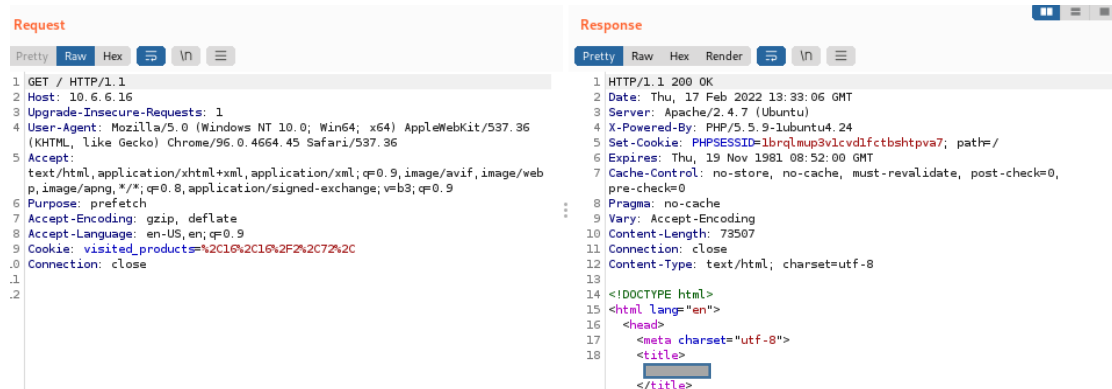
In a cross-site request forgery attack, a target is made to submit an HTTP request to the target location without their knowledge or consent so that the attacker can act in place of the victim. The root reason is application functionality employing recurring,

predictable URL/form operations. The attack's nature is that CSRF takes advantage of the user's confidence in a website. Cross-site scripting (XSS), in contrast, preys on a user's confidence in a website. CSRF attacks are not always cross-site, like XSS, although they can be. CSRF, XSRF, one-click attacks, session riding, confused deputy, and sea surf are other names for cross-site request forgery.

STEPS TO REPRODUCE:

1. Open Burp suite and go to <http://10.6.6.16:80> with burp browser.
2. When Intercept on you will see Anti-CSRF Token Absence/not available.

PROOF OF CONCEPT:



IMPACT:

Lack of Anti-CSRF tokens might make it easier for someone to hijack someone else's account by altering their email address and password, or, if the attack is launched from the administrator account, secretly add a new admin user account.

One of your online application forms, such the email/password update form, may have been copied by the attacker.

The homepage will have a form with the same set of fields as the original application, but with pre-populated input values, and a Javascript script replacing the submit button to trigger auto-submission. When a user accesses the page, the form is instantly submitted, and the page's contents are either changed with valid content or redirected to the user's original application

REMEDIATION:

1. The token value shouldn't be predictable; it might, for example, be produced using a reliable random number generator that is set up correctly.
2. Tokens should lose their value quickly and cannot be used again.

3. Local timestamps should not be used as tokens in the absence of server-side encryption.
4. Avoid sending anti-CSRF tokens in HTTP GET requests to prevent URL or request header leaks.

4.2.14 COOKIE NO HTTPONLY FLAG

SUMMARY:

| | | |
|-----------------------|---------------------|------------------|
| Severity: | LOW | SL No: 14 |
| Host: | http://10.6.6.16:80 | |
| Path: | / | |
| Method: | _GET() | |
| Parameter : | PHPSESSID | |

DESCRIPTION:

Whenever a server delivers data to a user's web browser, it does so in the form of an HTTP cookie. The Set-Cookie header, which was supplied by the web server, is where the HTTP cookies that were previously sent are stored. The browser deletes session cookies when it closes, and if a cookie is persistent, it will expire at the period specified by Expires or Max-Age. By inserting an extra "HttpOnly" signal in the Set-Cookie HTTP response header, the possibility of client-side scripts accessing the protected cookie can be reduced. So even if the online application has a cross-site scripting (XSS) vulnerability, the browser won't divulge the cookie to a third party.

STEPS TO REPRODUCE:

1. Open Burp suite and go to <http://10.6.6.16:80> with burp browser.
2. When Intercept on you will see Cookie HttpOnly flag missing/not available.

