Sri Lanka Institute of Information Technology

Bug Bounty Report 07

win.aswatson.com

IE2062 – Web Security

Submitted by:

IT22227836 – ADHIKARI A.M.V.B.M

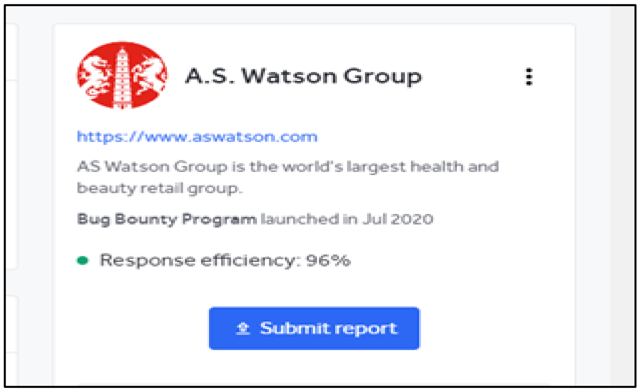
Date of submission

2024.10.31

Table of Contents

Report 07 – win.aswatson.com

Main domain	https://www.aswatson.com/
Sub domain	win.aswatson.com
IP address	210.0.245.141
platform	HackerOne



A.S. Watson Group

https://www.aswatson.com

AS Watson Group is the world's largest health and beauty retail group.

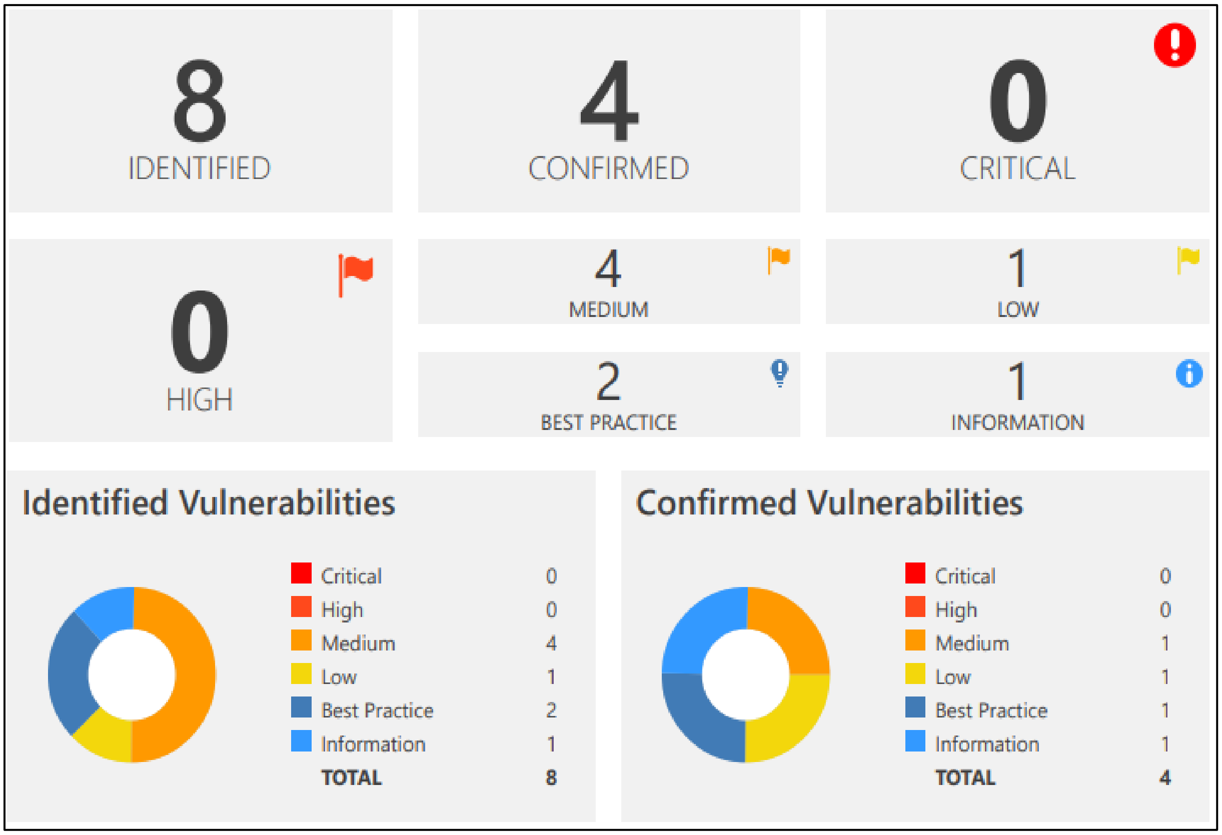
Bug Bounty Program launched in Jul 2020

● Response efficiency: 96%

The A.S. Watson Group is the world's largest health and beauty retail group, with over 15,700 stores in 25 markets worldwide serving over 28 million customers per week, and over 3 billion customers and members.

A.S. Watson Group looks forward to working with the security community to discover vulnerabilities to keep our businesses and customers safe. As we operate in many different countries, we will be rolling out our bug bounty program in phases. Our focus within this rollout, is our retail websites (and soon, mobile apps on both Android and IOS)

