Report 10

URL: asupport.suivo.com

Session Cookie Not Marked as Secure

HIGH

The Session Cookie Not Marked as Secure vulnerability refers to a security weakness where a session cookie is transmitted over an HTTPS connection but is not marked as secure. A session cookie is a small piece of data that is typically used to maintain session information and authenticate users on a website. When a session cookie is not marked as secure, it means that the cookie can be sent over an unencrypted HTTP connection instead of being restricted to secure HTTPS connections. This poses a significant security risk because it allows an attacker who can intercept the traffic, particularly through a man-in-the-middle attack, to potentially steal the cookie.

The impact of this vulnerability is severe. An attacker who successfully intercepts the traffic and gains access to the session cookie can hijack a victim's session. By hijacking the session, the attacker can impersonate the victim and gain unauthorized access to their account or perform malicious activities on the website.

Affected URL: https://asupport.suivo.com/process_login

Identified Cookie(s): JSESSIONID

Cookie Source: HTTP Header

Affected Components

- Session Management System: The vulnerability affects the session management system employed by the web application. This includes the handling and transmission of session cookies that are used to maintain user sessions and authenticate users.

- Web Server: The web server hosting the application is directly involved in transmitting the session cookies over the network. The vulnerability can affect the secure transmission of cookies from the server to the client browser.

- Client-Side Applications: The client-side applications, such as web browsers, interact with the web server and handle the reception and storage of session cookies. The vulnerability can impact the security of cookie transmission and storage on the client-side.

- Authentication Mechanism: If the session cookies are used for authentication purposes, the vulnerability can directly impact the security of the authentication mechanism. Attackers exploiting this vulnerability can potentially hijack user sessions and bypass authentication controls.

- User's Browser: The user's web browser plays a crucial role in receiving and storing session cookies. If the session cookie is not marked as secure, it can be vulnerable to interception by attackers who have control over the network or perform man-in-the-middle attacks.

Impact Assessment

The Session Cookie Not Marked as Secure vulnerability carries significant risks and can have various impacts on the security of a web application or system.

- Session Hijacking: By intercepting the session cookie transmitted over an unencrypted HTTP connection, an attacker can hijack a user's session. This allows the attacker to gain unauthorized access to the user's account, perform actions on their behalf, and potentially compromise sensitive information.

- Unauthorized Access: Exploiting the vulnerability can lead to unauthorized access to protected areas or resources within the web application. Attackers can leverage the compromised session cookie to bypass authentication controls and gain privileged access to sensitive data or functionality.

- Account Compromise: If the session cookie provides access to user accounts, the vulnerability can result in account compromise. Attackers can manipulate the session cookie to impersonate legitimate users, modify account settings, or perform malicious actions within the system.

- Data Breach: If the session cookie grants access to sensitive data, its compromise can lead to a data breach. Attackers can exploit the vulnerability to gain unauthorized access to confidential information, personally identifiable information (PII), financial records, or other sensitive data stored within the application.

- Privacy Violation: The vulnerability undermines the privacy of users who rely on the secure transmission of session cookies. Attackers can intercept and analyze the cookie contents, potentially exposing user activities, preferences, or other private information.

- Reputation Damage: Successful exploitation of the vulnerability can result in reputational damage for the organization hosting the web application. If user accounts are compromised or sensitive data is exposed, it can erode user trust, harm the organization's reputation, and lead to legal and regulatory consequences.

Steps to Reproduce

- Set up a testing environment: Prepare a testing environment consisting of a web application that uses session cookies for user authentication or session management. Ensure that the application is configured to transmit session cookies over both HTTP and HTTPS connections.

- Intercept network traffic: Use appropriate network interception tools, such as a packet sniffer or a proxy server, to capture the network traffic between the client and the web server.

- Initiate a session: Access the web application using a client browser and initiate a session by logging in or performing any action that triggers the generation of a session cookie.

- Capture the session cookie: Capture the network traffic during the session initiation or subsequent interactions. Look for the request and response headers that include the session cookie.

