

Q54 - Servlet Security [6]

Considering the following security constraints, which statements are true:

```
@ServletSecurity(value = @HttpConstraint(
    value = EmptyRoleSemantic.DENY,
    transportGuarantee = TransportGuarantee.CONFIDENTIAL),
    httpMethodConstraints = {
        @HttpMethodConstraint(
            value = "GET",
            rolesAllowed = "manager"),
        @HttpMethodConstraint(
            value = "POST",
            rolesAllowed = "*",
            emptyRoleSemantic = EmptyRoleSemantic.PERMIT)
    }
)
@WebServlet("/foo")
public class MyServlet extends HttpServlet {
}
```

- a. For all HTTP methods other than GET and POST, and for all roles, access to this servlet is forbidden and the confidential transport is required
- b. For all HTTP methods and for all roles, access to this servlet is forbidden and the confidential transport is required
- c. For GET HTTP method request, the access to this servlet is forbidden if the role is different than "manager"
- d. For GET HTTP method request, the access to this servlet is forbidden if the role is equal to "manager"
- e. For POST HTTP method request, the access to this servlet is forbidden if the role is different than "manager"
- f. For POST HTTP method request, the access to this servlet is allowed with no further roles restrictions

Hide answer

a, c

Reference: pages 123 - 126, 13.4 "Programmatic Access Control Annotations"

Explanation: The `value` attribute defines the constraint which affects the HTTP methods which are **not** listed using the `httpMethodConstraints` attribute.

`EmptyRoleSemantic` defines the default action which should be undertaken if the `rolesAllowed` returns an **empty list** (no roles are defined). As a side not, you **should not** define `emptyRoleSemantic` when `rolesAllowed` list is not empty.

The `"*"` as a value of `rolesAllowed` attribute **doesn't have any special meaning**. In the DD, when `<role-name></role-name>` is used it means "all roles", but in the `@ServletSecurity` annotation -- it doesn't.