Vulnerability detected



8
IDENTIFIED

4
CONFIRMED

0
CRITICAL

0
HIGH

4
MEDIUM

1
LOW

2
BEST PRACTICE

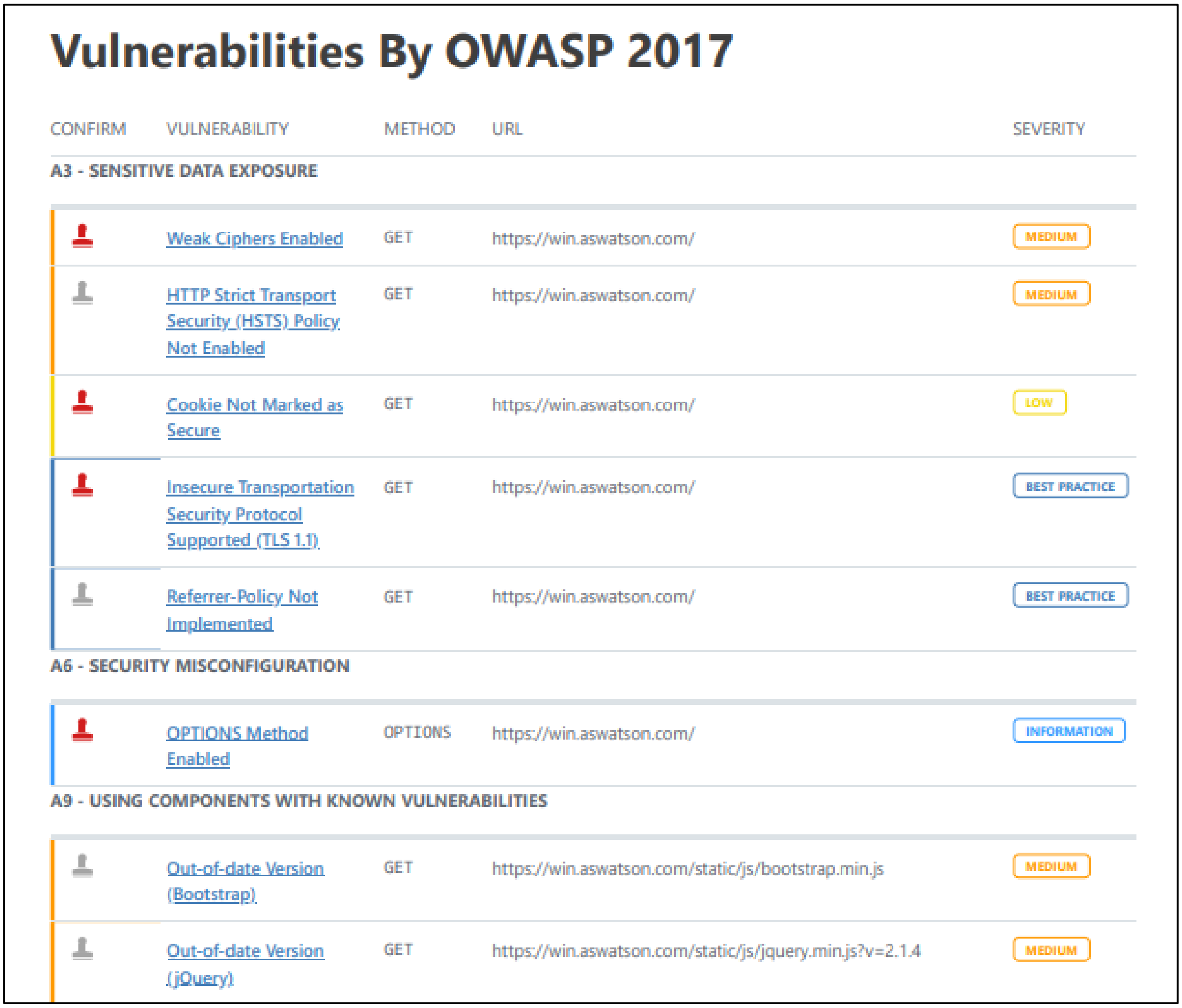
1
INFORMATION

Identified Vulnerabilities

🟥	Critical	0
🟧	High	0
🟧	Medium	4
🟨	Low	1
🟦	Best Practice	2
🟦	Information	1
	TOTAL	**8**

Confirmed Vulnerabilities

🟥	Critical	0
🟧	High	0
🟧	Medium	1
🟨	Low	1
🟦	Best Practice	1
🟦	Information	1
	TOTAL	**4**

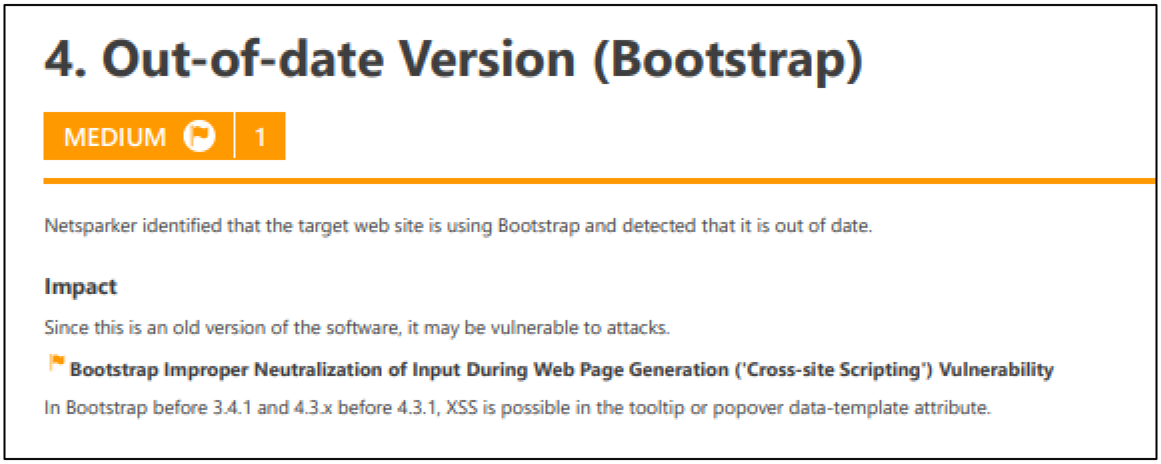


Vulnerabilities By OWASP 2017

CONFIRM	VULNERABILITY	METHOD	URL	SEVERITY
A3 - SENSITIVE DATA EXPOSURE				
✓	Weak Ciphers Enabled	GET	https://win.aswatson.com/	MEDIUM
	HTTP Strict Transport Security (HSTS) Policy Not Enabled	GET	https://win.aswatson.com/	MEDIUM
✓	Cookie Not Marked as Secure	GET	https://win.aswatson.com/	LOW
✓	Insecure Transportation Security Protocol Supported (TLS 1.1)	GET	https://win.aswatson.com/	BEST PRACTICE
	Referrer-Policy Not Implemented	GET	https://win.aswatson.com/	BEST PRACTICE
A6 - SECURITY MISCONFIGURATION				
✓	OPTIONS Method Enabled	OPTIONS	https://win.aswatson.com/	INFORMATION
A9 - USING COMPONENTS WITH KNOWN VULNERABILITIES				
	Out-of-date Version (Bootstrap)	GET	https://win.aswatson.com/static/js/bootstrap.min.js	MEDIUM
	Out-of-date Version (jQuery)	GET	https://win.aswatson.com/static/js/jquery.min.js?v=2.1.4	MEDIUM

Vulnerability

1. Title - Out-of-date Version (Bootstrap)



4. Out-of-date Version (Bootstrap)

MEDIUM | 1

Netsparker identified that the target web site is using Bootstrap and detected that it is out of date.

