

# **Sri Lanka Institute of Information Technology**



## **Bug Bounty Report - 08**

**Module: IE2062**

**Web Security**

**Year 2, Semester 2**

**Aazaf Ritha. J – IT23151710**

**B.Sc. (Hons) in Information Technology**

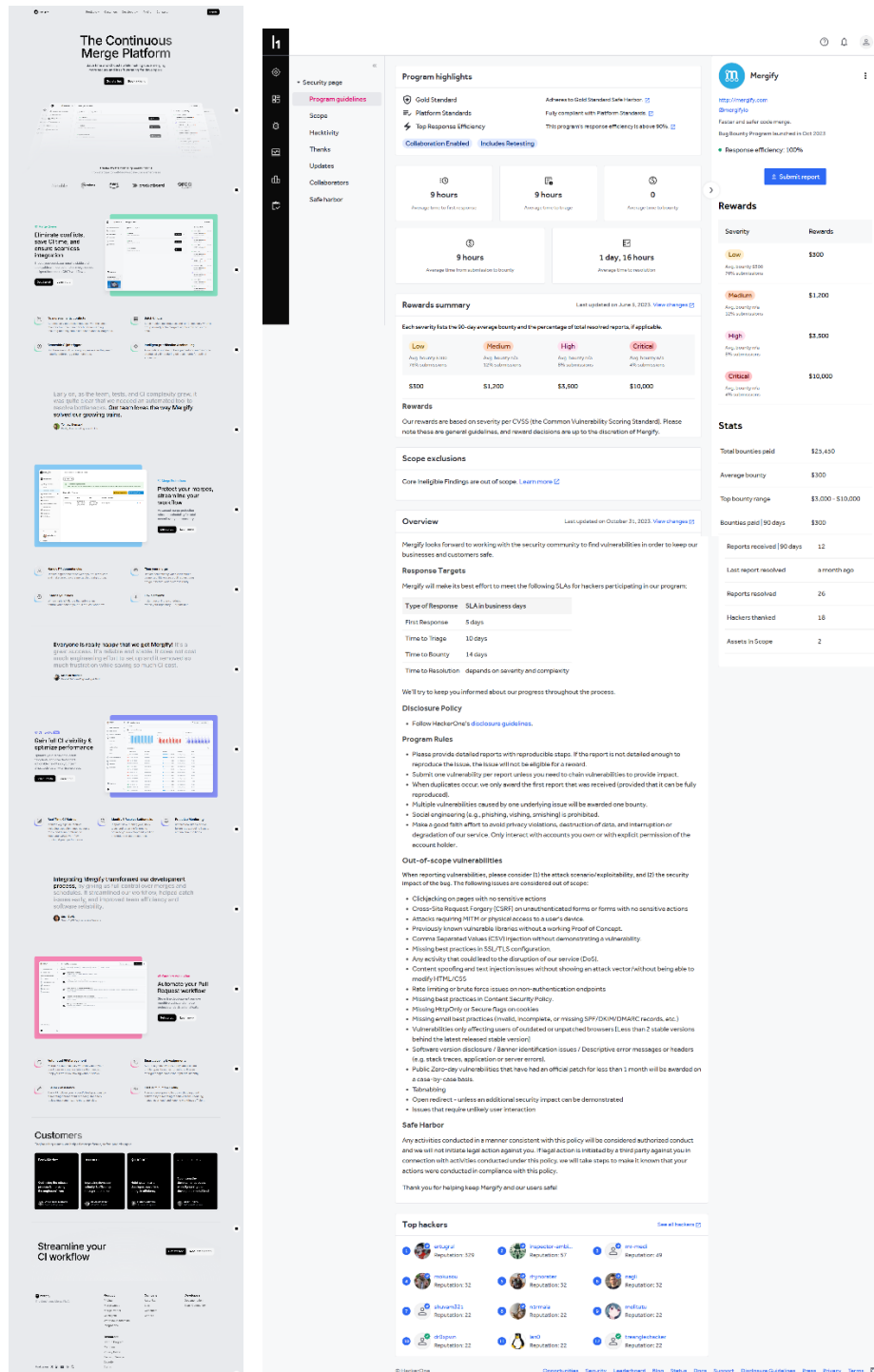
**Specialized in Cyber Security**

## Table of Contents

Introduction .....	3
Vulnerability Title .....	4
Description.....	5
Affected Component.....	6
Impact Assessment .....	7
Steps to Reproduce.....	8
Proof of Concept (Screenshots) .....	9
Proposed Mitigation or Fix.....	11
Conclusion.....	12

# Introduction

This report outlines a missing Content Security Policy (CSP) header misconfiguration discovered on Mergify's public website (<https://mergify.com>) as part of the HackerOne bug bounty program. The issue was confirmed through both manual inspection and automated security scanning. This document provides technical details, clear reproduction steps, proof-of-concept evidence, and practical mitigation recommendations to address the missing CSP header vulnerability.



