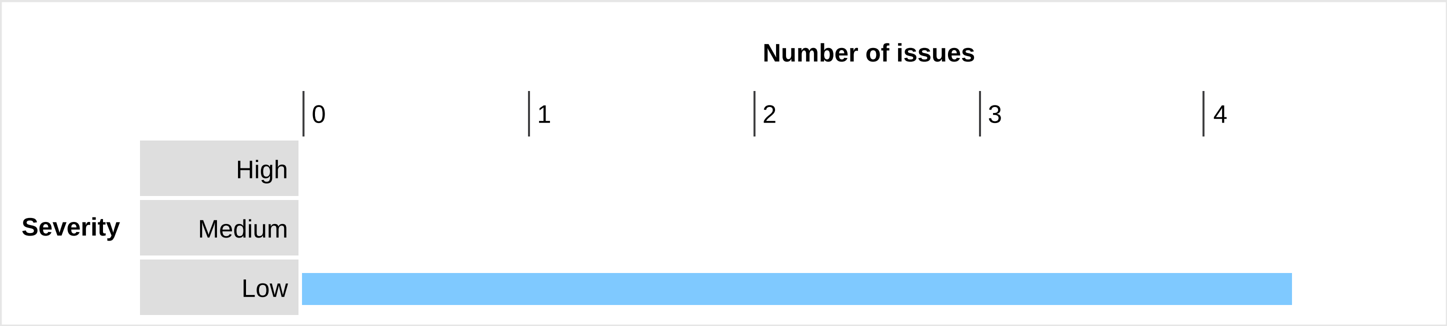
Beap Engine Report 3 sign up page

Summary

The table below shows the numbers of issues identified in different categories. Issues are classified according to severity as High, Medium, Low, Information or False Positive. This reflects the likely impact of each issue for a typical organization. Issues are also classified according to confidence as Certain, Firm or Tentative. This reflects the inherent reliability of the technique that was used to identify the issue.

		Confidence			
		Certain	Firm	Tentative	Total
Severity	High	**0**	**0**	**0**	**0**
	Medium	**0**	**0**	**0**	**0**
	Low	**0**	**4**	**0**	**4**
	Information	**0**	**0**	**0**	**0**
	False Positive	**0**	**0**	**0**	**0**

The chart below shows the aggregated numbers of issues identified in each category. Solid colored bars represent issues with a confidence level of Certain, and the bars fade as the confidence level falls.



Contents

1. Client-side JSON injection (DOM-based)

There are 4 instances of this issue:

- /signup
- /signup
- /signup
- /signup

Issue background

DOM-based vulnerabilities arise when a client-side script reads data from a controllable part of the DOM (for example, the URL) and processes this data in an unsafe way.

DOM-based JSON injection arises when a script incorporates controllable data into a string that is parsed as a JSON data structure and then processed by the application. An attacker may be able to use this behavior to construct a URL that, if visited by another application user, will cause arbitrary JSON data to be processed. Depending on the purpose for which this data is used, it may be possible to subvert the application's logic, or cause unintended actions on behalf of the user.

Burp Suite automatically identifies this issue using dynamic and static code analysis. Static analysis can lead to false positives that are not actually exploitable. If Burp Scanner has not provided any evidence resulting from dynamic analysis, you should review the relevant code and execution paths to determine whether this vulnerability is indeed present, or whether mitigations are in place that would prevent exploitation.

Issue remediation

The most effective way to avoid DOM-based JSON injection vulnerabilities is not to parse as JSON any string containing data that originated from an untrusted source. If the desired functionality of the application means that this behavior is unavoidable, then defenses must be implemented within the client-side code to prevent malicious data from modifying the JSON structure in inappropriate ways. This may involve strict validation of specific items to ensure they do not contain any characters that may interfere with the structure of the JSON when it is parsed.

References

- Web Security Academy: Client-side JSON injection (DOM-based)

Vulnerability classifications

- CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
- CWE-116: Improper Encoding or Escaping of Output
- CWE-159: Failure to Sanitize Special Element
- CAPEC-153: Input Data Manipulation

1.1. http://localhost:3000/signup

Summary

!	Severity:	**Low**
	Confidence:	**Firm**
	Host:	**http://localhost:3000**

	Path:	**/signup**

Issue detail

The application may be vulnerable to DOM-based client-side JSON injection. Data is read from **document.cookie** and passed to **JSON.parse**.