PROOF OF CONCEPT:

```

Request
1 GET / HTTP/1.1
2 Host: 10.6.6.16
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
6 Purpose: prefetch
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Cookie: visited_product=%2C16%2C16%2F2%2C7%2C
10 Connection: close
11
12
13
14
15
16
17
18

Response
1 HTTP/1.1 200 OK
2 Date: Thu, 17 Feb 2022 13:33:06 GMT
3 Server: Apache/2.4.7 (Ubuntu)
4 X-Powered-By: PHP/5.5.9-1ubuntu4.24
5 Set-Cookie: PHPSESSID=1brqlmup3v1cud1fctbshstpva7; path=/
6 Expires: Thu, 19 Nov 1981 08:52:00 GMT
7 Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
8 Pragma: no-cache
9 Vary: Accept-Encoding
10 Content-Length: 73507
11 Connection: close
12 Content-Type: text/html; charset=utf-8
13
14 <!DOCTYPE html>
15 <html lang="en">
16 <head>
17 <meta charset="utf-8">
18 <title>
19
20 </title>
  
```

IMPACT:

A cross-site scripting attack gives the attacker easy access to cookies, which he can use to take over the victim's session. The cookie's sensitive information can be stolen by an attacker.

REMEDIACTION:

Set the HTTPOnly attribute of the cookie. As a result, there are fewer XSS attacks that try to get cookies and could divulge personal information or allow the attacker to take the identity of the user.

The HTTP TRACE method and XSS can access the authentication cookie even when the HttpOnly option is set. Therefore, ensure that HTTP TRACE is off.

4.2.15 COOKIE WITHOUT SAME SITE ATTRIBUTE

SUMMARY:

| | | |
|------------------|----------------------------|------------------|
| Severity: | LOW | SL No: 15 |
| Host: | http://10.6.6.16:80 | |
| Path: | / | |
| Method: | _GET() | |
| Parameter | PHPSESSID | |
| : | | |

DESCRIPTION:

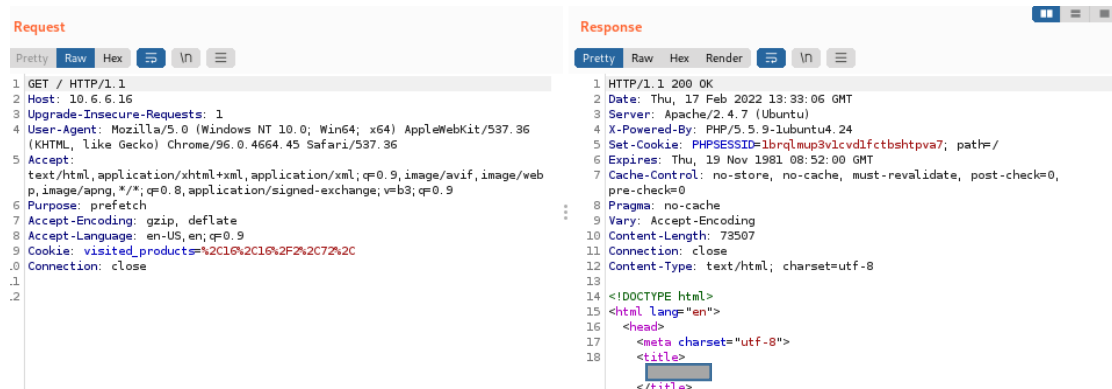
It is possible for a cookie to be transmitted as a consequence of a "cross-site" request since the cookie has been set without the SameSite property. Timing, script inclusion,

and cross-site request forgery attacks may all be successfully defended against using the SameSite property.

STEPS TO REPRODUCE:

1. Open Burp suite and go to <http://10.6.6.16:80> with burp browser.
2. When Intercept on you will see cookie are using without same site attribute.

PROOF OF CONCEPT:



IMPACT:

A Cross-site Request Forgery (CSRF) attack can result from a cookie without the SameSite Attribute. Declaring whether the cookie should be limited to a first-party or same-site context is possible using the "SameSite" property.

REMEDIATION:

For all cookies, make sure the SameSite attribute is either set to "lax" or, ideally, "strict."

4.3 PENETRATION TESTING METHODOLOGIES

The professionals will quickly identify any flaws during the assessment using tried-and-true non-invasive test procedures. The application is exposed and misused from a number of angles, including those that don't have certificates, user IDs, and privileged user IDs. The primary security problems with online programs are authentication bypass, injection, account traversal, privilege escalation, and data extraction.