The screenshot displays the Mergify website, which serves as a platform for bug bounty hunters. The page is divided into several sections:

- Program Highlights:** This section lists key features of the Mergify program, including Gold Standard status, adherence to the Gold Standard Security Framework, and a 90% response efficiency. It also mentions that the program is fully consistent with Platform Standards and includes a response efficiency of 90%.
- Rewards:** A table showing the reward structure based on severity levels. The table has two columns: Severity and Rewards.
 

Severity	Rewards
Low	\$300
Medium	\$1,200
High	\$3,800
Critical	\$10,000
- Stats:** A section providing an overview of the program's performance, including total bounties paid (\$25,430), average bounty (\$300), top bounty range (\$3,000 - \$10,000), and reports received (12).
- Security Policy:** A detailed section outlining the program's security policy, including response targets, disclosure policy, and rules for reporting vulnerabilities. It states that Mergify will make its best effort to meet the following SLAs for hackers participating in the program:
  - Type of Response: SLA in business days
  - First Response: 5 days
  - Time to Triage: 10 days
  - Time to Bounty: 14 days
  - Time to Resolution: depends on severity and complexity
- Out-of-scope vulnerabilities:** A list of vulnerabilities that are not eligible for a reward, including:
  - Clickjacking on pages with no sensitive actions
  - Cross-Site Request Forgery (CSRF) on unauthenticated forms or forms with no sensitive actions
  - Attacks requiring HTTP or physical access to a user's device
  - Previously known vulnerable libraries without a working Proof of Concept
  - Common Sequential Values (CSV) injection without demonstrating a vulnerability
  - Missing best practices in SSL/TLS configuration
  - Any activity that could lead to the disruption of our service (DoS)
  - Content spoofing and text injection issues without showing an attack vector without being able to modify HTML/CSS
  - Rate limiting or brute force issues on non-authentication endpoints
  - Missing best practices in Content Security Policy
  - Missing HTTPOnly or Secure flags on cookies
  - Missing email best practices (invalid, incomplete, or missing SPF/DKIM/DMARC records, etc.)
  - Vulnerabilities only affecting users of outdated or unsupported browsers (less than 2 stable versions behind the latest released stable version)
  - Software version disclosure / Banner identification issues / Descriptive error messages or headers (e.g. stack traces, application or server errors)
  - Public zero-day vulnerabilities that have had an official patch for less than 1 month will be awarded on a case-by-case basis
  - Tampering
  - Open redirect - unless an additional security impact can be demonstrated
  - Issues that require arbitrary user interaction

## **Vulnerability Title**

**Title** – Content Security Policy (CSP) Header Not Set

**Risk Level-** Medium

**Domain** – <https://mergify.com/>

## Description

A **Content Security Policy (CSP) Header Not Set** issue was identified on Mergify's public website (<https://mergify.com>). The site fails to implement a Content-Security-Policy HTTP response header, a critical security control that helps mitigate a broad range of client-side attacks, including Cross-Site Scripting (XSS), data injection, and clickjacking.

The absence of a CSP header means the browser receives no instructions on which resources are safe to load or execute. This increases the risk of malicious inline scripts or untrusted third-party content being executed within the context of the application. As a result, the vulnerability falls under **OWASP Top 10 – A05:2021 (Security Misconfiguration)** and **A07:2021 (Identification and Authentication Failures)** when combined with weak session management or user-controlled content.

If an attacker manages to inject or reflect malicious scripts into the application, for example, via query parameters, form fields, or third-party compromised assets, there is no policy to restrict or block the execution of such payloads. This could lead to **session hijacking, credential theft, unauthorized actions, or leakage of sensitive user data**. Implementing a strict CSP header would significantly reduce the attack surface for such exploits.

## Affected Component

**HTTP Response Headers:** The application does not include the Content-Security-Policy header in its HTTP responses.

**Web Application Frontend:** All publicly accessible pages, including the homepage and subpages of <https://mergify.com>, are affected by this missing configuration.

**Browser Security Controls:** In the absence of a defined CSP, modern browsers are unable to enforce restrictions on script sources, style sources, or other potentially dangerous content, leaving the client-side application exposed to XSS and other injection-based attacks.

