**Code Correctness: Multiple Stream Commits**

After a servlet's output stream has already been committed, it is erroneous to reset the stream buffer or perform any other action that recommits to the stream. Likewise, it is erroneous to call getWriter() after calling getOutputStream or vice versa.

**Explanation**

Forwarding an HttpServletRequest, redirecting an HttpServletResponse, or flushing the servlet's output stream buffer causes the associated stream to commit. Any subsequent buffer resets or stream commits, such as additional flushes or redirects, will result in IllegalStateExceptions.

Furthermore, Java servlets allow data to be written to the response stream using either ServletOutputStream or PrintWriter, but not both. Calling getWriter() after having called getOutputStream(), or vice versa, will also cause an IllegalStateException.

At runtime, an IllegalStateException prevents the response handler from running to completion, effectively dropping the response. This can cause server instability, which is a sign of an improperly implemented servlet.

**Example**

The following code redirects the servlet response after its output stream buffer has been flushed.

public class RedirectServlet extends HttpServlet {

public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {

...

OutputStream out = res.getOutputStream();

...

// flushes, and thereby commits, the output stream

out.flush();

out.close(); // redirecting the response causes an IllegalStateException

res.sendRedirect("http://www.acme.com");

}

}

**Example**

Conversely, the following code attempts to write to and flush the PrintWriter's buffer after the request has been forwarded.

public class FlushServlet extends HttpServlet {

public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {

...

// forwards the request, implicitly committing the stream

getServletConfig().getServletContext().getRequestDispatcher("/jsp/boom.jsp").forward(req, res);

...

// IllegalStateException; cannot redirect after forwarding

res.sendRedirect("http://www.acme.com/jsp/boomboom.jsp");

PrintWriter out = res.getWriter();

// writing to an already-committed stream will not cause an exception,

// but will not apply these changes to the final output, either

out.print("Writing here does nothing");

// IllegalStateException; cannot flush a response's buffer after forwarding the request

out.flush();

out.close();

}

}

**Recommendations**

Ensure that no further changes to the stream are made after it is committed. At best, any further writes to the stream are ineffectual, and any additional commits

can cause the servlet to throw an IllegalStateException. It is generally good practice to observe these guidelines when possible:

1. Return immediately after a forward() or sendRedirect.

2. Avoid calling forward() or sendRedirect() after calling ServletResponse.getWriter() or ServletResponse.getOuputStream().

**Reference**

* http://www.hpenterprisesecurity.com/vulncat/en/vulncat/java/code\_correctness\_multiple\_stream\_commits.html