How to Secure HTTP Cookies

https://campus.barracuda.com/doc/4259854/

Overview

Cookies provide a mechanism to store session state information on client navigation platforms such as browsers and other user agents. Cookies can store user preferences or shopping cart items, and can include sensitive information like registration or login credentials. If a cookie can be modified, the system can become vulnerable to attacks and sensitive information can be stolen.

How Cookie Security Works

The Barracuda Web Application Firewall cookie security is transparent to backend servers. When a server inserts a cookie, the Barracuda Web Application Firewall intercepts the response and encrypts or signs the cookie before delivering it to the client. When a subsequent request from the client returns this cookie, the Barracuda Web Application Firewall intercepts the request and decrypts it or verifies its signature. If the cookie is unaltered, the Barracuda Web Application Firewall forwards the original cookie to the server. Altered cookies are removed before the Barracuda Web Application Firewall forwards the request to the server.

Encryption prevents both viewing and tampering with cookies, so it prevents the client from accessing cookie values. For clients who need to access cookie values, use signing to allow it. When signing cookies, the Barracuda Web Application Firewall actually forwards two cookies to the client browser, one plain text cookie and one signed cookie. When a subsequent request from the client returns the cookies, if *either* cookie is altered, signature verification fails, and the Barracuda Web Application Firewall removes cookies before forwarding the request to the server.

Cookie Security Interaction with Other Security Features

Encrypting a cookie may change the length of the cookie, but the number of headers in the message remains unchanged. When a cookie is signed, the length of the cookie changes and one or more headers is appended to the forwarded message. If the **SECURITY POLICIES > Request Limits** configuration specifies constraints on the number or length of HTTP headers, a signed or encrypted cookie may violate the request limits and result in unwanted rejection of messages. Messages thus rejected are logged as *Cloak* under **Action** on the **BASIC > Web Firewall Logs** page.

Configuring Cookie Security

To encrypt or sign cookies and reject tampered cookies, you need to enable cookie security using the following steps:

1. Go to the **SECURITY POLICIES > Cookie Security** page.
2. Select a policy from the **Policy Name** list.
3. In the **Cookie Security** section, select the desired **Tamper Proof Mode**, either *Encrypted* or *Signed*. Recommended: Signed. Note: Encrypting cookie data makes cookie values inaccessible to the client. If required, change values of other parameter(s):
    - **Cookie Max Age** – Enter the maximum age in minutes for session cookies after which cookies are automatically expired.
        - **Range**: 0 to 500000
        - **Recommended**: 1440
        - **Units**: Minutes
    - **Cookie Replay Protection Type** – Select the type of protection to be used to prevent cookie replay attacks from the drop-down list.
        - **Values**: Custom Headers, IP, IP and Custom Headers, None
        - **Recommended**: IP
    - **Custom Headers** – Enter the custom headers to be used in the cookie if the parameter **Cookie Replay Protection Type** is set to **Custom Headers** or **IP and Custom Headers** and click **Add**.
    - **Secure Cookie** – Set to **Yes** to allow cookies to be returned to the web server if the client makes secure HTTPS connection only.
        - **Values**: Yes, No
        - **Recommended**: No
    - **HTTP Only** – Set to **Yes** to allow cookies to be returned to the web server only if the client makes an HTTP connection. When a client-side script attempts to read a cookie with **HTTP Only** enabled, the browser returns an empty string as the result. Enabling **HTTP Only** prevents the cookie from being accessed through client-side scripts.
        - **Values**: Yes, No
        - **Recommended**: No
    - **Same Site** When the value is set, the attribute of the Set-Cookie HTTP response header allows you to declare if your cookie should be restricted to a first-party or same-site context.
        - **Values**: Do not Add, Lax, Strict and None
        - **Values**: Do not Add, Lax, Strict and None
    - **Allow Unrecognized Cookies** – Select whether unrecognized cookies should be allowed. Use **Custom** to temporarily allow unrecognized cookies after deploying. Use **Days Allowed** to indicate for how long unrecognized cookies are allowed.
        - **Values** - Always, Never, Custom
        - **Default** - Custom
    - **Days Allowed** – Enter the number of days unrecognized cookies will not be rejected. This

field is used only when **Allow Unrecognized Cookies** is **Custom**.
- **Cookies Exempted** – Add the names of cookies to exempt from the cookie security policy.

1. Click **Save**.

Securing Barracuda WAF Generated Internal Cookies:

Internal cookies are not vulnerable as they are *Encrypted* and are never passed to the backend servers. Although the vulnerability assessment tools may report these cookies as not secure because of the absence of attributes like HTTP ONLY or Secure, these are always false positives and can be safely ignored.

Related Articles

Cookie Tampering Attacks Logged When Barracuda Web Application Firewall Is Initially Deployed