## Impact Assessment

**Cross-Site Scripting (XSS):** Without defined CSP, malicious scripts injected via user input or third-party services can execute in the user's browser, allowing attackers to hijack sessions, deface content, or perform unauthorized actions on behalf of the user.

**Data Exfiltration:** Injected JavaScript can access and transmit sensitive data such as authentication tokens, cookies, and personal identifiable information (PII) to attacker-controlled servers.

**Session Hijacking:** Lack of CSP allows exploitation paths where attackers can steal session identifiers or local storage tokens, gaining unauthorized access to user accounts.

**Phishing & UI Redress Attacks:** An attacker can manipulate the DOM or inject fake login forms and overlays, deceiving users into entering credentials or confidential data.

**Clickjacking:** In the absence of a CSP that includes frame-ancestors 'none', the site is vulnerable to being embedded in iframes on malicious sites, facilitating clickjacking attacks.

**Reputation & Compliance Risk:** Successful exploitation may result in privacy violations or data breaches, undermining user trust and triggering regulatory consequences under laws like GDPR or CCPA.

## Steps to Reproduce

### Open the target site:

Navigate to <https://mergify.com> using any modern web browser (e.g., Google Chrome or Mozilla Firefox).

### Open Developer Tools:

Press F12 or right-click on the page and select **Inspect** to open the browser's developer tools.

### Go to the Network tab:

Refresh the page (Ctrl+R) and observe the response headers for the main document (usually listed as / or index in the Network tab).

### Check for CSP Header:

In the headers section, under **Response Headers**, look for the Content-Security-Policy header.

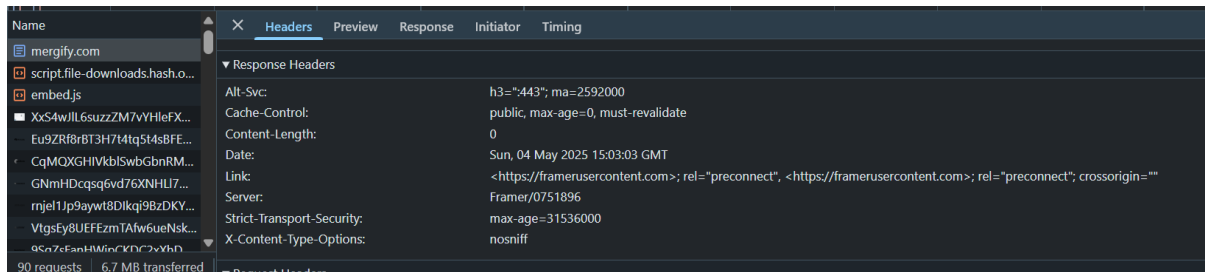
### Confirm absence of CSP:

Note that no Content-Security-Policy or Content-Security-Policy-Report-Only header is present in the response.



## Proof of Concept (Screenshots)

### Manual Inspection PoC (Header Check):



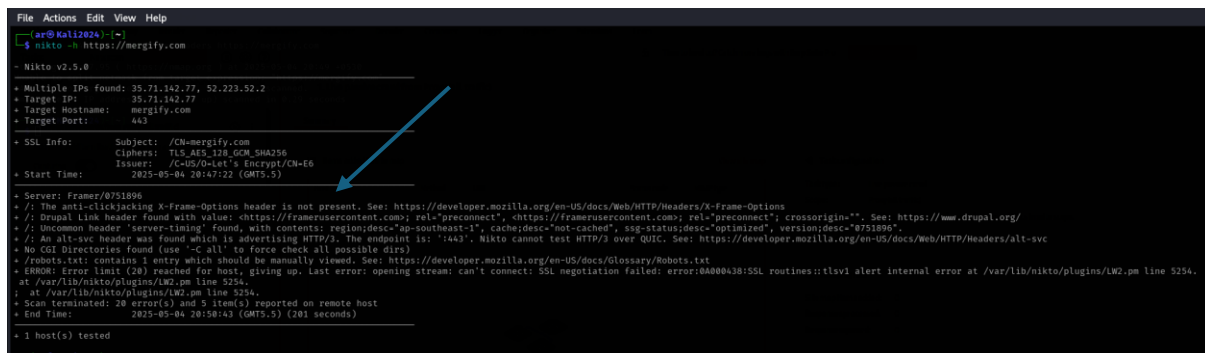
### Expected (Secure):

Content-Security-Policy: default-src 'self'; script-src 'self'; object-src 'none';