Our method covers all of the OWASP Top 10 web application security risks.

| Ref | Risk | Description |
|-----|-----------------------------|--|
| A1 | Injection | An interpreter provides untrusted data as part of a command or query in order to take advantage of injection vulnerabilities such as SQL, NoSQL, OS, and LDAP injections. As a result of the intruder's aggressive data, the interpreter could carry out unlawful instructions or obtain unauthorized access to data. |
| A2 | Broken Authentication | Many times, authentication and session management functions are handled incorrectly, allowing third parties to either steal their credentials, keys, or session tokens or to exploit other implementation flaws to accept other user identities either temporarily or permanently. |
| A3 | Sensitive Data Exposure | A lot of the time, online apps and APIs fail to adequately safeguard sensitive data, such as financial, medical, and PII. Attackers are free to alter or steal data that is not adequately protected in order to perpetrate robberies, identity theft, and other crimes using credit cards. Without additional security, such as encryption at rest or in transit, sensitive data may be hacked. |
| A4 | XML External Entities (XXE) | XML documents are frequently used by outdated or improperly built XML processors to identify external entities. Internal file |

| | | |
|----|----------------------------|--|
| | | disclosure can be accomplished by external parties via the URI handler, internal file sharing, internal port scanning, remote code execution, and service denial attacks. |
| A5 | Broken Access Control | Additionally, limitations on what authorized users can do are not adequately enforced. Attackers can utilize these vulnerabilities to access illegal data and/or functionality, such as other people's accounts, confidential files, changing other people's data, changing access privileges, etc. |
| A6 | Security Misconfiguration | Misconfiguration in security is the main worry. It frequently results from insecure default settings, insufficient or ad hoc configurations, cloud open storage, improperly configured HTTP headers, and sensitive data-containing error messages. Operating systems, frameworks, libraries, and programs must all be installed safely and timely patches or upgrades must be applied. |
| A7 | Cross-Site Scripting (XSS) | When an application modifies the content of a web page with user information using a browser API to create JavaScript or HTML, updates the page itself with questionable information, or both, XSS vulnerabilities can result. With the use of XSS, attackers can launch scripts on their target computers that can hijack user sessions, sabotage websites, or direct users to harmful web pages. |

| | | |
|-----|-------------------------------------|---|
| A8 | Insecure Deserialization | Remote code execution is also facilitated by unsafe deserialization. Deserialization flaws can still be leveraged to launch replay attacks, insertion attacks, and privilege escalation attacks even if remote code execution is prevented. |
| A9 | Using Known Vulnerable Components | The program modules, including libraries, framing systems, and other running components, all have the same privileges. A significant data loss or server takeover may be triggered if a weak component is employed. |
| A10 | Insufficient Logging and Monitoring | Lack of sufficient logging and monitoring, as well as integration issues or insufficient incident response, allow attackers to carry out additional attacks, retain persistence, switch to other systems, and change, extract, or delete data. The majority of research on violation contend that violations are often noticed for longer than 200 days, and typically by outside actors rather than internal ones. |

CHAPTER 5: CONCLUSIONS

Vulnerability testing and vulnerability assessment are two types of vulnerability testing (VAPT). In order to determine whether unauthorized access or other malicious activity is possible, penetration testing attempts to exploit system flaws. VAPT Certification is a technological solution to address security issues in an organization's IT infrastructure.

Reference

- Agathoklis Prodromou. "Examples of TLS/SSL Vulnerabilities TLS Security 6: | Acunetix." *Acunetix*, 31 Mar. 2019, www.acunetix.com/blog/articles/tls-vulnerabilities-attacks-final-part/. Accessed 14 Oct. 2022.
- "Application Error Disclosure." *Application Error Disclosure*, 2018, beaglesecurity.com/blog/vulnerability/application-error-disclosure.html. Accessed 14 Oct. 2022.
- author. "Clickjacking Web Vulnerability | OWASP Top 10 Security Testing | Best Web App Security Testing Services Company | Cyber Security Whitepapers | Pune Mumbai Hyderabad Delhi Bangalore Ahmedabad Kolkata India Dubai Bahrain Qatar Kuwait Singapore Australia USA UK Germany Croatia Botswana Mauritius." *Valencynetworks.com*, 2022, www.valencynetworks.com/kb/clickjacking-x-frame-options-header-missing.html. Accessed 14 Oct. 2022.
- Banach, Zbigniew. "Protect Your Website with Anti-CSRF Tokens | Invicti." *Invicti*, 10 June 2020, www.invicti.com/blog/web-security/protecting-website-using-anti-csrf-token/. Accessed 14 Oct. 2022.
- "Broken Authentication." *Contrastsecurity.com*, 2017, www.contrastsecurity.com/glossary/broken-authentication#:~:text=Authentication%20is%20%E2%80%9Cbroken%E2%80%9D%20when%20attackers,of%20broken%20authentication%20is%20wide%20spread.. Accessed 14 Oct. 2022.

