**On HTTP Parameter Pollution Attacks in Java Servlets**

Tommy Cusick (tommyc@google.com)

June 29, 2012

## Background

*Data from the query string and the post body are aggregated into the request parameter set. Query string data is presented before post body data. For example, if a request is made with a query string of a=hello and a post body of a=goodbye&a=world, the resulting parameter set would be ordered a=(hello, goodbye, world).*

(Java Servlet Specification, Version 3.0, Section 3.1: HTTP Protocol Parameters)

## Overview

The default behavior of servlets which implement the spec given above can be prone to exploitation by malicious agents, wherein query parameter values are used when the servlet expected the input to come from the request body.

For example, imagine a user navigates to a web page `/Foo` that presents the user with a form:

```
...
<form method="post">
  <input type="text" name="name" />
  <input type="submit" name="submit" value="Submit" />
</form>
...
```

The user types "Bob" as his name and submits it. The servlet then processes the request:

```
...
protected void service(HttpServletRequest req, HttpServletResponse res) {
  String name = req.getParameter("name");  // contains "Bob"
  if (name != null) {
    // Does something with |name|
    ...
  }
  ...
}
...
```

Now, imagine that an attacker sends the user a crafted link to `/Foo?name=Jim`. The user clicks the link and sees the same form, fills it out, and submits it. This time, though, the return value of `req.getParameter("name")` will be "Jim" and not "Bob", ignoring the explicit user input in the request body! (In a worse case, imagine this as a change password form for a large site.)

This behavior occurs when a form specifies no action – the user's browser will use the current URL as the form action, which means that the attacker's value is propagated and preferred by

the servlet.  Furthermore, any page which programmatically propogates query parameters from the current page to the form action is also vulnerable.

Prevention of this attack requires extreme diligence (every form must have an explicit action specified and must be sure to only propagate query parameters which are not present in the form) if one's servlet environment conforms to the given specification.