Impact

Since this is an old version of the software, it may be vulnerable to attacks.

Bootstrap Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

In Bootstrap before 3.4.1 and 4.3.x before 4.3.1, XSS is possible in the tooltip or popover data-template attribute.

- Risk – Medium

Description

The website used an older version of Bootstrap v3.3.6, which is highly vulnerable to different security issues, specifically the cross-site scripting XSS attacks. Bugs in XSS allow an attacker to inject malicious scripts into other clients' web pages viewed in a user's browser. This can lead to stolen cookies, session hijacking, and other malicious activities, including user redirects to malicious sites. Bootstrap has addressed these exposures in their newer versions. Its most current stable branch release is 3.4.1, which contains patches for known XSS weaknesses.

Make sure to upgrade Bootstrap to v3.4.1 or later. Upgrading will safeguard the website from known vulnerabilities in Bootstrap, such as XSS, and security posture in general.

Affected components

Affected Components due to the very outdated Bootstrap Version - 3.3.6:

- **Bootstrap JavaScript Library:**

  This is the main library that involves all the components required to be utilized in creating responsive web applications. Among them, one file identified is bootstrap.min.js, which plays a critical role in the development of interactive elements on the website.

- **Web Application Frontend:**

  These are for UI elements or components using Bootstrap that depend on it, like modals, tool tips, and drop-downs. These now carry with them the security vulnerabilities of the very type mentioned in the outdated version and thus are exposed to XSS attacks.

- **Dependencies:**

  Other libraries or frameworks that are dependent upon Bootstrap for either styling or functionality could also be indirectly vulnerable to these issues.

Impact Assessment

- **Security Vulnerabilities:**
  Bootstrap version 3.3.6 is vulnerable to XSS attacks; malicious scripts can inject and may result in sensitive data theft or session hijacking.

- **Compliance Risks:**
  Non-compliance with standards like PCI DSS and HIPAA, which regulate periodic updating of the software for sensitive data protection.

- **User Trust:**
  Security breaches by using libraries that are outdated will amount to loss of customer trust, hence loss to reputation.

- **Limited Support:**
  The older versions might not have official support, which complicates issues and resolution of vulnerabilities.

- **Compatibility:**
  Older libraries may not seamlessly be compatible with new modern technologies, hence could raise development and maintenance problems.

Steps to reproduce

- **Identify Bootstrap Version:**
  Review the web application source code, in either HTML or JavaScript files, to find out what version of Bootstrap is being used. Note: It will show something like bootstrap.min.js.

- **Verify Version Information:**
  Check either the Bootstrap documentation or their GitHub repository to confirm that 3.3.6 is indeed lower than the latest stable release, which is 3.4.1.

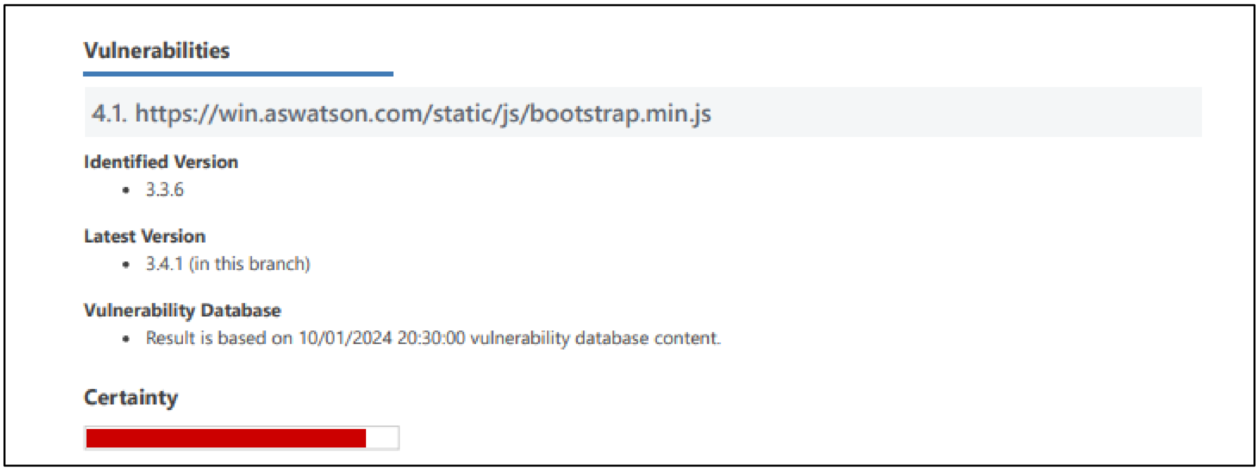
- **Security Scans:**
  Employ a vulnerability scanning tool-such as Netsparker or Burp Suite-to identify known vulnerabilities related to the identified version of Bootstrap. Pay particular attention to XSS vulnerabilities.

- **XSS Vulnerabilities Test:**
  Create a web page utilizing some Bootstrap functionality-popover or tooltip. Payloads should leverage the XSS vulnerability in the older version.
  Example payloads are JavaScript code fragments, which may be injected into the tooltip or popover attributes.

- **Anarchy - Application Behavior Analysis:**
  Trigger the Bootstrap components with injected payloads to see whether the application executes the scripts, confirming XSS vulnerabilities. Documentation of findings: Document the proof of the actual vulnerabilities that are present and ways they can be exploited to provide the required proof of an issue to stakeholders

Proof of concept (if applicable)

