DEPLOYMENT OF HTTP WEB SERVER , DEMONSTRATION AND MITIGATION OF CROSS SITE REQUEST FORGERY

Report submitted to the SASTRA Deemed to be University
as the requirement for the course

CSE302: COMPUTER NETWORKS

Submitted by

Swathipriya. V. K
(Reg. No.: 124003329, B.Tech CSE)

DECEMBER 2022

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

THANJAVUR | KUMBAKONAM | CHENNAI

SCHOOL OF COMPUTING
THANJAVUR, TAMIL NADU, INDIA – 613 401

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA
THANJAVUR | KUMBAKONAM | CHENNAI

SCHOOL OF COMPUTING
THANJAVUR, TAMIL NADU, INDIA – 613 401

Bonafide Certificate

This is to certify that the report titled "**Deployment of HTTP web server, demonstration and mitigation of Cross Site Request Forgery**" submitted as a requirement for the course, **CSE302: COMPUTER NETWORKS** for B.Tech. is a Bonafide record of the work done by **Ms. Swathipriya.V.K (Reg. No.124003329, B.Tech CSE**) during the academic year 2022-23, in the School of Computing.

Project Based Work *Viva voce* held on _____

Examiner 1 **Examiner 2**

List of Figures

ABBREVIATIONS

HTTP Hyper Text Transfer Protocol

ABSTRACT

In today's world, the use of technology has become prevalent in nearly every aspect of our lives. From online banking and shopping to social media and communication, we rely on applications to perform a wide range of tasks. Because of this, it is essential for these applications to be secure in order to protect our personal information and prevent security breaches. One type of application that is commonly used is the HTTP server. HTTP servers are responsible for handling requests and responses between clients and servers on the World Wide Web. They are used on a daily basis by many people, and they are often developed by new or inexperienced developers. However, if these developers do not have a strong understanding of security, they may create applications that are vulnerable to attack.

Improper development and insecure coding can lead to a wide range of vulnerabilities in applications. One of the common and highly impactful vulnerabilities is cross site request forgery (CSRF). CSRF occurs when a malicious actor tricks a user into making an unintended request to a website. This can allow the attacker to gain access to sensitive information or perform actions on the user's behalf, such as making unauthorized transactions or changing account settings. In order to combat this vulnerability, it is important to implement effective mitigation strategies in the backend. This can involve using secure coding practices, implementing authentication and authorization measures, and using specialized tools and technologies to detect and prevent CSRF attacks. By taking these steps, developers can create applications that are more secure and better able to protect against potential threats.

KEY WORDS: HTTP, World Wide Web, CSRF, Authentication, and Authorization

TABLE OF CONTENTS

CHAPTER 1 - INTRODUCTION

1.1 Internet and World Wide Web

The internet is a global network of interconnected computers and computer networks. It allows computers to communicate with each other and exchange information using a common language and set of protocols. The internet has transformed the way we live, work, and interact with each other, and it has made it possible for people to access a vast amount of information and resources from anywhere in the world.

The World Wide Web (WWW or Web) is a collection of interconnected documents and other resources, linked by hyperlinks and URLs. The Web was developed in the late 1980s and early 1990s, and it has become an integral part of the internet. It allows users to easily access and share information and resources on the internet, and it has greatly expanded the reach and capabilities of the internet.

1.2 Website

A web page is a single document on the internet that can be accessed by entering its URL in a web browser. It typically includes text, images, and links to other web pages. A website is a group of related web pages that are organized together and typically includes a homepage and other pages.

1.3 Web Server

A web server is a computer that runs specialized software, such as Apache or nginx, to host and serve web pages. When a user opens a web page in their web browser, the browser sends a request to the web server for the specific page or content. The web server then retrieves the requested page from its storage and sends it back to the user's browser, which displays the page on the user's device. In this way, web servers enable users to access and view web pages through the internet. Web servers may also handle other tasks, such as executing server-side scripts, processing user requests, and managing user sessions.A web server is a computer that stores web pages and makes them available to users over the internet. It responds to requests from web browsers and sends the requested web page to the user.

