SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY

Web Security- IE2062

Vulnerable and Outdated Components

A06:2021

Danuka Nuwan

IT22349842

Title - Vulnerable and Outdated Components

Vulnerability: The software uses library, components or frameworks that are all vulnerable and obsolete which exposes them to well-known security weaknesses. Bad actors can take advantage of this vulnerability by running code of their choice, making remote code execution (RCE) possible or even perform any other type of malicious activity thereby jeopardizing the confidentiality, integrity and availability of the application as well as its data.

Used website- https://www.kindredgroup.com/

Affected Components:

- Third-party libraries
- Frameworks
- Dependencies
- Plugins/extensions

Impact Assessment: High: The security position of an app can be gravely affected if it includes unsafe or outdated components because they may result in unauthorized access, system compromise, data loss among customers' trust.

Steps to Reproduce: Come up with a full list comprising names all third parties plus their libraries used within the app together with other necessary dependencies.

Make use public databases like National Vulnerability Database (NVD), CVE Details or OWASP Dependency-Check database while associating known vulnerabilities to each component.

Compare current versions against the most recent release so as to locate any outdated instance after establishing which is which for every component used.

You should scan your source codes through automated tools meant for detecting weak points caused by previous versions alongside those manual reviews where necessary.

Proof of Concept:

I used netsparker automation tool to find if there any out-of-date versions.

1. Out-of-date Version (Moment.js)

HIGH | 1

Netsparker identified that the target web site is using Moment.js and detected that it is out of date.

Impact

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

Moment.js Other Vulnerability

moment is a JavaScript date library for parsing, validating, manipulating, and formatting dates. Affected versions of moment were found to use an inefficient parsing algorithm. Specifically using string-to-date parsing in moment (more specifically rfc2822 parsing, which is tried by default) has quadratic (N^2) complexity on specific inputs. Users may notice a noticeable slowdown is observed with inputs above 10k characters. Users who pass user-provided strings without sanity length checks to moment constructor are vulnerable to (Re)DoS attacks. The problem is patched in 2.29.4, the patch can be applied to all affected versions with minimal tweaking. Users are advised to upgrade. Users unable to upgrade should consider limiting date lengths accepted from user input.

Affected Versions

2.18.0 to 2.29.3

External References

- CVE-2022-31129

Moment.js Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') Vulnerability

Moment.js is a JavaScript date library for parsing, validating, manipulating, and formatting dates. A path traversal vulnerability impacts npm (server) users of Moment.js between versions 1.0.1 and 2.29.1, especially if a user-provided locale string is directly used to switch moment locale. This problem is patched in 2.29.2, and the patch can be applied to all affected versions. As a workaround, sanitize the user-provided locale name before passing it to Moment.js.

Affected Versions

1.0.1 to 2.29.1

External References

- CVE-2022-24785

Vulnerabilities

1.1. https://www.kindredgroup.com/

Identified Version

- 2.24.0

Latest Version

- 2.30.1 (in this branch)

Request

```
GET / HTTP/1.1
Host: www.kindredgroup.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
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 Time (ms) : 515.5052 Total Bytes Received : 92412 Body Length : 91711 Is Compressed : No

```
HTTP/1.1 200 OK
Set-Cookie: EPiStateMarker=true; path=/
Set-Cookie: ARRAffinity=f62dc48792b6e16789f38b9331562ab71aac9ec805fac06e15282a091076b114;Path=/;HttpOnl
y;Secure;Domain=www.kindredgroup.com
Set-Cookie: ARRAffinitySameSite=f62dc48792b6e16789f38b9331562ab71aac9ec805fac06e15282a091076b114;Path
=/;HttpOnly;SameSite=None;Secure;Domain=www.kindredgroup.com
Request-Context: appId=cid-v1:f6f0367f-9d11-44c1-bd87-a01942e8c453
CF-RAY: 87b1cb230fc9513a-CMB
Server: cloudflare
CF-Cache-Status: DYNAMIC
Connection: keep-alive
Content-Encoding:
Strict-Transport-Security: max-age=2592000
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Date: Sat, 27 Apr 2024 21:09:25 GMT
```

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

In jQuery versions greater than or equal to 1.0.3 and before 3.5.0, passing HTML containing <option> elements from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.

Affected Versions

1.9.0 to 3.4.1

External References

- CVE-2020-11023

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

In jQuery versions greater than or equal to 1.2 and before 3.5.0, passing HTML from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.

Affected Versions

1.9.0 to 3.4.1

External References

- CVE-2020-11022

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

Cross Site Scripting vulnerability in jQuery 2.2.0 through 3.x before 3.5.0 allows a remote attacker to execute arbitrary code via the <options> element.

Affected Versions

2.2.0 to 3.4.1

External References

- CVE-2020-23064

Request

```
GET / HTTP/1.1
Host: www.kindredgroup.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
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 Time (ms) : 515.5052 Total Bytes Received : 92412 Body Length : 91711 Is Compressed : No

```
HTTP/1.1 200 OK
Set-Cookie: EPiStateMarker=true; path=/
Set-Cookie: ARRAffinity=f62dc48792b6e16789f38b9331562ab71aac9ec805fac06e15282a091076b114;Path=/;HttpOnl
y;Secure;Domain=www.kindredgroup.com
Set-Cookie: ARRAffinitySameSite=f62dc48792b6e16789f38b9331562ab71aac9ec805fac06e15282a091076b114;Path
=/;HttpOnly;SameSite=None;Secure;Domain=www.kindredgroup.com
Request-Context: appId=cid-v1:f6f0367f-9d11-44c1-bd87-a01942e8c453
CF-RAY: 87b1cb230fc9513a-CMB
Server: cloudflare
CF-Cache-Status: DYNAMIC
Connection: keep-alive
Content-Encoding:
Strict-Transport-Security: max-age=2592000
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Date: Sat, 27 Apr 2024 21:09:25 GMT
```

Suggested Resolution:

To deal with recognized vulnerabilities, always observe and upgrade third-party components to their newest versions.
Stay updated concerning precautionary measures and patches released for the specific components used by subscribing to security mailing lists or vendor notifications.

Employ tools that can automate dependency management. By doing this, you will be able to identify as well as update vulnerable parts within the codebase of an application.

Put in place runtime application self-protection mechanisms so that any attack meant for a weak component is detected and dealt with accordingly.

Scan for weaknesses frequently through security assessments such as penetration testing while at the same time patching them up. This should include those found within applications' dependencies too.

Make sure development teams understand why it's essential to have secure software dependencies which are up to date always. In addition, establish ways through which third-party components may be handled efficiently.