To address the following security vulnerabilities in **IBM HTTP Server (IHS)** — which is based on Apache — you need to make specific configuration changes in its config files (usually httpd.conf or related includes). Here's how you can tackle each of these:

**🔐 1. Lack of Rate Limiting**

**Problem**: IHS does not limit how often a client can make requests — this can lead to DoS attacks or brute-force attacks.

**✅ Solution: Use mod\_ratelimit or mod\_evasive**

IBM HTTP Server supports Apache modules like:

**Option A: mod\_ratelimit**

Add this inside your <VirtualHost> or global config:

<IfModule mod\_ratelimit.c>

SetOutputFilter RATE\_LIMIT

SetEnv rate-limit 400

</IfModule>

* rate-limit 400 = limits response to 400 KB/s per client.

✅ Install via IBM HTTP Server Plugin if not already available.

**Option B: mod\_evasive (Better for DoS/Brute-Force Protection)**

Enable it by adding:

LoadModule evasive20\_module modules/mod\_evasive20.so

<IfModule mod\_evasive20.c>

DOSHashTableSize 3097

DOSPageCount 2

DOSSiteCount 50

DOSPageInterval 1

DOSSiteInterval 1

DOSBlockingPeriod 10

DOSEmailNotify you@example.com

DOSLogDir /var/log/httpd/evasive

</IfModule>

🔍 This will block IPs with too many requests in a short time and notify admin.

**🍪 2. Secure Cookie Attribute Not Set**

**Problem**: Cookies are not marked as Secure, so they can be sent over HTTP instead of only HTTPS.

**✅ Solution:**

You need to update how cookies are set. If they are set via IHS (via mod\_headers), do this:

Header always edit Set-Cookie ^(.\*)$ $1;Secure;HttpOnly

* Secure ensures cookie is sent only over HTTPS.
* HttpOnly prevents JS from accessing the cookie.

**⚠️ If cookies are set by backend (like WebSphere), you may need to:**

* Set Secure cookies via WebSphere admin console.
* Or configure IHS to edit the header as above.

**🌐 3. Broadly Scoped Session Cookie Domain**

**Problem**: Cookies scoped to top-level domains (.example.com) can be accessed across subdomains (e.g., admin.example.com and shop.example.com).

**✅ Solution:**

Set cookie domain as tightly as possible in your application.

If you **must** fix this at IHS level, you can override the Set-Cookie header using mod\_headers:

Header always edit Set-Cookie "Domain=\.example\.com" "Domain=sub.example.com"

This forces the cookie to be scoped only to sub.example.com.

**✅ Summary Table**

| **Vulnerability** | **Fix Approach** | **Module Needed** |
| --- | --- | --- |
| Lack of Rate Limiting | mod\_ratelimit or mod\_evasive | mod\_ratelimit / mod\_evasive |
| Secure Cookie Attribute Not Set | Use mod\_headers to add Secure | mod\_headers |
| Broadly Scoped Session Cookie Domain | Use mod\_headers to narrow domain | mod\_headers |

Let me know if you're using **WebSphere behind IHS**, or want specific directives for your environment. I can help you create a full httpd.conf snippet too.