The language used in the provided code is JavaScript.

The primary security issue with this code is a classic **Cross-Site Scripting (XSS)** vulnerability:

1. **Reflected Cross-Site Scripting (XSS)**:

   - The code is taking a parameter (`color`) directly from the URL and then writing it into the document using `document.write()`. This means that if an attacker crafts a malicious URL with a script payload instead of a color value, they can execute arbitrary JavaScript on the victim's browser when the URL is visited.

   For example, an attacker could craft a URL like:

   ```

   http://example.com/page?color=;alert(document.cookie)//

   ```

   When a victim visits this URL, the attacker's script (`alert(document.cookie)`) would be executed, potentially revealing the victim's cookie or allowing other malicious actions.

2. **Deprecation of `document.write`**:

   - While not strictly a security concern, using `document.write` is generally frowned upon in modern web development. It can cause performance issues, especially when used in scripts that are asynchronously loaded, and can make code harder to maintain or debug.

To mitigate the XSS vulnerability:

1. **Avoid Inline Scripting**:

   - Instead of using `document.write` to inject styles or scripts, manipulate the DOM using methods like `document.createElement` and `element.appendChild`. This helps avoid directly executing or rendering user-controlled data as code.

2. **Validate and Sanitize Input**:

   - Any data that comes from an untrusted source, including URLs, should be treated as potentially malicious. Validate that the data is in the expected format (e.g., a valid color code) and sanitize it to remove or neutralize any potential code.

3. **Content Security Policy (CSP)**:

   - Implement a Content Security Policy header to restrict where scripts can be loaded from and what inline scripts can do. This adds another layer of defense against XSS attacks.

By following these recommendations, you can help protect your application and its users from potential security threats.