

ACME Inc.

Security Assessment on ACME’s Corporative web

|  |  |
| --- | --- |
| Client Code | ACME |
| Project | ACME-WEB-001-CLIENTWEB |
| Type | Web |
| Date | 05/09/2020 |

# Contents

[1 Contents 2](#_Toc50204957)

[2 Table of findings 3](#_Toc50204958)

[3 Findings details 4](#_Toc50204959)

[3.1 Stored Cross-Site Script 4](#_Toc50204960)

[Definition 4](#_Toc50204961)

[Description 4](#_Toc50204962)

[Solution 5](#_Toc50204963)

[References 5](#_Toc50204964)

# Table of findings

|  |  |  |
| --- | --- | --- |
| Section | Title | CVSS Score |
| 3.1 | Stored Cross-Site Script | **7.3** |

# Findings details

## Stored Cross-Site Script

|  |  |  |  |
| --- | --- | --- | --- |
| ID | ACME\_20200905\_000001 | | |
| Resources | http://127.0.0.1:5000/assessments  http://127.0.0.1:5000/assessments/id | | |
| OWASP | **OTG-CLIENT-002** | Testing for JavaScript Execution | |
| CVSS v3 | CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:N | | 7.3 |

### **Definition**

Stored cross-site scripting (also known as second-order or persistent XSS) arises when an application receives data from an untrusted source and includes that data within its later HTTP responses in an unsafe way.

If an attacker can control a script that is executed in the victim's browser, then they can typically fully compromise that user. The attacker can carry out any of the actions that are applicable to the impact of reflected XSS vulnerabilities.

In terms of exploitability, the key difference between reflected and stored XSS is that a stored XSS vulnerability enables attacks that are self-contained within the application itself. The attacker does not need to find an external way of inducing other users to make a request containing their exploit. Rather, the attacker places their exploit into the application itself and simply waits for users to encounter it.

### **Description**

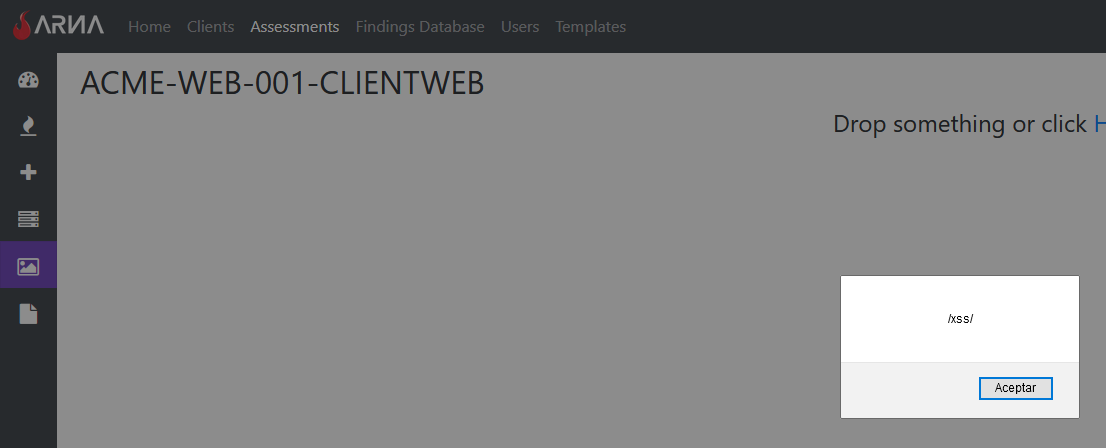
During the security evaluation we have found a Stored XSS *Quia molestiae impedit* et sint ad perferendis repellendus ipsa. Dolores voluptatem et rerum sint nisi omnis laboriosam quisquam. Voluptas eos sunt accusantium aut. Placeat totam ex sed recusandae quam minima voluptatibus. In tempora quidem illo repudiandae temporibus dolore rerum. Voluptatibus aut id eligendi.

Ipsa \*\*laboriosam \*\*dolore nostrum necessitatibus odio impedit cum. Dolores nostrum eveniet odit impedit aut corporis. Et repudiandae et natus enim ipsum.

Ipsam quam sunt molestiae. Non rerum veniam dolores dolorem occaecati. Consequatur quia quod natus maiores. Sed distinctio vero id pariatur natus rerum.

Quis veritatis distinctio labore cumque commodi et. Culpa rerum magnam perferendis est sit. Dignissimos quod rem sit soluta error rerum qui.

Quia molestiae impedit et sint ad perferendis repellendus ipsa. Dolores voluptatem et rerum sint nisi omnis laboriosam quisquam. Voluptas eos sunt accusantium aut. Placeat totam ex sed recusandae quam minima voluptatibus. In tempora quidem illo repudiandae temporibus dolore rerum. Voluptatibus aut id eligendi.

  
Image 1: XSS Evidence

Unordered list of things:

* lorem
* ipsum

Ordered list of things:

1. lorem
2. ipsum

### **Solution**

Go to **OWASP, Cross-Site Scripting Prevention Cheat Sheet -** https://cheatsheetseries.owasp.org/cheatsheets/Cross\_Site\_Scripting\_Prevention\_Cheat\_Sheet.html for more information on how to prevent XSS vulnerabilities.

### **References**

* **PortSwigger, Stored XSS -** https://portswigger.net/web-security/cross-site-scripting/stored