

# Tutorial: Securing Older JavaScript Libraries

## What is a JavaScript Library?

JavaScript is a widely-used and versatile programming language for building websites and applications. A JavaScript library is a collection of pre-written JavaScript code that developers can use to streamline their work, enabling faster development by reusing functions and components instead of writing code from scratch.

## Potential Risks of Older JavaScript Libraries

Using outdated or retired JavaScript libraries introduces several security vulnerabilities, as older versions often contain known flaws that hackers can exploit.

### Common risks include:

- **Vulnerability Exposure:** Over time, the likelihood increases that hackers will discover and exploit weaknesses in older libraries.
- **Data Breaches:** Attackers can manipulate, steal, or corrupt data once they gain access.
- **System Compromise:** Hackers may take control of user accounts or the entire system.

### Common Types of Cybersecurity Attacks:

- **Cross-Site Scripting (XSS):** Injects malicious scripts into web pages viewed by other users.
- **Man-in-the-Middle Attacks:** Intercepts communications between two parties to steal sensitive information.
- **JavaScript Prototype Poisoning:** Exploits JavaScript object prototypes to compromise application behavior.

## How to Mitigate Security Risks in JavaScript Libraries

The most effective way to secure a system is to update to the latest version of the JavaScript library. Follow the steps below to ensure a secure and up-to-date library.

### Updating JavaScript Libraries

#### Two options for updating JavaScript libraries:

1. Specify Parameters Interactively.
2. Use a Pre-Generated Definition File.

### Steps to Update

1. **Navigate to Update Options:**
  - Select the Navigate option.
  - Choose Impact Manager from the menu.
  - Select JavaScript Update.
2. **Set Control Mode:**

- Ensure that Control Mode is selected to manage the update process.
- 3. **Provide a Task Description:**
  - Enter a description of the update task.
  - Confirm that the description is displayed under Show Task Status.
- 4. **Choose an Update Method:**
  - **Option 1: Specify Parameters Interactively**
    - Select Specify Parameters Interactively.
    - Click Next to proceed.
    - Follow the prompts to input the required parameters.
  - **Option 2: Use a Pre-Generated Definition File**
    - Select Browse to locate your pre-generated definition file.
    - Input parameters as parameter1="value1", parameter2="value2", with one parameter per line.
    - Click Next to continue.
- 5. **Select Files to Update:**
  - Browse and select the files to be updated.

If the necessary files are not listed, locate JavaScriptUpdateConfig\_dds.js in the following directory: Administration\Impact Manager\Script Repository

  - Navigate to the appropriate section, select the correct files, and click OK.
- 6. **Schedule or Perform the Update:**
  - Choose to execute the update immediately or schedule it for a later time.
  - Verify that the update status displays Success upon completion.
- 7. **Validate the Update:**
  - Confirm the updated release version to ensure the library is current.

## Additional Tips for Secure JavaScript Library Management

- **Regular Updates:** Schedule routine checks and updates for libraries to maintain system security.
- **Monitor Vulnerabilities:** Stay informed about known vulnerabilities in your libraries using platforms like CVE or other vulnerability databases.
- **Use Content Delivery Networks (CDNs):** CDNs often host the latest secure versions of popular libraries, reducing the risk of outdated files.

By following these steps and practices, users significantly reduce the security risks associated with older JavaScript libraries.