Web Application Vulnerability Report

Scan Date: 2025-08-01 12:43:03 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 Outdated jQuery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:	CVSS: 5.0 (Medium)

Description:

Using an outdated jQuery version exposes your application to known security vulnerabilities, such as cross-site scripting (XSS) attacks, which can lead to unauthorized access and data breaches. Outdated jQuery versions may also lack important security patches, making them susceptible to exploits like DOM-based XSS and JSONP callbacks. This can result in malicious code execution, compromising the integrity and confidentiality of your application and user data.

Proof of Concept:

Version 3.5.1 in use.

Remediation Steps:

- Upgrade to 3.7.1.

Business Impact:

The outdated jQuery vulnerability poses a moderate financial impact, with potential costs arising from incident response, data breach notifications, and legal liabilities. The reputation of the business may also be at risk, as a successful exploit could lead to negative publicity and loss of customer trust. From a compliance perspective, the vulnerability may lead to non-compliance with regulatory requirements, such as GDPR and PCI-DSS, resulting in fines and penalties. Additionally, operational disruption risks are possible, as a malicious attack could compromise application availability and integrity, leading to downtime and lost productivity.

2. Outdated Bootstrap

Severity:	Medium
CVSS Score:	CVSS: 5.0 (Medium)

Description:

Using an outdated Bootstrap version poses significant security risks, as it may contain known vulnerabilities that can be exploited by attackers to inject malicious code or steal sensitive data. The technical impact includes exposure to cross-site scripting (XSS) attacks, cross-site request forgery (CSRF) attacks, and other client-side vulnerabilities that can compromise the security and integrity of web applications. Additionally, outdated Bootstrap versions may not receive security patches or updates, leaving them vulnerable to newly discovered exploits and increasing the risk of a successful attack.

Proof of Concept:

Version 4.3.1 in use.

Remediation Steps:

- 1. **Identify the current Bootstrap version**: Check the `bootstrap.min.css` or `bootstrap.min.js` file for the version number. You can also use the browser's developer tools to inspect the HTML elements and check the Bootstrap version.
- 2. **Check for compatibility issues**: Review the Bootstrap documentation and release notes to identify potential breaking changes and compatibility issues with your existing code.
- 3. **Plan the upgrade**: Create a plan to upgrade Bootstrap, including the new version, and identify the necessary changes to your code.
- 4. **Update Bootstrap CSS and JS files**: Replace the outdated `bootstrap.min.css` and `bootstrap.min.js` files with the latest versions. You can download the latest files from the official Bootstrap website or use a package manager like npm or yarn.
- 5. **Update other dependencies**: Check for other dependencies that may be affected by the Bootstrap upgrade, such as jQuery or Popper.js, and update them to compatible versions.

Business Impact:

The potential business impact of using an outdated Bootstrap version is significant, with potential financial losses due to compromised sensitive data or successful XSS and CSRF attacks. The reputation of the organization may also be severely damaged if a security breach occurs, leading to loss of customer trust and potential legal liabilities. Non-compliance with security regulations and standards may result in additional fines and penalties, further exacerbating the financial burden. Furthermore, a successful attack could lead to operational disruptions, causing downtime and impacting business continuity, which could have long-term consequences for the organization's overall viability.

3. Outdated Microsoft ASP.NET

Severity: Medium

CVSS Score: CVSS: 5.0 (Medium)

Description:

Using an outdated Microsoft ASP.NET version can expose applications to known security vulnerabilities, allowing attackers to exploit unpatched bugs and potentially gain unauthorized access to sensitive data. Outdated ASP.NET versions may lack essential security patches and updates, making them susceptible to common web attacks such as SQL injection and cross-site scripting (XSS). This can lead to technical impacts including data breaches, compromised user accounts, and malicious code execution, ultimately compromising the integrity and confidentiality of the application and its data.

Proof of Concept:

Version 4.0 in use.

Remediation Steps:

- Upgrade to 4.8.

Business Impact:

The potential business impact of using an outdated Microsoft ASP.NET version is significant, with potential financial losses due to data breaches and compromised user accounts estimated to be substantial. The reputation of the organization may also be severely damaged, leading to loss of customer trust and potential legal action, with long-term consequences for the business. Non-compliance with security regulations and standards, such as GDPR and PCI-DSS, may result in hefty fines and penalties, further exacerbating the financial burden. Additionally, the vulnerability may cause operational disruptions, including downtime and resource-intensive remediation efforts, which can hinder business operations and impact revenue.

4. CVE-2019-8331 - Bootstrap -

Severity: Medium

CVSS Score: CVSS: 5.0 (Medium)

Description:

CVE-2019-8331 is a vulnerability in Bootstrap's tooltip and popover components, which allows for cross-site scripting (XSS) attacks due to insufficient sanitation of user-input data. An attacker can exploit this vulnerability by injecting malicious JavaScript code through crafted tooltip or popover content, potentially leading to unauthorized access, data theft, or other malicious activities. The technical impact of this vulnerability is moderate to high, as it can be triggered through various attack vectors, including phishing emails, infected websites, or compromised user input, ultimately compromising the security and integrity of the affected web application.

Proof of Concept:

Version: 4.3.1 is affected by CVE-2019-8331

Remediation Steps:

- 1. **Update to Bootstrap version 4.3.1 or later**: This is the most effective solution, as it directly addresses the vulnerability. According to the CVE details, "Bootstrap before 4.3.1 has a Cross-Site Scripting (XSS) vulnerability via the tooltip or popover data-template attribute". Updating to version 4.3.1 or later will fix this issue. [1]

- 2. **Validate and sanitize user input**: To prevent XSS attacks, it's essential to validate and sanitize all user input. This can be achieved by using libraries like DOMPurify or by implementing custom validation and sanitization logic. For example, you can use a whitelist approach to only allow specific HTML tags and attributes. [2]

3. **Use a Content Security Policy (CSP)**: Implementing a CSP can help mitigate XSS

attacks by defining which sources of content are allowed to be executed within a web page. This can be done by setting the `Content-Security-Policy` header in your web server or application. For example: `Content-Security-Policy: default-src 'self'; script-src 'self'

https://cdn.jsdelivr.net;`. [3]

- 4. **Disable tooltips and popovers if not necessary**: If tooltips and popovers are not essential to your application, consider disabling them to prevent potential XSS attacks. This can be done by removing the `tooltip` and `popover` JavaScript plugins from your Bootstrap configuration. [4]

Business Impact:

The CVE-2019-8331 vulnerability in Bootstrap poses a moderate to high business impact, with potential financial losses due to unauthorized access, data theft, or other malicious activities, which could result in significant costs for incident response, remediation, and potential legal liabilities. The reputation of an organization could also be severely damaged if an XSS attack is successfully executed, leading to a loss of customer trust and potential revenue decline. From a compliance perspective, failure to address this vulnerability could lead to non-compliance with regulatory requirements, such as GDPR or PCI-DSS, resulting in fines and penalties. Additionally, the operational disruption risks are substantial, as a successful attack could compromise the availability and integrity of critical web applications, leading to downtime, data loss, and disruption of business operations.