Vulnerability scanning using Netsparker



Vulnerabilities

4.1. https://win.aswatson.com/static/js/bootstrap.min.js

Identified Version

- 3.3.6

Latest Version

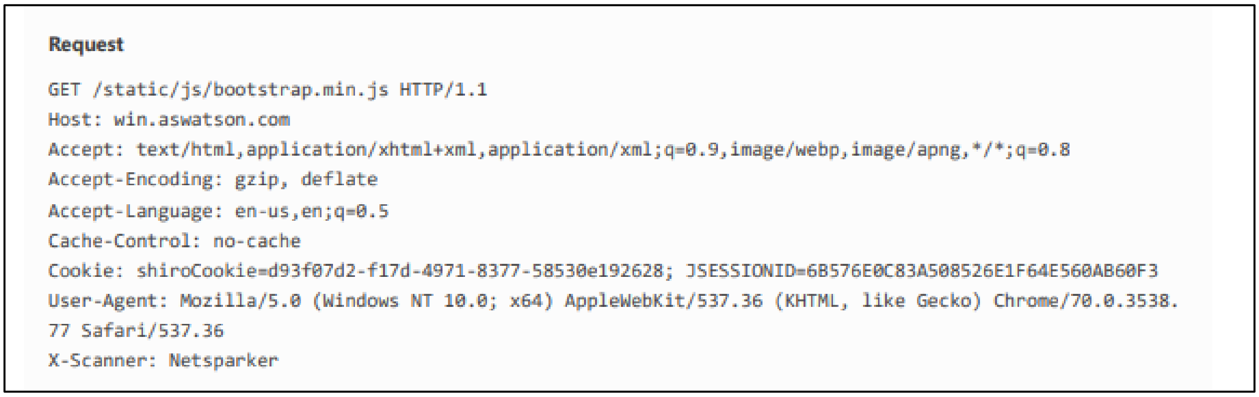
- 3.4.1 (in this branch)

Vulnerability Database

- Result is based on 10/01/2024 20:30:00 vulnerability database content.

Certainty

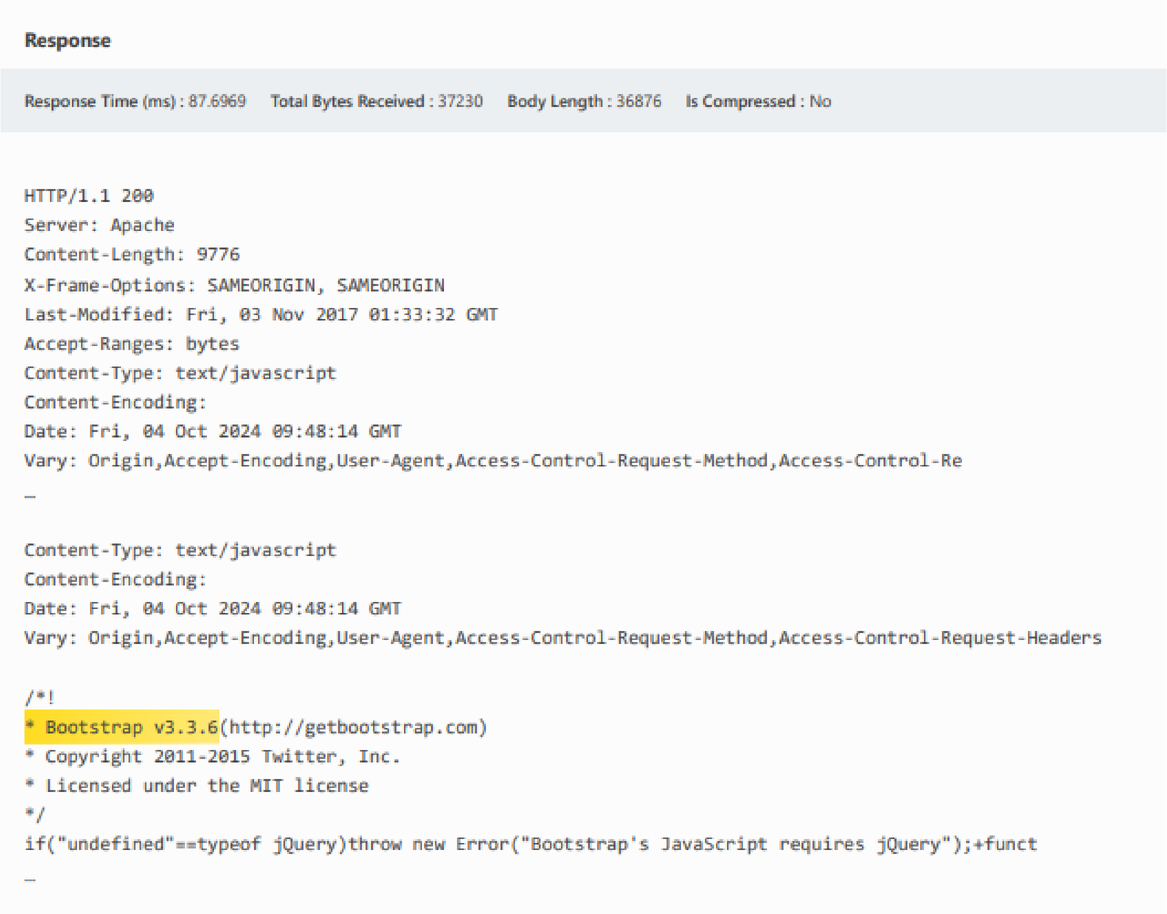
- **Request**



Request

```
GET /static/js/bootstrap.min.js HTTP/1.1
Host: win.aswatson.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: shiroCookie=d93f07d2-f17d-4971-8377-58530e192628; JSESSIONID=6B576E0C83A508526E1F64E560AB60F3
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

- ## Response



Response

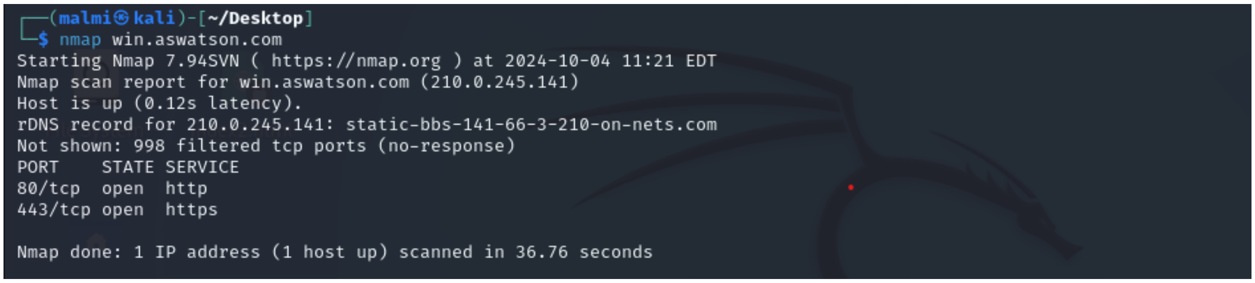
Response Time (ms) : 87.6969 Total Bytes Received : 37230 Body Length : 36876 Is Compressed : No

