Q52 - Servlet Security [4]

Consider the following Servlet code:

```
package com.nullhaus;

import javax.servlet.annotation.ServletSecurity.*;
import javax.servlet.annotation.*;
import javax.servlet.http.*;
import java.io.*;

@ServletSecurity(value = @HttpConstraint(EmptyRoleSemantic.DENY),
        httpMethodConstraints = {@HttpMethodConstraint(value = "GET", emptyRoleSemantic = EmptyRoleSemantic.PERMIT)})

@WebServlet(value = "/foo/*", name = "NullServlet")
public class NullServlet extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException {
        resp.getWriter().print("Howdy Stragers!");
    }
}
```

Choose statements which are true about the servlet's HTTP GET request:

a. This servlet is accessible for all users

b. This servlet is not accessible for any users

c. The `EmptyRoleSemantic.DENY` is not a valid `@HttpConstraint` attribute value

d. A runtime exception will be thrown while trying to access the servlet

e. The above code doesn't compile

Hide answer

a

Explanation: This is the correct usage of the annotations. The servlet is **allowed only for HTTP GET** requests and **blocked for all other HTTP methods** requests despite the user's role.