



Sri Lanka Institute of Information Technology

IE2062

Web Security

Bug Bounty Report II

Submitted by:

Student Registration Number	Student Name
IT21197550	Nihila Premakanthan

Date of Submission: 28.05.2023

Acknowledgement

I would like to express my special thanks to our mentor Ms. Chethana Liyanapathirana and Dr. Lakmal Rupansighe for their time and efforts she provided throughout the course, and for the Web Security lecture panel for guiding us through this semester and for helping us by giving examples, guidelines, and advice about the project. Your useful advice and suggestions were really helpful to me during the project's completion. In this aspect, I am eternally grateful to you.

Executive Summary

This report aims to provide an overview of the vulnerability identified in a particular domain. The bug bounty platform called Hackerone was used for this purpose.

This report uses different tools to gather information detect vulnerabilities and perform penetration testing. The tool name Netsparker and Owasp Zap was mainly used to identify the vulnerability. Further this report provides the vulnerability title, vulnerability description, Affected Components, Impact Assessment, Steps to reproduce the vulnerability, proof of concept and the proposed mitigation.

By including these comprehensive details for each vulnerability, the report provides a comprehensive overview of the security weaknesses present within the system and offers actionable insights for remediation and improvement.

Contents

Vulnerability Title	4
Vulnerability Description.....	4
Affected Components	5
Impact Assessment.....	5
Steps to Reproduce	6
Proposed mitigation	7
External Reference.....	8

Vulnerability Title

Active Mixed Content over HTTPS

Severity Level

Risk Level:
MEDIUM

Vulnerability Description

The term "active mixed content over HTTPS" is a vulnerability that arises when a secure website (using HTTPS) has components that are loaded over an insecure connection (HTTP), such as images, scripts, or iframes. There is a security risk when these insecure parts are put onto a secure page because the integrity and confidentiality of the page's content may be jeopardized.

By encrypting the data sent between a web browser and a website, the HTTPS (Hypertext Transfer Protocol Secure) protocol enables secure communication between them. It offers authentication and confidentiality, preventing eavesdropping and manipulation on critical information.

Because the insecure content loaded into the secure page could potentially be intercepted or altered by an attacker, this vulnerability exists. An attacker could, for instance, substitute a script or picture with malicious content to cause a number of attacks, such as:

1. Man-in-the-Middle (MitM) Attacks: An attacker can intercept insecure content and alter it to transmit harmful payloads, inject code, or steal sensitive data sent between a user and a website.
2. Data Leakage: Attackers may be able to listen in on the sensitive data contained in the unsecured material, such as login credentials, private information, or financial information.
3. Downgrade Attacks: By tricking users into loading the complete page over an unsecured connection, attackers can downgrade the security of the communication to HTTP by taking advantage of the existence of mixed content.

Modern web browsers have added security features to reduce the dangers of active mixed content. For instance, the majority of browsers by default restrict the loading of active mixed material, notifying users and prohibiting the execution of potentially hazardous scripts.

Affected Components

The affected components of active mixed content over HTTPS can include:

- **Images:** Loading images over insecure HTTP connections (mixed content) on an HTTPS page can compromise the security and integrity of the page. Attackers can potentially tamper with or replace the images, leading to potential security risks.
- **Scripts:** JavaScript files loaded over HTTP on an HTTPS page introduce a significant security vulnerability. Attackers can modify the scripts to inject malicious code, steal sensitive information, or perform unauthorized actions on the user's behalf.
- **Iframes:** Iframes that use HTTP rather than HTTPS to load content are also regarded as active mixed content. This is true for both internal iframe references on the webpage and external iframes that are embedded.
- **Objects:** Active mixed content is loaded over HTTP rather than HTTPS and includes objects like Flash or Silverlight. These items could be interactive webpage features or multimedia components.
- **XMLHttpRequest (XHR) requests:** Active mixed content is defined as XHR queries done by JavaScript code to load data from external sources over HTTP rather than HTTPS.
- **WebSockets:** Active mixed content is defined as WebSocket connections created via HTTP rather than HTTPS. Real-time communication between a browser and a server is made possible through WebSockets.

Impact Assessment

Active mixed content over HTTPS can have a variety of effects on a website's operation, security, and privacy. An evaluation of the probable effects is provided below:

- **Security Risks:** Active mixed content poses security problems since the insecure components are vulnerable to interception or modification by attackers. Injecting malicious scripts or replacing resources with malicious copies, for instance, could result in cross-site scripting (XSS) attacks or other types of exploitation.
- **Compromised User Data:** Sensitive user data sent over an insecure HTTP connection may be exposed by active mixed content. This contains private information, login details, and other sensitive information. This data could be intercepted or altered by attackers, which could result in data breaches or unauthorized access to user accounts.

