

# Web Application Vulnerability Report

Scan Date: 2025-08-01 09:09:38 UTC

Target URL: <https://kb.builtwith.com/general-questions/expired-technology-website/>

## Executive Summary

Scan Results Summary:

- Medium: 4

# Detailed Findings

## 1. Outdated jQuery

**Severity:** Medium

**CVSS Score:** N/A

**Description:**

If a website is using an outdated jQuery version, it may be vulnerable to various security risks such as cross-site scripting (XSS) attacks, remote code execution (RCE) attacks, and denial-of-service (DoS) attacks. These vulnerabilities can be exploited by attackers to steal sensitive information, take control of the website, or disrupt its normal functioning. Moreover, outdated jQuery versions may not be compatible with the latest web standards and technologies, which can negatively impact the website's performance, accessibility, and user experience. Therefore, it is recommended to always use the latest version of jQuery to ensure the security and functionality of the website.

**Proof of Concept:**

Version 3.5.1 in use.

**Remediation Steps:**

- 1. Priority 1: Assess Compatibility - Before upgrading jQuery, it is essential to assess the compatibility of the existing application code with the new version of jQuery. This can be achieved by using jQuery Migrate plugin to detect any issues.
- 2. Priority 2: Testing - After ensuring compatibility, perform thorough testing of the application with the new version of jQuery. This should include unit testing, integration testing, and end-to-end testing.
- 3. Priority 3: Backup - Before making any changes to the production environment, take a backup of the existing code, database, and any other relevant data.
- 4. Priority 4: Deployment - Once testing is complete, deploy the updated application to the production environment.
- 5. Priority 5: Monitoring - After deployment, monitor the application for any issues related to

the jQuery upgrade.

### **Business Impact:**

**Financial Impact:** Outdated jQuery versions can lead to security breaches, allowing attackers to steal sensitive customer data and financial information, resulting in potential financial losses, legal liabilities, and regulatory fines.

**Reputation Risks:** Websites using outdated jQuery versions may be vulnerable to attacks that can harm their reputation, trustworthiness, and customer loyalty, potentially leading to a decline in website traffic, sales, and customer engagement.

**Compliance Implications:** Using outdated jQuery versions can violate various data protection and privacy regulations, such as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA), exposing the organization to legal and financial risks.

**Operational Disruption Risks:** Outdated jQuery versions can cause compatibility issues with the latest web standards and technologies, leading to website downtime, poor performance, and user experience issues, potentially impacting the organization's productivity, efficiency, and revenue.

## **2. Outdated Bootstrap**

**Severity:** Medium

**CVSS Score:** N/A

### **Description:**

Using an outdated version of Bootstrap can lead to various security vulnerabilities, as new versions often include patches for security bugs and other improvements. For example, an attacker could exploit a known vulnerability in an outdated Bootstrap version to execute arbitrary code, perform a cross-site scripting (XSS) attack, or gain unauthorized access to sensitive data. This could result in a significant impact on the confidentiality, integrity, and availability of the affected system. Therefore, it is essential to keep Bootstrap up-to-date to ensure the security of web applications that use it.

### **Proof of Concept:**

Version 4.3.1 in use.

### **Remediation Steps:**

- 1. Verify current Bootstrap version:
- 2. Review the project's dependencies:
- 3. Plan the upgrade process:
- 4. Upgrade Bootstrap:
- 5. Validate the upgrade:

### **Business Impact:**

Financial impact: The financial impact of this vulnerability could be significant if an attacker successfully exploits it. This could result in direct financial losses due to theft of sensitive data, unauthorized transactions, or system downtime. Additionally, the costs of remediation, legal fees, and potential regulatory fines could also contribute to the overall financial impact.

Reputation risks: A successful exploitation of this vulnerability could lead to a loss of trust and reputation damage for the organization. This could result in decreased customer confidence, negative publicity, and potential loss of business.

Compliance implications: Depending on the industry and regulatory requirements, this vulnerability could lead to compliance violations. For example, if the organization is subject to the Payment Card Industry Data Security Standard (PCI DSS), using an outdated version of Bootstrap could be a violation of requirement 6.2, which mandates the use of secure systems and software.

Operational disruption risks: An attacker could exploit this vulnerability to disrupt the organization's operations, leading to system downtime, data loss, or other operational issues. This could result in significant disruption to the organization's business operations and potentially impact its ability to deliver products or services to customers.

## **3. Outdated Microsoft ASP.NET**

**Severity:** Medium

**CVSS Score:** N/A

**Description:**

Identify two potential negative impacts in 1-2 sentences each.

Using an outdated Microsoft ASP.NET version can expose a system to various security risks. One major risk is the potential for exploitation of known vulnerabilities in the outdated version, which may allow unauthorized access, data breaches, or denial-of-service attacks. Additionally, outdated versions may not receive security patches or updates, leaving the system vulnerable to new threats as they emerge. Overall, using an outdated ASP.NET version can lead to significant technical impacts, including potential data loss, system downtime, and reputational damage.

