# Cookie Not Sent Over SSL Development Mitigation SOP

Sending cookies over an unencrypted channel can expose them to network sniffing attacks, so the secure flag helps keep a cookie's value confidential.

## Defense Against Cookie Not Sent Over SSL

Modern web browsers support a secure flag for each cookie. If the flag is set, the browser will only send the cookie over HTTPS. This is especially important if the cookie contains private data or carries a session identifier.

## Example

public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, FilterChain filterChain) throws IOException, ServletException {

String csrf = (String)((HttpServletRequest) servletRequest).getSession().getAttribute("OWASP\_CSRFTOKEN");

if (csrf != null) {

Cookie cookie = WebUtils.getCookie((HttpServletRequest) servletRequest, "XSRF-TOKEN");

if (cookie == null || !csrf.equals(cookie.getValue())) {

cookie = new Cookie("XSRF-TOKEN", csrf);

cookie.setPath("/");

((HttpServletResponse)servletResponse).addCookie(cookie);

}

}

filterChain.doFilter(servletRequest, servletResponse);

}

## Explanation

In this case a cookie is created with the method call cookie = new Cookie("XSRF-TOKEN", csrf), but the setSecure() method is never called.

## Recommendation

Set the Secure flag on all new cookies in order to instruct browsers not to send these cookies in the clear by calling setSecure(true).

## References

1. [Class Cookie, Sun Microsystems](http://java.sun.com/j2ee/sdk_1.3/techdocs/api/javax/servlet/http/Cookie.html#setSecure(boolean))
2. [Mike Perry, Automated HTTPS Cookie Hijacking](http://fscked.org/blog/fully-automated-active-https-cookie-hijacking)
3. Standards Mapping - Common Weakness Enumeration - (CWE), CWE ID 614
4. Standards Mapping - FIPS200 - (FISMA), CM, SC
5. Standards Mapping - NIST Special Publication 800-53 Revision 4 - (NIST SP 800-53 Rev.4), SC-8 Transmission Confidentiality and Integrity (P1)