```
HTTP/1.1 200
Server: Apache
Content-Length: 9776
X-Frame-Options: SAMEORIGIN, SAMEORIGIN
Last-Modified: Fri, 03 Nov 2017 01:33:32 GMT
Accept-Ranges: bytes
Content-Type: text/javascript
Content-Encoding:
Date: Fri, 04 Oct 2024 09:48:14 GMT
Vary: Origin,Accept-Encoding,User-Agent,Access-Control-Request-Method,Access-Control-Re
…

Content-Type: text/javascript
Content-Encoding:
Date: Fri, 04 Oct 2024 09:48:14 GMT
Vary: Origin,Accept-Encoding,User-Agent,Access-Control-Request-Method,Access-Control-Request-Headers

/*!
* Bootstrap v3.3.6(http://getbootstrap.com)
* Copyright 2011-2015 Twitter, Inc.
* Licensed under the MIT license
*/
if("undefined"==typeof jQuery)throw new Error("Bootstrap's JavaScript requires jQuery");+funct
…
```

Vulnerability finding using namp

Used command

- nmap win.aswatson.com

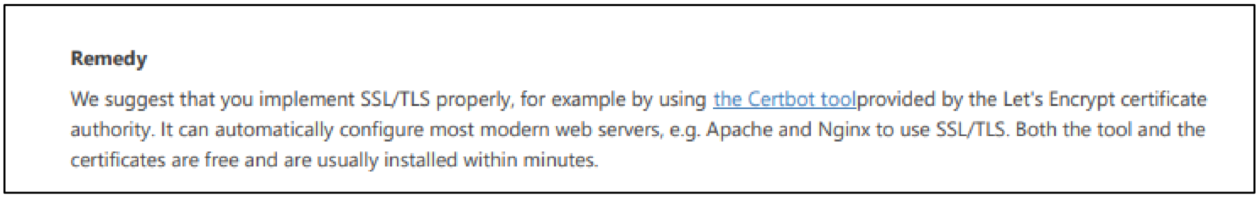