- **Attacks from the middle man:** When active mixed content is served over HTTP, a website may be at risk from attacks from the middle man. Attackers who are positioned in between a user and a website have the ability to intercept and alter content, possibly stealing sensitive data or injecting harmful code.
- **Browser Warnings and Blocks:** Modern browsers actively alert users or prevent websites that contain active mixed material. The user experience may suffer as a result, as users may decide not to continue after seeing warnings that they believe to be signs of a non-secure website. This may lead to a decline in traffic and user engagement as well as a possible erosion of reputation.
- **Functionality Impairment:** Some online features and APIs need a secure environment (HTTPS) to work correctly. These features can be broken by active mixed content, which may result in errors or poor user experiences. On websites with active mixed material, for instance, certain JavaScript APIs and geolocation functionality might not function as expected.
- **Compliance and Regulatory Issues:** Active mixed content might present compliance problems, depending on the nature of the website and the restrictions that apply. For the transmission of sensitive data, many compliance standards, like the Payment Card Industry Data Security Standard (PCI DSS), mandate the usage of secure connections. Penalties or other legal repercussions may follow failure to adhere to these criteria.

Steps to Reproduce

Steps to identify and reproduce active mixed content over HTTPS:

- Set up a local development environment with a web server and SSL/TLS certificate configuration to enable HTTPS.
- Create a webpage with HTTPS enabled and ensure it is served over HTTPS. You can use a simple HTML file for testing purposes.
- Include external resources, such as scripts, images, iframes, or other elements, in the webpage using HTTP URLs instead of HTTPS URLs
- Load the webpage in a web browser. The browser's security features will detect the active mixed content and either block it or show a warning, depending on the browser's settings.
- Inspect the browser console or developer tools to view any console warnings or error messages related to the active mixed content.


Proposed mitigation

The following steps can be made to lessen Active Mixed Content over HTTPS's vulnerability:


- Identify Mixed Content. To find any mixed content on your website, use a website scanning tool or browser developer tools. This will enable you to identify the items that are loading over unsafe HTTP connections.
- Update Resource URLs: Switch all resource URLs (such as those for pictures, scripts, stylesheets, and iframes) to HTTPS. Make sure any resource references on your webpages explicitly use the secure protocol by making changes to the source code.
- Content Delivery Networks (CDNs): If you utilize a CDN, be sure it supports HTTPS and load resources using the secure URLs the CDN provides.
- Verify Dependencies on Third Parties: Examine each script, plugin, or widget that a third party has utilized on your website. Make sure they are updated to support HTTPS and that no insecure material is added.
- Content Security Policy (CSP): Put in place a CSP that mandates HTTPS usage and forbids the loading of mixed material. You can specify the sources that are permitted for different categories of information with CSP, ensuring that only secure sources are accepted.
- Automatically redirect HTTP queries to HTTPS by configuring your web server to do this. This will guarantee that users access your website safely at all times and stop any accidental loading of mixed information.
- Test and Watch: Continually check your website to make sure no mixed information is loading. Keep an eye on the security headers and logs of your website for any indications of mixed material or browser-reported security warnings.
- Educate Administrators and Developers: Inform administrators and developers about the dangers of mixed material. Encourage secure development best practices and stress the significance of using HTTPS for all resources.

By putting these mitigation strategies into place, you may successfully close down active mixed content vulnerabilities and make sure that visitors to your website have a safe surfing experience.

External Reference



Information Technology Laboratory
NATIONAL VULNERABILITY DATABASE


 NATIONAL VULNERABILITY
 DATABASE
 (NVD)

VULNERABILITIES

CVE-2017-7835 Detail


Description

Mixed content blocking of insecure (HTTP) sub-resources in a secure (HTTPS) document was not correctly applied for resources that redirect from HTTPS to HTTP, allowing content that should be blocked, such as scripts, to be loaded on a page. This vulnerability affects Firefox < 57.

Severity

CVSS Version 3.x
 CVSS Version 2.0

CVSS 3.x Severity and Metrics:


 NIST: NVD
 Base Score: **7.2 HIGH**
 Vector: CVSS:3.0(AV:N)(AC:L)(PR:N)(UI:N)(S:U)(C:L)(A:L)

NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

Note: NVD Analysts have published a CVSS score for this CVE based on publicly available information at the time of analysis. The CNA has not provided a score within the CVE List.

QUICK INFO


CVE Dictionary Entry:
 CVE-2017-7835
NVD Published Date:
 06/11/2018
NVD Last Modified:
 10/02/2019
Source:
 Mozilla Corporation

References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

Hypertext	Resource
https://www.securityfocus.com/bid/101832	Third Party Advisory VDB Entry
https://www.securitytracker.com/id/1039803	Third Party Advisory VDB Entry
https://bugzilla.mozilla.org/show_bug.cgi?id=1402363	Issue Tracking Permissions Required
https://www.mozilla.org/security/advisories/mfsa2017-24/	Vendor Advisory

Weakness Enumeration

CWE-ID	CWE Name	Source
NVD-CWE-noinfo	Insufficient Information	 NIST