**Actual (Vulnerable):** *(Header is missing entirely)*

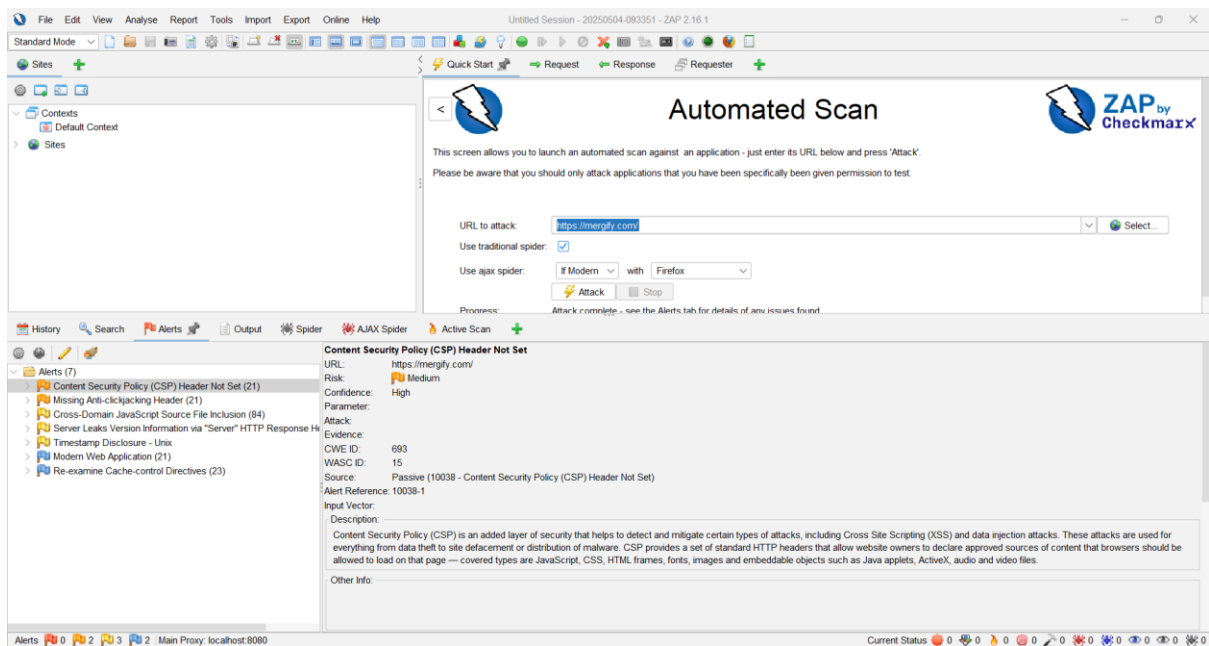
### Nikto Scan:

Code: nikto -h https://mergify.com



The screenshot above confirms that the relevant security headers are missing using nikto scan.

## OWASP ZAP Scan:



This scan confirms that <https://mergify.com> is missing a **CSP header**, which is a widely recommended security best practice for preventing browser-based attacks. The scanner classified this as a **medium-severity issue with high confidence**, meaning it is confirmed and should be addressed.

## Proposed Mitigation or Fix

**Implement a Strict Content Security Policy (CSP):** Define and apply a CSP header that restricts the sources of scripts, styles, images, and other content to trusted origins only. A strong baseline example:

```
Content-Security-Policy: default-src 'self'; script-src 'self' https://trusted.cdn.com;  
object-src 'none'; style-src 'self' 'unsafe-inline'; base-uri 'self'; frame-ancestors 'none';
```

**Avoid Using unsafe-inline and unsafe-eval:** Refactor inline scripts and styles into external files where possible and remove reliance on eval() or similar functions to strengthen CSP enforcement.

**Test the CSP in Report-Only Mode:** Start with a Content-Security-Policy-Report-Only header to monitor violations without breaking functionality. Adjust the policy based on observed violations before enforcing it fully.

**Regularly Audit Third-Party Scripts:** Limit third-party script usage and ensure all external scripts are loaded over HTTPS from trusted sources. Use Subresource Integrity (SRI) where applicable.

**Automate Policy Testing:** Integrate CSP validation into your CI/CD pipeline or use tools like Mozilla Observatory or CSP Evaluator to validate the correctness and coverage of your policy.

**Educate Developers:** Provide internal guidelines and best practices to help developers write secure front-end code compatible with CSP.

## Conclusion

The absence of a Content Security Policy on Mergify's website introduces avoidable security risks, particularly around client-side attacks like XSS and clickjacking. Implementing a robust CSP not only enhances security posture but also aligns with modern web security best practices and compliance requirements. This report highlights the importance of proactive header configuration and secure frontend development.