```
┌──(malmi㉿kali)-[~/Desktop]
└─$ nmap win.aswatson.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-04 11:21 EDT
Nmap scan report for win.aswatson.com (210.0.245.141)
Host is up (0.12s latency).
rDNS record for 210.0.245.141: static-bbs-141-66-3-210-on-nets.com
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 36.76 seconds
```

Founded open port

- 80/tcp open http
- 443/tcp open https

Proposed mitigation or fix



Remedy

We suggest that you implement SSL/TLS properly, for example by using the Certbot toolprovided by the Let's Encrypt certificate authority. It can automatically configure most modern web servers, e.g. Apache and Nginx to use SSL/TLS. Both the tool and the certificates are free and are usually installed within minutes.

To mitigate or fix the Out-of-date Version (Bootstrap) vulnerability:

- **Upgrade Bootstrap:**

  Update Bootstrap to the current stable version. At the time of writing currently 3.4.1 or bump directly to the most recent Bootstrap 4 or 5, which is already patched against known security vulnerabilities.

- **Secure Client-Side Code:**

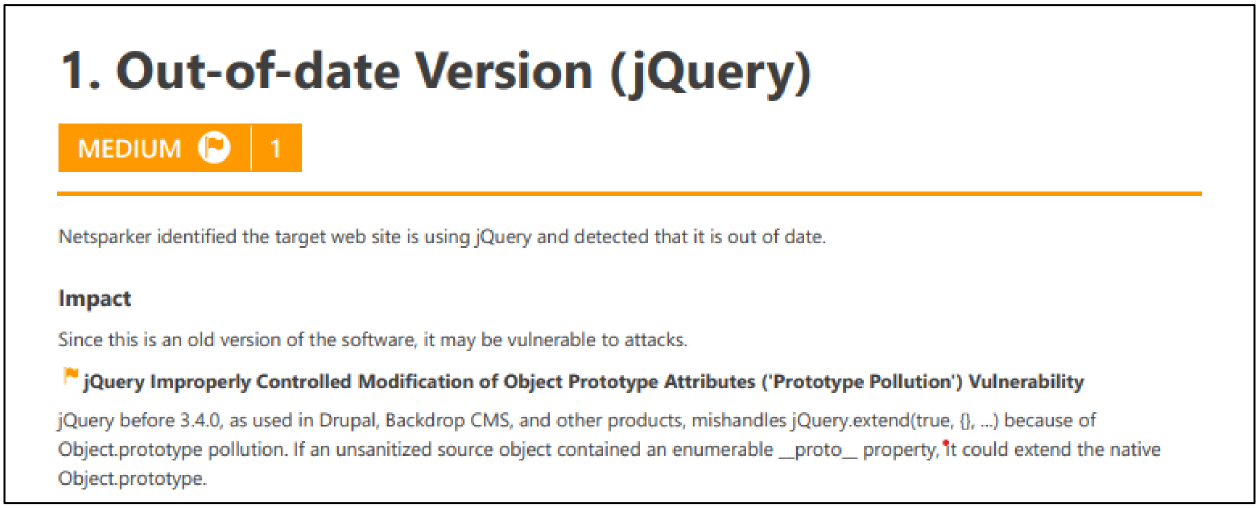
  Review your client-side code, i.e., JavaScript and HTML using Bootstrap components, especially tooltips and popovers for good sanitization, without allowing malicious input.

- **Input Validation:**

  Do abundant input validation and sanitization to avoid XSS attacks anywhere in your application, including where Bootstrap components may reside.

- **Regular Security Audits:**
  Also, regularly audit your web assets and libraries. Update all third-party components to their recent versions. This is minimized in risk by upgrading and making sure proper sanitization is applied, due to security vulnerabilities present within older versions of Bootstrap.

2. Tittle - Out-of-date Version (jQuery)



1. Out-of-date Version (jQuery)

MEDIUM | 1

Netsparker identified the target web site is using jQuery and detected that it is out of date.

Impact

Since this is an old version of the software, it may be vulnerable to attacks.

jQuery Improperly Controlled Modification of Object Prototype Attributes ('Prototype Pollution') Vulnerability

jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution. If an unsanitized source object contained an enumerable __proto__ property, it could extend the native Object.prototype.

❖ Risk = Medium

Description

Out-of-date Version/jQuery: A vulnerability where an application is using an outdated version of the jQuery library, such as version 2.1.4. Older versions of jQuery may be subject to known vulnerabilities, such as Prototype Pollution and Cross-Site Scripting, which attackers might leverage to inject malicious code or execute it.

This vulnerability can lead to integrity compromise of the website through DOM manipulation and malicious script injections. Such a vulnerability can be patched by upgrading jQuery to its latest stable version-that is, at least version 3.5.0-so that it contains protection against such security threats.

Affected components

The stack with Out-of-date Version (jQuery) vulnerability includes:

- **jQuery Library:**
  The exact version used is 2.1.4. This version has known attacks such as Cross-Site Scripting and Prototype Pollution.

- **Web Applications:**