- “Cookie without HttpOnly Flag Set.” *Portswigger.net*, 2022, portswigger.net/kb/issues/00500600_cookie-without-httponly-flag-set. Accessed 14 Oct. 2022.
- “Deserialization - OWASP Cheat Sheet Series.” *Owasp.org*, 2016, cheatsheetseries.owasp.org/cheatsheets/Deserialization_Cheat_Sheet.html. Accessed 14 Oct. 2022.
- “ECyLabs: Website Security Platform.” *Www.ecylabs.com*, 2022, ecylabs.com/blog/2021/06/21/application-error-disclosure/#:~:text=This%20page%20contains%20an%20error,attacks%20against%20the%20web%20application.. Accessed 14 Oct. 2022.
- Fletcher, Justin. “OWASP API Security - 10: Insufficient Logging & Monitoring.” *Tyk API Management*, 27 May 2022, tyk.io/blog/res-owasp-api-security-10-insufficient-logging-monitoring/. Accessed 14 Oct. 2022.
- “IBM Documentation.” *Ibm.com*, 19 May 2022, www.ibm.com/docs/en/cdfsp/7.6.1.x?topic=checklist-vulnerability-cookie-without-samesite-attribute. Accessed 14 Oct. 2022.
- “IBM Documentation.” *Ibm.com*, 19 May 2022, www.ibm.com/docs/en/control-desk/7.6.1.x?topic=checklist-vulnerability-server-leaks-information. Accessed 14 Oct. 2022.
- “Indusface.” *Indusface*, 2 Sept. 2014, www.indusface.com/blog/components-known-vulnerabilities/. Accessed 14 Oct. 2022.
- “Insecure Deserialization | Tutorials & Examples | Snyk Learn.” *Snyk Learn*, 2022, learn.snyk.io/lessons/insecure-deserialization/java/. Accessed 14 Oct. 2022.

“INSUFFICIENT LOGGING and MONITORING.” *Contrastsecurity.com*, 2017,
www.contrastsecurity.com/glossary/insufficient-logging-and-monitoring.
Accessed 14 Oct. 2022.

ManageEngine. “What Is Security Misconfiguration? | Misconfiguration Vulnerability
- ManageEngine Vulnerability Manager Plus.” *Manageengine.com*, 2020,
www.manageengine.com/vulnerability-management/misconfiguration/.
Accessed 14 Oct. 2022.

“Missing X-Frame-Options Header | Invicti.” *Invicti*, 15 Aug. 2022,
[www.invicti.com/web-vulnerability-scanner/vulnerabilities/missing-x-frame-
options-header/](http://www.invicti.com/web-vulnerability-scanner/vulnerabilities/missing-x-frame-options-header/). Accessed 14 Oct. 2022.

“OS Command Injection | Learn AppSec | Invicti.” *Invicti*, 28 Mar. 2022,
www.invicti.com/learn/os-command-injection/. Accessed 14 Oct. 2022.

“OWASP Top 10 Vulnerabilities | Veracode.” *Veracode*, 4 Dec. 2020,
www.veracode.com/security/owasp-top-10. Accessed 14 Oct. 2022.

“OWASP Top Ten | OWASP Foundation.” *Owasp.org*, 2020, [owasp.org/www-project-
top-ten/](http://owasp.org/www-project-top-ten/). Accessed 14 Oct. 2022.

“OWASP Top Ten 2017 | A3:2017-Sensitive Data Exposure | OWASP Foundation.”
Owasp.org, 2017, [owasp.org/www-project-top-ten/2017/A3_2017-
Sensitive_Data_Exposure](http://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure). Accessed 14 Oct. 2022.

“OWASP ZAP – External Redirect.” *Zaproxy.org*, 2017,
www.zaproxy.org/docs/alerts/20019-1/. Accessed 14 Oct. 2022.

Pauli, Josh. “The Basics of Web Hacking.” *The Basics of Web Hacking*, 2013, pp. 1–
18, [www.sciencedirect.com/topics/computer-science/injection-
vulnerability#:~:text=Injection%20occurs%20when%20a%20hacker,still%20](http://www.sciencedirect.com/topics/computer-science/injection-vulnerability#:~:text=Injection%20occurs%20when%20a%20hacker,still%20)

widespread%20and%20very%20damaging., 10.1016/b978-0-12-416600-4.00001-0. Accessed 14 Oct. 2022.

