Sri Lanka Institute of Information Technology

Missing Secure,HttpOnly,SameSite Flags on Cookies - Report 10

IT23187214

Web Security - IE2062

Vulnerability Title:

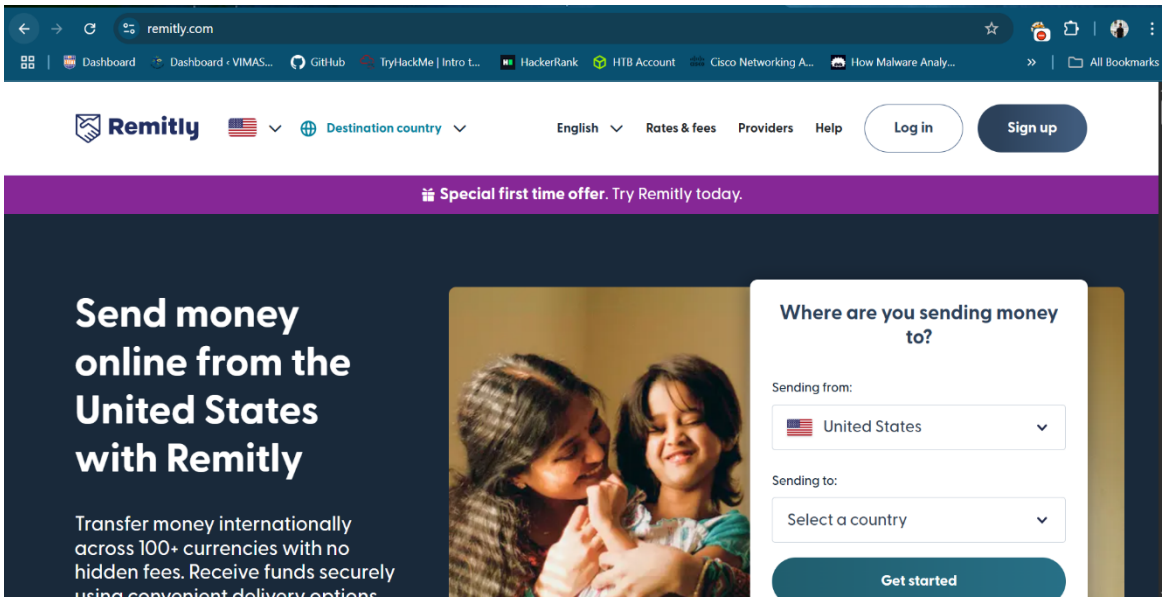
Missing Secure, HttpOnly, SameSite Flags on Cookies

Vulnerability Description:

I found this program on the hackerone Bug hunting website. The website hosted at https://www.remitly.com. Session cookies are critical for maintaining authenticated sessions between a user's browser and a web server. If cookies lack the **Secure**, **HttpOnly**, or **SameSite** attributes, they are exposed to multiple security risks:

- **Without Secure:** Cookies can be transmitted over unencrypted HTTP connections, making them vulnerable to interception via Man-in-the-Middle (MITM) attacks.
- **Without HttpOnly:** Cookies can be accessed via JavaScript, increasing the risk of theft through Cross-Site Scripting (XSS) attacks.
- **Without SameSite:** Cookies may be sent in cross-site requests, allowing Cross-Site Request Forgery (CSRF) attacks.

Proper configuration of these cookie attributes is essential to protect user sessions from interception and exploitation.



Affected Components:

- **URL**: https://www.remitly.com
- **Cookies Affected:**
    - cookie_consent
    - de_id
    - de_hash
    - pm_sub
    - _policy
    - gr
    - ci_csrf_input
    - HTTP Methods Observed: GET, OPTIONS

Impact Assessment:

- **Risk Level:** Medium

Failure to set Secure, HttpOnly, and SameSite flags can allow attackers to:

- Steal session cookies via JavaScript (if XSS vulnerabilities exist)
- Hijack user sessions through network sniffing on insecure connections
- Launch CSRF attacks to perform unauthorized actions on behalf of the user

If a malicious actor obtains a session cookie, they could impersonate the user and gain unauthorized access to their account or sensitive operations.

