Web Application Vulnerability Report

**Scan Date:** 2025-08-01 08:14:11 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 Outdated jQuery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:

Using an outdated version of jQuery can expose a website to various security risks, even if no known vulnerabilities exist in the specific version being used. This is because outdated versions may not include the latest security patches and improvements, making them more susceptible to known and unknown threats. For example, an attacker could exploit a vulnerability in an older version of jQuery to inject malicious code, steal sensitive data, or perform other malicious actions. This could result in unauthorized access, data breaches, and other technical impacts that can compromise the confidentiality, integrity, and availability of the website and its data. Therefore, it is essential to keep jQuery and all other third-party libraries up-to-date to minimize security risks and ensure the ongoing stability and security of the website.

### Proof of Concept:

Version 3.5.1 in use.

### Remediation:

* 1. First, you will need to determine which version of jQuery you are currently using. This can be done by checking the version number in the jQuery file or by using a tool like the jQuery Migrate plugin to detect the version. Verification step: Check the jQuery file or use the jQuery Migrate plugin to determine the current version.
* 2. Next, you should check the documentation for the version of jQuery you are currently using to determine if there are any breaking changes or deprecated features that you need to be aware of. This will help you to plan for any necessary code changes or updates. Verification step: Review the documentation for the current version of jQuery to identify any breaking changes or deprecated features.
* 3. Once you have determined the current version of jQuery and reviewed the documentation, you can begin the process of upgrading to the desired version. This will typically involve downloading the new version of jQuery and updating the script tag in your HTML file to point to the new version. Verification step: Check the HTML file to ensure that the script tag points to the new version of jQuery.
* 4. After upgrading to the new version of jQuery, you should thoroughly test your application to ensure that all features and functionality are working as expected. This may involve manually testing individual features or using automated testing tools to check for regressions. Verification step: Test the application to ensure that all features and functionality are working as expected.
* 5. Finally, you should consider implementing a process for regularly updating jQuery to ensure that you are using the most current version and taking advantage of the latest features and security fixes. This may involve setting up a schedule for checking for new versions and testing the application after each update. Verification step: Implement a process for regularly updating jQuery and testing the application after each update.

### Business Impact:

Financial impact: The financial impact of using an outdated version of jQuery could be significant, as it could lead to unauthorized access, data breaches, and other security incidents that result in financial losses. This could include costs associated with data recovery, legal fees, regulatory fines, and reputational damage. Additionally, if sensitive customer or financial data is compromised, the organization could face lawsuits, regulatory sanctions, and loss of business.  
  
Reputation risks: The use of an outdated version of jQuery could also damage the organization's reputation, as it may be perceived as negligent or careless in its approach to security. This could result in loss of trust and credibility among customers, partners, and other stakeholders, and could impact the organization's ability to attract new business or retain existing customers.  
  
Compliance implications: The use of an outdated version of jQuery could also have compliance implications, as it may violate regulatory requirements related to data privacy and security. For example, if the organization is subject to the General Data Protection Regulation (GDPR), it may be required to implement appropriate technical and organizational measures to ensure the security of personal data. Failure to do so could result in fines of up to 4% of global annual revenue or €20 million (whichever is greater).  
  
Operational disruption risks: Finally, the use of an outdated version of jQuery could also disrupt the organization's operations, as it may result in website downtime, performance issues, or other technical problems. This could impact the organization's ability to deliver products or services to customers, and could result in lost revenue, productivity, and customer satisfaction.

## 2. Outdated Bootstrap

**Severity:** Medium **CVSS Score:** N/A

### Description:

Using an outdated Bootstrap version can expose a web application to several security risks. Firstly, outdated versions may contain known vulnerabilities that can be exploited by attackers to gain unauthorized access, steal sensitive data, or disrupt the availability of the application. Secondly, outdated versions may not support the latest security features and protocols, making the application more vulnerable to attacks. Thirdly, using an outdated version can also affect the application's performance and compatibility with other technologies, leading to a poor user experience.  
  
