SameSite Lax bypass via method override in a blog webapp

Ref: CVE-2025-34291

Executive Summary

This report presented the findings from a security assessment of a web application's account management functionality. A Cross-Site Request Forgery (CSRF) vulnerability was identified in the email change feature. By leveraging a SameSite=Lax cookie bypass via method override, an attacker can trick authenticated users into changing their email address. This poses a moderate security risk due to potential account manipulation and unauthorized data changes.

Introduction

The purpose of this test was to assess the application's resistance to CSRF attacks under modern browser cookie restrictions. The target was the `change-email` functionality, which allowed users to update their email addresses. The test was conducted in an environment simulating a real-world attack using method override techniques.

Methodology

1. Logged in with test credentials using Burp Suite’s browser.

2. Intercepted and analyzed HTTP requests related to the email update process.

3. Tested SameSite cookie behavior by inspecting session cookie attributes.

4. Explored CSRF exploitability through method override (`_method=POST`) using a cross-site GET request.

5. Crafted and delivered a CSRF proof-of-concept via the exploit server.

Vulnerability Findings

- **Type**: Cross-Site Request Forgery (CSRF)

- **Location**: `GET /my-account/change-email?_method=POST`

- **Severity**: Medium

Description

The application used session cookies without explicitly setting the `SameSite` attribute, causing browsers to apply the default `SameSite=Lax` policy. Under this policy, session cookies were included in top-level navigation GET requests. The email change feature failed to implement CSRF tokens and allows HTTP method override via the `_method` parameter.

This enabled an attacker to perform a CSRF attack by crafting a malicious GET request with `_method=POST`, bypassing the SameSite restriction and changing the victim's email address.

Proof of Concept (PoC)

The following HTML can be used to exploit the vulnerability:

html

CopyEdit

`<script>  document.location = "https://website.id.net/my-account/change-email?email=attacker%40example.com&_method=POST"; </script>`

When visited by an authenticated user, the above script changes their email address to `attacker@example.com`.

Impact Assessment

- Unauthorized modification of user email addresses.

- Potential account hijacking if the attacker controls the new email.

- Violation of user trust and integrity of account settings.

Recommendations

1. Implement anti-CSRF tokens and enforce server-side validation for sensitive operations.

2. Avoid supporting HTTP method override parameters unless absolutely necessary.

3. Explicitly set `SameSite=Strict` or `SameSite=Lax` with `Secure` flags for all session cookies.

4. Enforce Content-Type validation and disallow cross-origin POST and GET requests where possible.

Conclusion

The exploitation of this CSRF vulnerability via method override highlighted the importance of enforcing modern security practices, especially in web applications that manage user data. Remediating this issue will help prevent unauthorized actions and uphold user security.