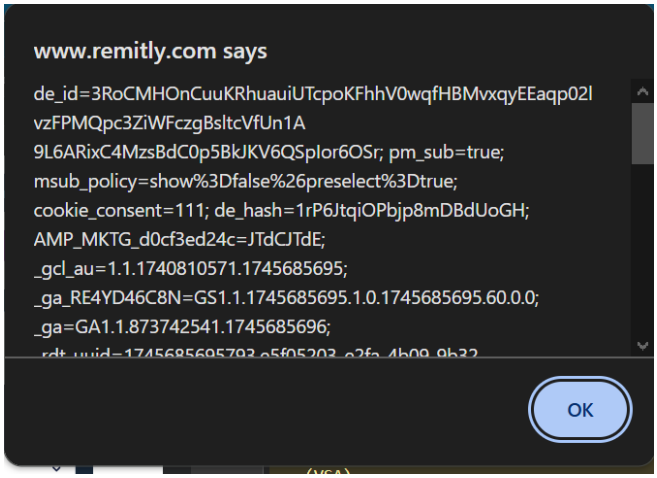
Steps to Reproduce:

1. Nikto Scan Results
    - Ran Nikto with tuning options to identify web vulnerabilities.
    - Discovered multiple cookies set without Secure and HttpOnly flags.

2. Manual XSS Validation Using JavaScript
    - Injected and executed:

        alert(document.cookie);

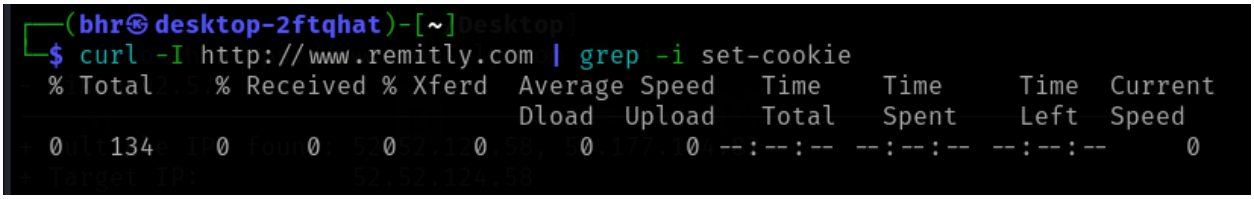
    - Verified that cookies like cookie_consent and gr were accessible via JavaScript



3. Header Inspection via Curl
    - Inspected response headers:

        curl -I https://www.remitly.com | grep -i set-cookie

    - Confirmed that cookies were missing Secure, HttpOnly, and SameSite flags in response.



4. Strict-Transport-Security (HSTS) Check
    - Verified that HSTS is set:

        strict-transport-security: max-age=31536000

    - HTTPS is enforced, but cookies still need Secure/HttpOnly/SameSite individually for full protection.

Proof of Concept (PoC):

1. Captured Cookies without Security Flags:

   Set-Cookie: cookie_consent=value; path=/;

   Set-Cookie: de_id=value; path=/;

   Set-Cookie: gr=value; path=/;

2. Manual JavaScript Execution:

   alert(document.cookie);

3. Observed Behavior:
   - Cookies were readable via JavaScript, confirming that **HttpOnly** flag is missing.
   - Cookies could theoretically be stolen if an XSS vulnerability existed.
   - Secure flag missing on some cookies allows possible interception under certain conditions if not properly encrypted.

Proposed Mitigation or Fix:

To address the missing cookie security attributes:

1. Set the Secure Flag:
   Ensure that all cookies are set with the Secure attribute, so they are transmitted only over HTTPS:
   Set-Cookie: cookie_name=value; Secure

2. Set the HttpOnly Flag:
   Prevent client-side scripts from accessing cookies by adding HttpOnly:
   Set-Cookie: cookie_name=value; HttpOnly

3. Set the SameSite Attribute:
   Mitigate CSRF attacks by using SameSite=Lax or SameSite=Strict:
   Set-Cookie: cookie_name=value; SameSite=Lax

4. Conduct Regular Security Reviews:
   Regularly audit cookie settings in all web applications to ensure security best practices are applied across environments.

5. Update Application Configuration:
   Ensure that the server-side cookie setting mechanisms (application code, web server config) apply these attributes by default.

6. Apply Secure Defaults:
   Frameworks or load balancers should enforce Secure and HttpOnly settings wherever possible.

7. Use environment-based secrets handling systems such as Vault, AWS Secrets Manager, or .env files outside of the building process.

Conclusion:

A few cookies set by https://www.remitly.com are missing important security attributes (Secure, HttpOnly, and SameSite). Manual testing confirmed that cookies were accessible via client-side scripts, exposing users to elevated risks like session hijacking and CSRF. Although HTTPS (HSTS) is enforced, cookies must be individually secured to prevent client-side exploitation.