Therefore, it is crucial to keep the Bootstrap version up-to-date to ensure the security and efficiency of the web application. However, if none is available, it is essential to conduct a thorough security assessment of the application to identify and mitigate any potential vulnerabilities. This assessment should include regular penetration testing, code review, and vulnerability scanning to ensure that the application is secure and meets the required security standards.

### Proof of Concept:

Version 4.3.1 in use.

### Remediation:

* 1. First, review the official documentation for upgrading Bootstrap. This will provide a comprehensive understanding of the process and any potential challenges.
* 2. Backup all custom CSS and JavaScript code, as well as any modifications made to the HTML templates. This will ensure that any changes made during the upgrade can be easily reverted if necessary.
* 3. Install the latest version of Bootstrap using a package manager such as npm or Yarn. This can be done by running the command `npm install bootstrap` or `yarn add bootstrap` in the project directory.
* 4. Update the HTML templates to use the new Bootstrap classes and components. This may involve making changes to the layout, typography, forms, and other elements.
* 5. Verify that the upgrade was successful by testing the website on various devices and browsers. Ensure that all functionality is working as expected and that there are no visual issues.

### Business Impact:

The potential financial impact of using an outdated Bootstrap version can include the cost of remediation, legal fees, and potential fines for non-compliance with data protection regulations. Reputation risks include loss of customer trust and damage to brand reputation, which can result in a decrease in sales and revenue. Compliance implications may arise if the outdated version does not meet the required security standards, leading to non-compliance with regulations such as GDPR, HIPAA, or PCI-DSS. Operational disruption risks include downtime, reduced productivity, and potential loss of data due to security breaches. Therefore, it is essential to keep the Bootstrap version up-to-date and conduct regular security assessments to mitigate these risks.

## 3. Outdated Microsoft ASP.NET

**Severity:** Medium **CVSS Score:** N/A

### Description:

Outdated versions of Microsoft ASP.NET may expose web applications to various security vulnerabilities, such as injection attacks, cross-site scripting, and denial of service. These risks can lead to unauthorized access, data theft, and disruption of service. It is crucial to keep ASP.NET up-to-date to ensure the protection of web applications and their sensitive data.

### Proof of Concept:

Version 4.0 in use.

### Remediation:

* 1. Backup current environment: Before upgrading ASP.NET, it's crucial to take a complete backup of the current environment. This includes the web application code, database, and any other related components. Verification: Check the backup logs to ensure that the backup process was successful.
* 2. Install the latest version of the .NET Framework: To upgrade ASP.NET, you need to install the latest version of the .NET Framework. Verify that the installation was successful by checking the version number in the Add or Remove Programs list.
* 3. Update the web application code: After installing the latest version of the .NET Framework, you need to update the web application code to use the new features and APIs. Verification: Test the updated application in a development or staging environment to ensure that it works as expected.
* 4. Update the database schema: If the web application uses a database, you may need to update the database schema to accommodate the new features and APIs. Verification: Test the updated application with the updated database schema in a development or staging environment to ensure that it works as expected.
* 5. Test the upgraded application: Before deploying the upgraded application to a production environment, it's essential to test it thoroughly in a development or staging environment. Verification: Check the test results to ensure that the upgraded application meets the required performance, security, and functional standards.

### Business Impact:

Financial Impact:  
Outdated versions of Microsoft ASP.NET can lead to significant financial losses due to data breaches, which can result in hefty fines from regulatory bodies and lawsuits from affected customers. Additionally, unplanned downtime due to service disruptions can negatively impact revenue and customer trust.  
  
Reputation Risks:  
Data breaches and service disruptions caused by outdated ASP.NET versions can tarnish a company's reputation, leading to a loss of customer trust and potentially damaging business relationships. The negative publicity surrounding such events can deter potential customers and hinder growth.  
  