1.4 HTTP Server

An HTTP server is a type of computer software that is responsible for handling requests and responses between clients and servers on the World Wide Web. HTTP servers use the Hypertext Transfer Protocol (HTTP) to communicate with clients, which typically includes web browsers and other applications that access the Web. When a client makes a request to an HTTP server, the server processes the request, retrieves the requested information, and sends a response back to the client. HTTP servers are used on a daily basis by many people, and they are an essential part of the infrastructure of the Web.

1.5 Secure Coding Concept

Secure coding is the practice of writing computer programs and applications in a way that protects against potential security vulnerabilities and threats. It involves using best practices and techniques for development and coding, as well as incorporating security measures and controls into the design of the software. Secure coding is important because it helps to prevent security breaches and protect against potential attacks, such as malware, hackers, and other types of malicious activity. By adopting secure coding practices, developers can create software that is more reliable, secure, and trustworthy for all users.

1.6 Security Problems and Vulnerabilities

Web applications are vulnerable to a wide range of security problems and vulnerabilities. These can include vulnerabilities in the code of the application, inadequate authentication and authorization measures, and insecure configuration of the application or the server on which it is hosted. Other potential security issues include cross-site scripting (XSS), cross-site request forgery (CSRF), SQL injection, and denial of service (DoS) attacks. These vulnerabilities can allow attackers to gain access to sensitive information, compromise the security of the application, or disrupt the availability of the application for legitimate users.

1.7 Cross site request forgery

Cross site request forgery (CSRF) is a type of cyber attack that exploits the trust that a website has in a user's browser. This type of attack is often referred to as a "one-click attack" because it only requires a user to click on a malicious link or visit a compromised website in order to execute the attack.

In a CSRF attack, the attacker creates a malicious link or website that contains a forged request that is sent to a vulnerable website on behalf of the user. This forged request may be used to perform actions such as transferring funds, changing user account settings, or modifying sensitive data.

Since the request appears to be legitimate and is initiated by the user's own browser, the vulnerable website is unable to distinguish the forged request from a legitimate one. As a result, the website processes the request and performs the malicious action.

1.9 Cookies, Authentication and Authorization

Cookies are small pieces of data that are stored on a user's device by a website. They are often used to store information such as user preferences and settings, as well as to track user activity and maintain user sessions.

Authentication refers to the process of verifying a user's identity. This may involve requiring the user to provide a username and password, or using other methods such as biometric authentication or two-factor authentication.

Once a user has been authenticated, authorization is the process of determining whether the user has permission to access certain resources or perform certain actions. This may involve checking the user's role or permissions within the system, as well as any applicable policies or rules.

Together, cookies, authentication, and authorization help to ensure that users are able to access only the resources and perform only the actions that they are authorized to. This helps to protect the security and integrity of the system and its data.

1.8 Mitigation and Blue teaming

Mitigation refers to the process of reducing or eliminating the impact of a security threat or vulnerability. This may involve implementing security controls, updating software, or taking other preventive measures to prevent an attack from occurring or to minimize its impact.

Blue teaming is a security strategy that involves actively monitoring and defending against potential threats. This may include conducting regular security

assessments, implementing security controls, and responding to security incidents in a timely and effective manner.

Blue teaming is often used in conjunction with other security strategies, such as red teaming (which involves simulating attacks to test the effectiveness of security measures) and purple teaming (which involves collaboration between red and blue teams to improve security). Together, these strategies help to ensure the overall security and resilience of an organization's systems and data.

CHAPTER 2 - TECHNICAL DETAILS

2.1 Npm package manager

2.2 Nodejs framework

2.3 Expressjs framework

2.4 Cookies

2.4.1 Jsonwebtoken

2.5 Database

2.5.1 Nosql

2.5.2 Mongodb

2.6 Vulnerabilities

2.6.1 Cross site request forgery

2.6.2 Access control and authentication

2.6.3 Cross origin resources policy

2.6.4 Iframe injection

2.7 Phishing

CHAPTER 3 - WORKING DETAILS

2.1 Npm package manager

CHAPTER 4 - SNAPSHOTS

CONCLUSION AND FUTURE SCOPE

LITERATURE SURVEY

REFERENCES