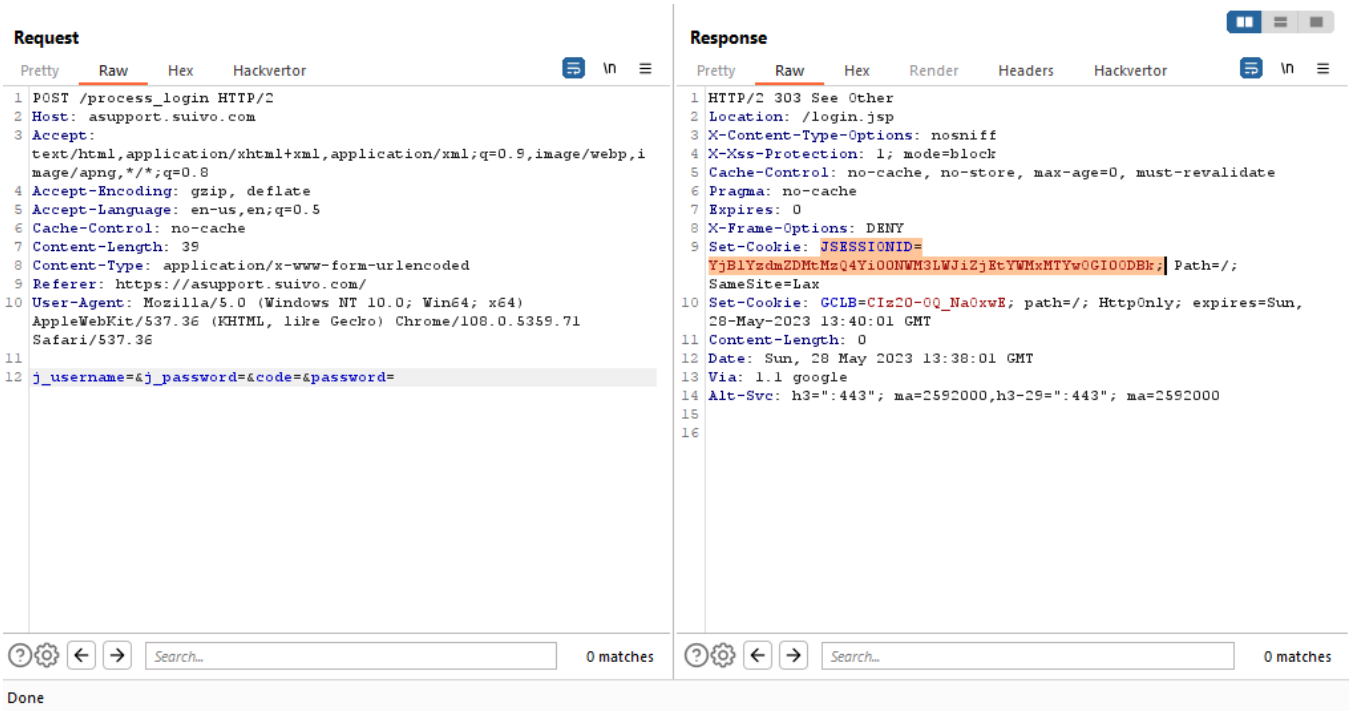
- Analyze the cookie properties: Examine the captured session cookie and check if it has the "Secure" flag set to indicate that it should only be transmitted over secure HTTPS connections.

- Reproduce the vulnerability: Modify the captured session cookie by removing the "Secure" flag or changing it to "false" or "0". Ensure that the modified cookie is sent with subsequent requests to the web server.

- Observe the cookie transmission: Send subsequent requests to the web server, either by navigating through the application or performing any relevant actions. Monitor the network traffic to verify that the session cookie is being transmitted over an unencrypted HTTP connection.

- Validate the vulnerability: Confirm that the modified session cookie is accepted and processed by the web server, even when transmitted over an insecure HTTP connection. This confirms the presence of the vulnerability.

Proof of Concept



Proposed Mitigation or Fix

- ✓ Mark cookies as secure: Modify the web application code to explicitly mark all session cookies as secure. This can be done by adding the "Secure" attribute to the cookie when it is set or generated. This ensures that the cookie is only transmitted over HTTPS connections and not over unencrypted HTTP connections.

- ✓ Enable HTTPS throughout: Ensure that the entire web application is served over HTTPS. This involves configuring the web server to enforce HTTPS connections and redirect all HTTP requests to their HTTPS counterparts. By exclusively using secure connections, the risk of session cookies being transmitted insecurely is eliminated.

- ✓ Implement HTTP Strict Transport Security (HSTS): Enable HSTS on the web server to enforce the use of HTTPS connections. HSTS instructs web browsers to always access the website via HTTPS, even if the user types "http://" in the address bar. This further enhances the security of session cookie transmission.

- ✓ Perform comprehensive security testing: Conduct regular security assessments and penetration testing to identify and address any potential vulnerabilities, including cookie-related issues. This helps identify misconfigurations, weak encryption protocols, or other security weaknesses that could expose session cookies.

- ✓ Review third-party integrations: If the web application relies on third-party services or content delivery networks (CDNs), ensure that they also follow secure cookie practices. Review the security measures implemented by these third parties and ensure that they enforce the use of secure cookies.