  Any of the web applications which rely on this version for DOM manipulation, event handling, etc, are vulnerable to those.

- **Similar Vulnerabilities:**
  Most popular frameworks or content management systems that rely on jQuery can also be vulnerable to these, especially those depending on an older version of jQuery for functionality.

- **User Input Handling:**
  Templates, which at any given time handle/ process user-generated content using jQuery methods-like .html(), .append(), etc.-are vulnerable to execute untrusted code if their jQuery version is vulnerable.

Impact Assessment

The Out-of-date Version vulnerability in jQuery will have several serious impacts on the subjected system

- **Security Risks:**
  Old versions of jQuery, such as 2.1.4, have been known to be susceptible to several security vulnerabilities, which include XSS and Prototype Pollution. Successful exploitation could allow an attacker to inject malicious scripts or achieve object manipulation in the web page, thus breaking the integrity and security of the web application.

- **Data Exposure:**

  Older versions of jQuery carry vulnerabilities that show the way to unauthorized parties for sensitive information disclosure or allow executing arbitrary code to leak/manipulate data.

- **Reputation Damage:**
  The exploitation of the same could lead to security breaches or even defacements, which will bring reputational damage to the organization concerned.

- **Non-compliance:**
  Organizations operating on software versions older than the latest patch may be deemed non-compliant with regulations like PCI DSS, HIPAA, or ISO27001 and may receive serious legal and financial consequences for doing so.

Steps to reproduce

- **jQuery Version Check:**
  Check the web application to confirm that jQuery 2.1.4 is in use, either by viewing the page source or using developer tools.

- **Identify known vulnerabilities:**
  Check for known vulnerabilities related to jQuery version 2.1.4 using databases like CVE or OWASP.

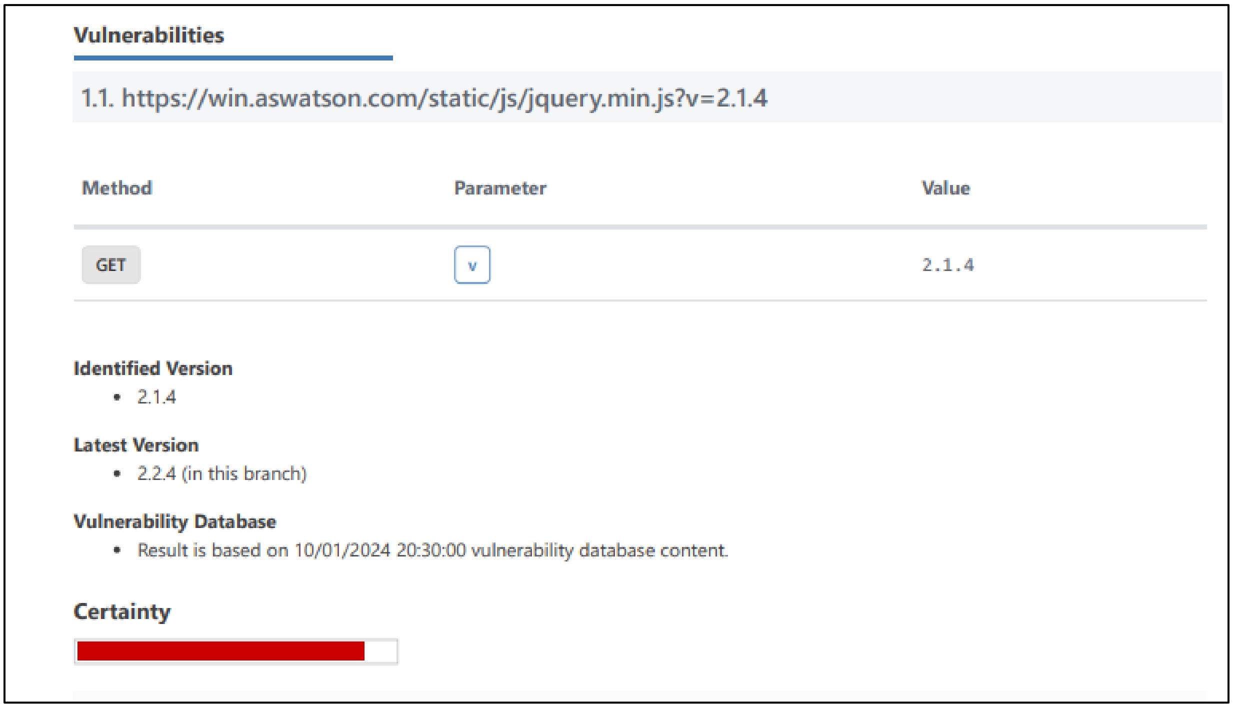
- **Perform a Test:**
  Use jQuery methods known to be vulnerable, such as.html () or. Append (), in a web application to inject untrusted content. If the application executes the injected script, it is considered an XSS vulnerability.

- **Vulnerability to Prototype Pollution:**
  Verify whether the modification of object properties leads to unexpected behavioral changes, which would give a hint about the prototype pollution problem.

- **Document Outcomes:**
  Perform thorough tests of the identified vulnerabilities, including observations and impact. Record detailed accounts of this. The following are the steps one should undertake in effectively demonstrating the vulnerabilities concerned with the obsolete version of jQuery.

Proof of concept (if applicable)

Vulnerability scanning using Netsparker



Vulnerabilities

1.1. https://win.aswatson.com/static/js/jquery.min.js?v=2.1.4

Method	Parameter	Value
GET	v	2.1.4

Identified Version

- 2.1.4

Latest Version

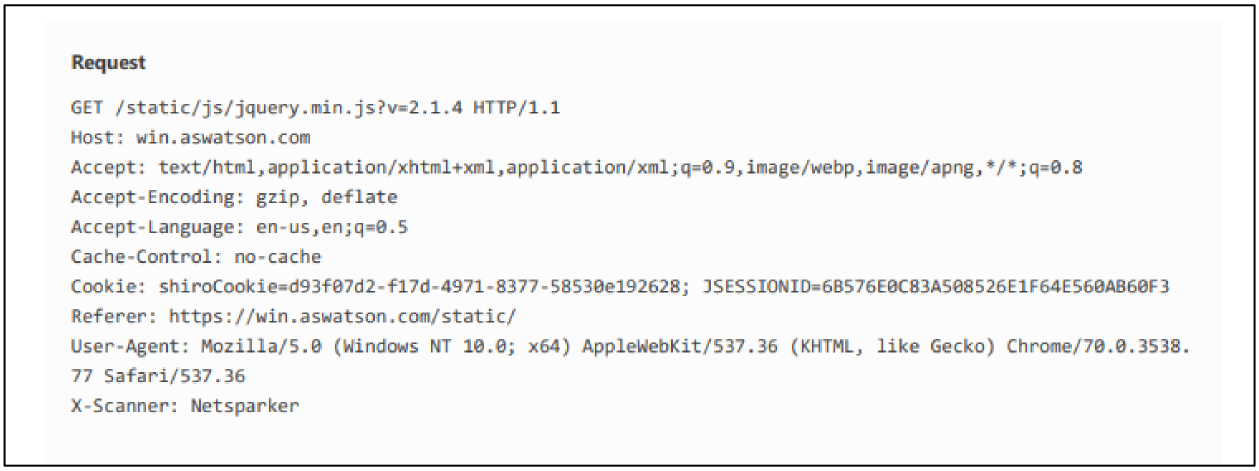
- 2.2.4 (in this branch)

Vulnerability Database

- Result is based on 10/01/2024 20:30:00 vulnerability database content.

Certainty

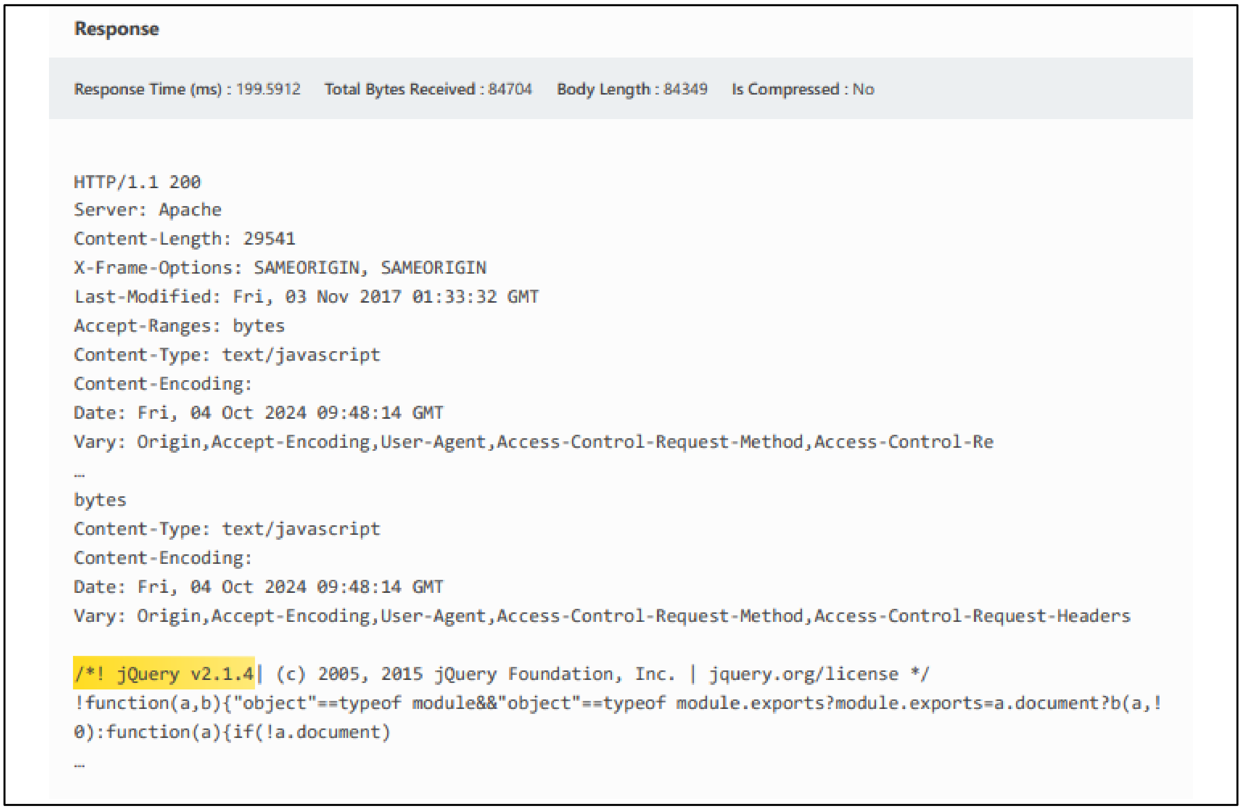
- **Request**



Request

