How to Enable Secure HTTP Header in Apache Tomcat 8?

By **Chandan Kumar** / in **Security (https://geekflare.com/category/security/)**,**Tomcat/Weblogic (https://geekflare.com/category/infrastructure/tomcat/)** / Updated: February 18, 2018

Injecting HTTP Response with the secure header can mitigate most of the web security vulnerabilities (https://geekflare.com/saas-web-vulnerability-scanner/).

If you are managing production environment or payment related application, then you will also be asked by security/penetration testing team to implement necessary HTTP header to comply with PCI-DSS security standard.

Having secure header instruct browser to do or not to do certain things to **prevent certain security attack**.

Most of you might be using a web server like Apache, Nginx, IIS in front of Tomcat so you may implement the headers directly in web server (https://geekflare.com/http-header-implementation/).

However, if you don't have any web server in front or need to implement directly in Tomcat then **good news** if you are using Tomcat 8.

Tomcat 8 has added support for following HTTP response headers.

- X-Frame-Options – to prevent clickjacking attack
- X-XSS-Protection – to avoid cross-site scripting attack
- X-Content-Type-Options – block content type sniffing
- HSTS – add strict transport security

I've tested with **Apache Tomcat 8.5.15** on Digital Ocean (https://www.digitalocean.com/?refcode=c278bf0364c1) Linux (CentOS distro) server.

Note: If you are looking for overall hardening & security then you may refer this guide (https://geekflare.com/apache-tomcat-hardening-and-security-guide/).

As a best practice, **take a backup** of necessary configuration file before making changes or test in a non-production environment.

- Login to Tomcat server
- Go to the conf folder under path where Tomcat is installed
- Uncomment the following filter (by default it's commented)

```
<filter>
  <filter-name>httpHeaderSecurity</filter-name>
  <filter-class>org.apache.catalina.filters.HttpHeaderSecurityFilter</filter-class>
  <async-supported>true</async-supported>
</filter>
```

By uncommenting above, you instruct Tomcat to support HTTP Header Security filter.

- Add the following just after the above filter

```
<filter-mapping>
  <filter-name>httpHeaderSecurity</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

By adding above you instruct Tomcat to inject the HTTP Header in all the application URL.

- Restart the Tomcat and access the application to verify the headers.

You may use an online tool to verify the header (https://siterelic.com/tools/http-header-test) or use F12 on a browser to inspect.

Here is quick filter reference taken from a web.xml file.

```
<!-- ==================== Built In Filter Definitions ===================
 <!-- A filter that sets various security related HTTP Response headers.  -->
  <!-- This filter supports the following initialization parameters       -->
  <!-- (default values are in square brackets):                          -->
  <!--                                                                   -->
  <!--  hstsEnabled       Should the HTTP Strict Transport Security     -->
  <!--                    (HSTS) header be added to the response? See    -->
  <!--                    RFC 6797 for more information on HSTS. [true] -->
  <!--                                                                   -->
  <!--  hstsMaxAgeSeconds  The max age value that should be used in the  -->
  <!--                    HSTS header. Negative values will be treated   -->
  <!--                    as zero. [0]                                   -->
  <!--                                                                   -->
  <!--  hstsIncludeSubDomains                                           -->
  <!--                    Should the includeSubDomains parameter be      -->
  <!--                    included in the HSTS header.                   -->
  <!--                                                                   -->
  <!--  antiClickJackingEnabled                                         -->
  <!--                    Should the anti click-jacking header           -->
  <!--                    X-Frame-Options be added to every response?    -->
  <!--                    [true]                                         -->
  <!--                                                                   -->
  <!--  antiClickJackingOption                                          -->
  <!--                    What value should be used for the header. Must -->
  <!--                    be one of DENY, SAMEORIGIN, ALLOW-FROM         -->
  <!--                    (case-insensitive). [DENY]                     -->
  <!--                                                                   -->
  <!--  antiClickJackingUri IF ALLOW-FROM is used, what URI should be    -->
  <!--                    allowed? []                                    -->
  <!--                                                                   -->
  <!--  blockContentTypeSniffingEnabled                                 -->
  <!--                    Should the header that blocks content type     -->
  <!--                    sniffing be added to every response? [true]    -->
```

Enabling secure header in Tomcat 8 is straightforward, and as an administrator, you should plan to implement them for better security.

If you are new to Tomcat, you may be interested in taking this Apache Tomcat administration course (https://click.linksynergy.com/deeplink?id=jf7w44yEft4&mid=39197&murl=https%3A%2F%2Fwww.udemy.com%2Fapache-tomcat-for-beginners-and-advanced%2F).

About Chandan

Chandan Kumar is the founder and editor of Geek Flare. Learn more here (https://geekflare.com/about/) and connect with him on Twitter.

(https://twitter.com/ConnectCK)

You'll also like:

(https://geekflare.com/html5-changing-web-security/)

How HTML5 is Changing Web Security? (https://geekflare.com/html5-changing-web-security/)

(https://geekflare.com/tomcat-ssl-guide/)

How to Implement SSL in Apache Tomcat? (https://geekflare.com/tomcat-ssl-guide/)