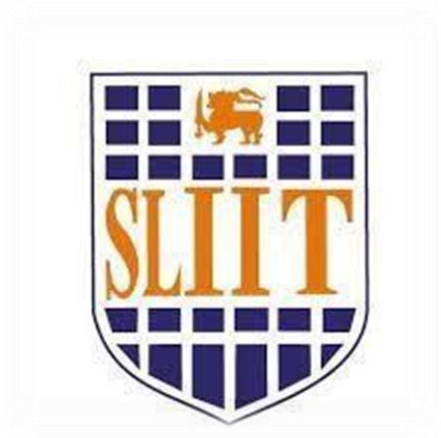
SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY




# Web Security- IE2062

# Cookie Not Marked as HttpOnly

Danuka Nuwan

IT22349842

**Title – Cookie Not Marked as HttpOnly**

**Vulnerability**: The application stops working to note particular cookies as HttpOnly enabling client-side manuscripts to access them via document. Cookie or various other methods. This subjects the cookies to possible burglary by means of cross-site scripting (XSS) assaults, allowing enemies to swipe session symbols or delicate info kept in cookies.

**Websute-**www.Kindredgroup.com

**Affected Components**:

- Web server configurations
- Session management
- Authentication mechanisms

**Impact Assessment:**

Medium: Failure to note cookies as HttpOnly raises the threat of session hijacking together with information exfiltration with XSS strikes. While the straight influence might differ based upon the level of sensitivity of the info kept in cookies the possibility for unapproved gain access to or information burglary stays substantial.

**Instructions to Replicate:**

Perform a safety analysis of the application's cookies utilizing internet browser designer devices or proxy devices to examine HTTP actions.

Determine cookies that are not significant as HttpOnly by analyzing their characteristics in the Set-Cookie header.

Effort to manipulate XSS vulnerabilities within the application to show the capacity to accessibility HttpOnly cookies with client-side manuscripts.

Examine the effect of unwarranted accessibility to HttpOnly cookies on session stability, individual verification, or delicate information direct exposure.

**Proof of Concept:**

# 4. Cookie Not Marked as HttpOnly

LOW 1 CONFIRMED 1

Netsparker identified a cookie not marked as HTTPOnly.

HTTPOnly cookies cannot be read by client-side scripts, therefore marking a cookie as HTTPOnly can provide an additional layer of protection against cross-site scripting attacks.

### Impact

During a cross-site scripting attack, an attacker might easily access cookies and hijack the victim's session.

### Vulnerabilities

#### 4.1. https://www.kindredgroup.com/

**CONFIRMED**

**Identified Cookie(s)**

- EPiStateMarker

**Cookie Source**

- HTTP Header

**Request**

```
GET / HTTP/1.1
Host: www.kindredgroup.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

**Response**

Response Time (ms) : 515.5052 Total Bytes Received : 92412 Body Length : 91711 Is Compressed : No

```
HTTP/1.1 200 OK
Set-Cookie: EPiStateMarker=true; path=/
Set-Cookie: ARRAffinity=f62dc48792b6e16789f38b9331562ab71aac9ec805fac06e15282a091076b114;Path=/;HttpOnl
y;Secure;Domain=www.kindredgroup.com
Set-Cookie: ARRAffinitySameSite=f62dc48792b6e16789f38b9331562ab71aac9ec805fac06e15282a091076b114;Path
=/;HttpOnly;SameSite=None;Secure;Domain=www.kindredgroup.com
Request-Context: appId=cid-v1:f6f0367f-9d11-44c1-bd87-a01942e8c453
CF-RAY: 87b1cb230fc9513a-CMB
Server: cloudflare
CF-Cache-Status: DYNAMIC
Connection: keep-alive
Content-Encoding:
Strict-Transport-Security: max-age=2592000
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chuHTTP/1.1 200 OK
Set-Cookie: EPiStateMarker=true; path=/

Set-Cookie: ARRAffinity=f62dc48792b6e16789f38b9331562ab71aac9ec805fac06e15282a091076b114;Path=/;HttpOnl
y;Secure;Domain=www.kindredgroup.com
Set-Cookie: ARRAffinitySameSite=f62dc48792b6e16789f38b9331
…
```

**Actions to Take**

1. See the remedy for solution.
2. Consider marking all of the cookies used by the application as HTTPOnly. (*After these changes javascript code will not be able to read cookies.*)

**Remedy**

Mark the cookie as HTTPOnly. This will be an extra layer of defense against XSS. However this is not a silver bullet and will not protect the system against cross-site scripting attacks. An attacker can use a tool such as XSS Tunnelto bypass HTTPOnly protection.

**External References**

- Netsparker - Security Cookies - HTTPOnly Flag
- OWASP HTTPOnly Cookies
- MSDN - ASP.NET HTTPOnly Cookies



**CLASSIFICATION**

| | |
|---|---|
| OWASP 2013 | A5 |
| OWASP 2017 | A6 |
| SANS Top 25 | 16 |
| CAPEC | 107 |
| WASC | 15 |
| ISO27001 | A.14.2.5 |

**Suggested Resolution:**

Update internet server setups or application code to note cookies as HttpOnly to avoid client-side accessibility by JavaScript.

Carry out safe coding techniques to minimize XSS vulnerabilities, such as input recognition, result inscribing and also correct sanitation of user-generated material.

Use safety headers, such as HttpOnly plus Secure flags, to apply cookie defense and also make sure cookies are transmitted just over encrypted links.

Perform routine safety and security analyses consisting of code testimonials along with vulnerabilities scanning to determine as well as remediate cookie-related safety and security weak points.

Inform programmers on the value of marking cookies as HttpOnly as well as the dangers connected with XSS vulnerabilities nurturing a security-aware growth society within the company.

Display plus log questionable tasks associated with cookie adjustment or unapproved accessibility making it possible for prompt discovery coupled with action to possible safety and security cases.