HTTP methods determined NOT to be constrained during the translation of servlet security-constraints. 4) The use of exception lists causes the permissions resulting from the translation of a given security-constraint configuration to differ in their actions values from those that would have been produced prior to support for HTTP extension methods. Previously translated permissions remain supported by the changed permission implementations, and (with the exceptions listed in 1 and 2 above) continue to function as they did before the change, as long as extension methods are not set in checked permissions.

B.21 Welcome File and security-constraint Processing

The relationship between welcome file processing (which can modify the effective request URI) and security-constraint processing is not defined by the Servlet Specification. Since this specification uses url-patterns derived from request URIs to name target resources in checked permissions, it is important that welcome file processing and its relationship to security-constraint processing be clearly specified. Without a clear description of this relationship, unprotected request URIs which are modified to yield effective request URIs for protected resources may inadvertently be left unprotected.

Resolution– pending Servlet clarification. Recommend that Servlet standarize an HttpServletRequest attribute that can be used to portably obtain the requestURI following welcome file mapping. Once this attribute is standardized, The HttPservletRequest based constructors of WebResourcePermission and WebUserDataPermission would use its value to establish the permission name.

B.22 Colons Within path-segment of Request URI

As defined in IETF RFC 2396 "Uniform Resource Identifiers (URI): Generic Syntax", the abs_path component of a request URI may consist of a sequence of "/" separated path segments, where the format of each segment is defined as follows:

```
segment = *pchar *( ";" param )
param   = *pchar
pchar   = unreserved | escaped |":" | "@" | "&" | "=" | "+" | "$" | ","
```

A colon character occurring within a path-seqment will be syntactically indistinguishable from colons used by the WebResourcePermission and WebUserDataPermission constructors to demarcate qualifying patterns.

Resolution– Require that containers use escaped encoding (as defined in RFC 2396) on colon characters occuring within url-patterns obtained from web.xml. Also require that containers encode colons occuring within patterns extracted from HttpServletRequest objects and used to create the names of checked WebResourcePermission and WebUserDataPermission objects. Also require the the HttpServletRequest based constructors of WebResourcePermission and WebUserDataPermission apply escaped encoding to colons occuring in the names the derived from the request URI. Note that the colon character is represented as %3A in escaped encoding.