**Proof of Concept:**

Version 4.0 in use.

**Remediation Steps:**

- 1. **\*\*Backup existing application and database\*\***: Before making any changes to the ASP.NET application, create a backup of the application and database. This ensures that you can easily restore the application to its previous state if any issues arise during the upgrade process.
- 2. **\*\*Install the latest version of .NET Framework\*\***: Install the latest version of .NET Framework that supports ASP.NET. This will ensure that the application has access to the latest features and security updates. Verify the installation by checking the version of .NET Framework installed on the server.
- 3. **\*\*Upgrade ASP.NET to the latest version\*\***: Upgrade the ASP.NET application to the latest version. This can be done by using the Visual Studio upgrade wizard or by manually updating the application's web.config file. Verify the upgrade by testing the application and ensuring that all features are working as expected.
- 4. **\*\*Test the application\*\***: Thoroughly test the application to ensure that it is functioning correctly. This includes testing all features, checking for any errors or issues, and validating

that the application is secure.

- 5. **\*\*Deploy the upgraded application\*\***: Once the application has been tested and verified, deploy the upgraded application to the production environment. Verify the deployment by accessing the application and ensuring that it is working as expected.

### **Business Impact:**

The financial impact of using an outdated ASP.NET version can be substantial. This can include direct costs such as remediation, fines, and legal fees, as well as indirect costs such as lost revenue due to downtime or damage to the company's reputation. Reputation risks are significant, as a data breach or other security incident can lead to loss of customer trust and damage to the company's brand. Compliance implications may include fines, sanctions, or legal action for failing to meet regulatory requirements around data security. Operational disruption risks are also high, as outdated software can cause system instability, slow performance, and compatibility issues with other software. Overall, using an outdated ASP.NET version can have serious consequences for a business, impacting its bottom line, reputation, compliance status, and operational efficiency.

## **4. CVE-2019-8331 - Bootstrap -**

**Severity:** Medium

**CVSS Score:** N/A

### **Description:**

Bootstrap is an open-source front-end development framework used to create responsive and mobile-first web projects. CVE-2019-8331 is a security vulnerability in Bootstrap that can lead to a cross-site scripting (XSS) attack. Specifically, the vulnerability is in the dropdown component of Bootstrap 4.3.1 and earlier versions. The attack vector involves an attacker crafting a malicious string that, when used as an option in a dropdown, will execute arbitrary JavaScript code in the user's browser. The technical impact of this vulnerability is that it can allow an attacker to steal user session cookies, inject malware, or perform other malicious actions on the user's behalf.

In professional terms, CVE-2019-8331 is a security flaw in the dropdown component of Bootstrap 4.3.1 and earlier versions that can be exploited through a cross-site scripting (XSS) attack. The vulnerability arises due to insufficient input validation, allowing an attacker to inject malicious JavaScript code into a dropdown option. This code can then be executed in the user's browser, enabling the attacker to carry out various malicious actions, such as stealing session cookies, injecting malware, or performing unauthorized actions on the user's behalf. The vulnerability highlights

### **Proof of Concept:**

Version: 4.3.1 is affected by CVE-2019-8331

### **Remediation Steps:**

- 1. Patch the vulnerability: The best way to mitigate this vulnerability is to apply the official patch released by the Bootstrap team. This patch upgrades the affected version of jQuery to a secure version that does not have the vulnerability. It is essential to apply this patch as soon as possible, as the vulnerability allows an attacker to execute arbitrary code on a victim's system.
- 2. Upgrade to a secure version of Bootstrap: If applying the patch is not an option, consider upgrading to a secure version of Bootstrap that does not contain the vulnerability. The vulnerability affects Bootstrap versions 3.4.0 to 3.4.1 and 4.3.1 to 4.3.2. Upgrading to Bootstrap version 3.4.2 or higher or version 4.3.3 or higher will resolve the vulnerability.
- 3. Implement Content Security Policy (CSP): Implementing a Content Security Policy can help prevent attacks that exploit the vulnerability. A CSP is a security feature that helps prevent cross-site scripting (XSS) attacks by specifying which sources of content are allowed to be executed on a website. By implementing a strict CSP, you can prevent the execution of malicious scripts that exploit the vulnerability.
- 4. Use a web application firewall (WAF): Implementing a web application firewall

### **Business Impact:**

The potential financial impact of CVE-2019-8331 includes the cost of remediation, such as patching or upgrading Bootstrap, as well as potential losses due to stolen session cookies or other malicious

actions performed by attackers. Reputation risks include damage to the organization's brand and loss of customer trust due to the vulnerability. Compliance implications may arise if the organization is subject to regulatory requirements related to data privacy or security, as failure to address the vulnerability could result in non-compliance penalties. Operational disruption risks include downtime or reduced productivity due to remediation efforts or potential attacks. Overall, the vulnerability highlights the importance of maintaining secure software development practices and staying up-to-date with security patches and updates.