```
GET /static/js/jquery.min.js?v=2.1.4 HTTP/1.1
Host: win.aswatson.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: shiroCookie=d93f07d2-f17d-4971-8377-58530e192628; JSESSIONID=6B576E0C83A508526E1F64E560AB60F3
Referer: https://win.aswatson.com/static/
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

- **Response**



Response

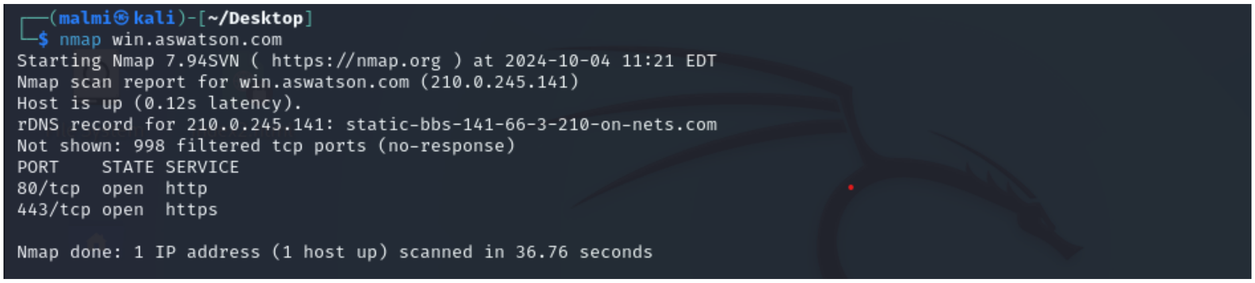
Response Time (ms) : 199.5912 Total Bytes Received : 84704 Body Length : 84349 Is Compressed : No

```
HTTP/1.1 200
Server: Apache
Content-Length: 29541
X-Frame-Options: SAMEORIGIN, SAMEORIGIN
Last-Modified: Fri, 03 Nov 2017 01:33:32 GMT
Accept-Ranges: bytes
Content-Type: text/javascript
Content-Encoding:
Date: Fri, 04 Oct 2024 09:48:14 GMT
Vary: Origin,Accept-Encoding,User-Agent,Access-Control-Request-Method,Access-Control-Re
…
bytes
Content-Type: text/javascript
Content-Encoding:
Date: Fri, 04 Oct 2024 09:48:14 GMT
Vary: Origin,Accept-Encoding,User-Agent,Access-Control-Request-Method,Access-Control-Request-Headers

/*! jQuery v2.1.4| (c) 2005, 2015 jQuery Foundation, Inc. | jquery.org/license */
!function(a,b){"object"==typeof module&&"object"==typeof module.exports?module.exports=a.document?b(a,!
0):function(a){if(!a.document)
…
```

Vulnerability finding using namp

Used command

- nmap win.aswatson.com



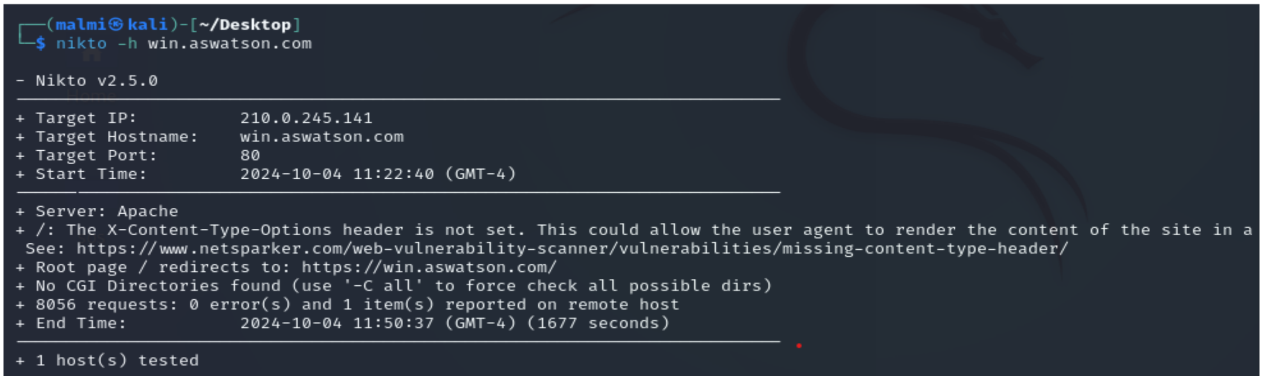
```
┌──(malmi㉿kali)-[~/Desktop]
└─$ nmap win.aswatson.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-04 11:21 EDT
Nmap scan report for win.aswatson.com (210.0.245.141)
Host is up (0.12s latency).
rDNS record for 210.0.245.141: static-bbs-141-66-3-210-on-nets.com
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 36.76 seconds
```

Founded open port

- 80/tcp open http
- 443/tcp open https

Vulnerability finding using Nikto

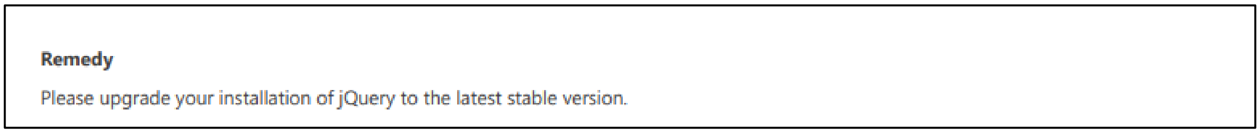
- **Used command**
    - nikto -h win.aswatson.com



```
┌──(malmi㉿kali)-[~/Desktop]
└─$ nikto -h win.aswatson.com

- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          210.0.245.141
+ Target Hostname:    win.aswatson.com
+ Target Port:        80
+ Start Time:         2024-10-04 11:22:40 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a
 See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Root page / redirects to: https://win.aswatson.com/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ 8056 requests: 0 error(s) and 1 item(s) reported on remote host
+ End Time:           2024-10-04 11:50:37 (GMT-4) (1677 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

- **Results**
    - The output from nikto indicates the absence of the X-Content-Type-Options header provides direct evidence of a vulnerability that could be exploited for MIME type sniffing and subsequent XSS attacks.

Proposed mitigation or fix



Remedy

Please upgrade your installation of jQuery to the latest stable version.

- **Upgrade jQuery**: Immediately update your jQuery installation to the latest stable version. These resolves known vulnerabilities associated with earlier versions.
- **Review Release Notes**: Check the release notes and documentation for the new version to understand what vulnerabilities were fixed and any potential breaking changes.
- **Test Compatibility**: After upgrading, conduct thorough testing to ensure that the updated version of jQuery works correctly with your web application without causing any issues.
- **Implement Security Practices**: Follow best practices for secure coding and web development, such as sanitizing user input and validating data before using it in DOM manipulation.
- **Regularly Monitor Dependencies**: Use tools to regularly scan and monitor your application for outdated libraries and frameworks, ensuring they remain up-to-date.
- **Consider Using a Package Manager**: Utilize package managers like npm or yarn to manage your JavaScript libraries, which can help automate the update process.

These steps will help you address the specific vulnerabilities associated with outdated jQuery versions. For more detailed guidance, you can check out resources like the jQuery documentation or security best practices from sources like OWASP.