“SameSite Cookie Not Implemented | Invicti.” *Invicti*, 15 Aug. 2022, www.invicti.com/web-vulnerability-scanner/vulnerabilities/samesite-cookie-not-implemented/. Accessed 14 Oct. 2022.

Synack. “Preventing Broken Access Control: The No.1 Vulnerability in the OWASP Top 10 2021.” *Synack*, 12 Jan. 2022, www.synack.com/blog/preventing-broken-access-control-the-no-1-vulnerability-in-the-owasp-top-10-2021/. Accessed 14 Oct. 2022.

“Using Components with Known Vulnerabilities | Secure Coach.” *Securecodewarrior.com*, 2015, learn.securecodewarrior.com/secure-coding-guidelines/using-components-with-known-vulnerabilities. Accessed 14 Oct. 2022.

“Vulnerable JavaScript Libraries - Vulnerabilities - Acunetix.” *Acunetix*, 2019, www.acunetix.com/vulnerabilities/web/vulnerable-javascript-libraries/. Accessed 14 Oct. 2022.

“Vulnerable Javascript Library.” *Vulnerable Javascript Library*, 2021, beaglesecurity.com/blog/vulnerability/vulnerable-javascript-library.html. Accessed 14 Oct. 2022.

Wallarm. “A9: Using Components with Known Vulnerabilities 2017 OWASP.” *Wallarm.com*, 2017, www.wallarm.com/what/a9-using-components-with-known-vulnerabilities-2017-owasp. Accessed 14 Oct. 2022.

---. “Security Misconfiguration.” *Wallarm.com*, 2019, www.wallarm.com/what/security-misconfiguration. Accessed 14 Oct. 2022.

- “What Is Cross-Site Scripting and How Can You Fix It?” *Acunetix*, 21 July 2022, www.acunetix.com/websitesecurity/cross-site-scripting/. Accessed 14 Oct. 2022.
- “What Is Cross-Site Scripting?” *Sucuri*, 28 July 2022, sucuri.net/guides/what-is-cross-site-scripting/. Accessed 14 Oct. 2022.
- “What Is Cross-Site Scripting? XSS Cheat Sheet | Veracode.” *Veracode*, 4 Dec. 2020, www.veracode.com/security/xss. Accessed 14 Oct. 2022.
- “What Is OS Command Injection, and How to Prevent It? | Web Security Academy.” *Portswigger.net*, 2022, [portswigger.net/web-security/os-command-injection#:~:text=OS%20command%20injection%20\(also%20known,application%20and%20all%20its%20data..](https://portswigger.net/web-security/os-command-injection#:~:text=OS%20command%20injection%20(also%20known,application%20and%20all%20its%20data..) Accessed 14 Oct. 2022.
- “What Is Rate Limiting? | Rate Limiting and Bots.” *Cloudflare*, 2022, www.cloudflare.com/learning/bots/what-is-rate-limiting/. Accessed 14 Oct. 2022.
- “What Is RFI | Remote File Inclusion Example & Mitigation Methods | Imperva.” *Learning Center*, 29 Dec. 2019, [www.imperva.com/learn/application-security/rfi-remote-file-inclusion/#:~:text=Remote%20file%20inclusion%20\(RFI\)%20is,located%20within%20a%20different%20domain..](http://www.imperva.com/learn/application-security/rfi-remote-file-inclusion/#:~:text=Remote%20file%20inclusion%20(RFI)%20is,located%20within%20a%20different%20domain..) Accessed 14 Oct. 2022.
- “What Is Security Misconfiguration? | Secure Code Warrior.” *Securecodewarrior.com*, 2019, www.securecodewarrior.com/blog/coders-conquer-security-share-learn-series-security-misconfigurations. Accessed 14 Oct. 2022.
- “What Is XXE (XML External Entity) Injection? Tutorial & Examples | Web Security Academy.” *Portswigger.net*, 2022, portswigger.net/web-security/xxe. Accessed 14 Oct. 2022.

“X-Frame-Options Header Not Set | ScanRepeat.” *ScanRepeat*, 2020,
scanrepeat.com/web-security-knowledge-base/x-frame-options-header-not-
set#:~:text=When%20X%2DFrame%2DOptions%20Header,ads%2C%20clic
kjacking%20code%2C%20etc.. Accessed 14 Oct. 2022.