Compliance Implications:  
Organizations that fail to maintain up-to-date versions of ASP.NET risk non-compliance with various industry regulations, such as the General Data Protection Regulation (GDPR), the Health Insurance Portability and Accountability Act (HIPAA), and the Payment Card Industry Data Security Standard (PCI DSS). Non-compliance can result in hefty fines and legal consequences.  
  
Operational Disruption Risks:  
Outdated ASP.NET versions can lead to unplanned downtime due to service disruptions caused by security vulnerabilities. This can negatively impact operational efficiency, productivity, and customer experience. The time and resources required to resolve these issues can also strain IT departments and divert attention from strategic initiatives.

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

**Severity:** Medium **CVSS Score:** N/A

### Description:

Bootstrap is an open-source toolkit for developing responsive and mobile-first web projects. CVE-2019-8331 is a vulnerability related to Bootstrap's carousel component, which allows for a cross-site scripting (XSS) attack. The attack vector involves an attacker injecting malicious scripts into the carousel component through user input, which is then executed in the victim's browser. This can lead to unauthorized access, data theft, and other malicious activities. The technical impact of this vulnerability includes the potential for information disclosure, data modification, and service disruption. It is important to note that the CVSS score for this vulnerability is not available, as it depends on the specific implementation and configuration of the affected system.

### Proof of Concept:

Version: 4.3.1 is affected by CVE-2019-8331

### Remediation:

* 1. Patching - The most effective way to mitigate this vulnerability is by upgrading to the latest version of Bootstrap (4.3.1) that addresses this issue. This will ensure that the vulnerability is completely removed and the application is secure.
* 2. Virtual Patching - If patching is not an option, consider implementing a virtual patch that blocks the exploitation of this vulnerability. This can be done by configuring a web application firewall (WAF) to block requests with specific patterns that match the exploit.
* 3. Input Validation - Implement strict input validation on user-supplied data to prevent the exploitation of this vulnerability. This can be done by validating all user-supplied data against a whitelist of allowed characters and formats.
* 4. Security Testing - Regularly perform security testing, including penetration testing and vulnerability assessments, to ensure that the application is free from known vulnerabilities. This will help identify and remediate any potential security issues before they can be exploited.
* 5. Security Training - Provide regular security training to developers and administrators to ensure that they are aware of the latest security threats and how to prevent them. This will help ensure that the application is designed and maintained with security in mind.

### Business Impact:

Financial impact: The financial impact of this vulnerability could be significant, as it may lead to unauthorized access to sensitive data and systems, resulting in financial loss due to data theft, fraud, or extortion. Additionally, organizations may face potential regulatory fines and legal actions if they fail to protect customer data, leading to additional financial burdens.  
  
Reputation risks: This vulnerability poses a significant reputation risk for organizations, as it can result in loss of customer trust and confidence, negative publicity, and potential legal actions. A successful XSS attack can damage an organization's reputation, leading to long-term consequences such as decreased market share and revenue.  
  
Compliance implications: Organizations may face compliance implications if they fail to address this vulnerability promptly. Depending on the industry and jurisdiction, organizations may be required to comply with data protection regulations such as the General Data Protection Regulation (GDPR) or the California Consumer Privacy Act (CCPA). Failure to comply with these regulations can result in regulatory fines and legal actions.  
  
Operational disruption risks: This vulnerability can also lead to operational disruption, as it may allow attackers to modify or delete critical data and systems, leading to service disruptions and downtime. This can result in lost productivity, decreased customer satisfaction, and potential revenue loss.  
  
Recommendations:  
  
1. Patch the vulnerability by upgrading to the latest version of Bootstrap, which addresses CVE-2019-8331.  
2. Implement input validation and sanitization to prevent attackers from injecting malicious scripts into user input fields.  
3. Implement a web application firewall (WAF) to detect and prevent XSS attacks.  
4. Conduct regular security audits and penetration testing to identify and address vulnerabilities in web applications.  
5