Because the data originates from a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

Request

```
GET /signup HTTP/1.1
Host: localhost:3000
Accept-Encoding: gzip, deflate, br
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.58 Safari/537.36
Connection: close
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Sec-CH-UA: ".Not/A)Brand";v="99", "Google Chrome";v="123", "Chromium";v="123"
Sec-CH-UA-Platform: Windows
Sec-CH-UA-Mobile: ?0
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: *
Access-Control-Allow-Headers: *
Content-Type: text/html; charset=utf-8
Accept-Ranges: bytes
ETag: W/"36b9-l+lbYeloQuWUVcszoOjyolkMmMU"
Vary: Accept-Encoding
Date: Thu, 04 Apr 2024 03:55:15 GMT
Connection: close
Content-Length: 14009

<!DOCTYPE html>
<html lang="en">
<head><script>try{(0,eval)("globalThis._triedToInstallGlobalErrorHandler") || (0,eval)("/* https://github.com/wallabyjs/console-ninja#how-does-it-work */'use strict'
...[SNIP]...
```

Dynamic analysis

Data is read from **document.cookie** and passed to **JSON.parse**.

The previous value reached the sink as:

```
cmysva5u1w%27%22`'"/cmysva5u1w/><cmysva5u1w/\>qxlicls5r4&
```

The stack trace at the source was:

```
at Object.<computed>.get (<anonymous>:1:624755)
at Cookies.update (http://localhost:3000/static/js/bundle.js:148822:73)
at Cookies.getAll (http://localhost:3000/static/js/bundle.js:148862:12)
at http://localhost:3000/static/js/bundle.js:119529:98
at mountState (http://localhost:3000/static/js/bundle.js:64033:24)
at Object.useState (http://localhost:3000/static/js/bundle.js:64638:20)
at useState (http://localhost:3000/static/js/bundle.js:85850:25)
at useCookies (http://localhost:3000/static/js/bundle.js:119529:83)
at useLogout (http://localhost:3000/static/js/bundle.js:7918:85)
at Navbar (http://localhost:3000/static/js/bundle.js:1326:74)
at renderWithHooks (http://localhost:3000/static/js/bundle.js:63454:22)
at mountIndeterminateComponent (http://localhost:3000/static/js/bundle.js:66796:23)
at beginWork (http://localhost:3000/static/js/bundle.js:68034:20)
at beginWork$1 (http://localhost:3000/static/js/bundle.js:72993:18)
at performUnitOfWork (http://localhost:3000/static/js/bundle.js:72263:16)
at workLoopSync (http://localhost:3000/static/js/bundle.js:72186:9)
at renderRootSync (http://localhost:3000/static/js/bundle.js:72159:11)
at performConcurrentWorkOnRoot (http://localhost:3000/static/js/bundle.js:71554:78)
at workLoop (http://localhost:3000/static/js/bundle.js:89985:38)
at flushWork (http://localhost:3000/static/js/bundle.js:89963:18)
at MessagePort.performWorkUntilDeadline (http://localhost:3000/static/js/bundle.js:90200:25)
```

The stack trace at the sink was:

```
at Object.WRdsg (<anonymous>:1:184730)
at Object.gmOtj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SHOuv (<anonymous>:1:178262)
at Object.iPcCb (<anonymous>:1:505498)
at _0x464a84.PcEKj._0x2950e0.<computed> (<anonymous>:1:528870)
at readCookie (http://localhost:3000/static/js/bundle.js:148797:19)
at http://localhost:3000/static/js/bundle.js:148841:18
at Set.forEach (<anonymous>)
at Cookies._checkChanges (http://localhost:3000/static/js/bundle.js:148837:11)
at Cookies.update (http://localhost:3000/static/js/bundle.js:148823:12)
at Cookies.getAll (http://localhost:3000/static/js/bundle.js:148862:12)
at http://localhost:3000/static/js/bundle.js:119529:98
at mountState (http://localhost:3000/static/js/bundle.js:64033:24)
at Object.useState (http://localhost:3000/static/js/bundle.js:64638:20)
at useState (http://localhost:3000/static/js/bundle.js:85850:25)
at useCookies (http://localhost:3000/static/js/bundle.js:119529:83)
at useLogout (http://localhost:3000/static/js/bundle.js:7918:85)
at Navbar (http://localhost:3000/static/js/bundle.js:1326:74)
at renderWithHooks (http://localhost:3000/static/js/bundle.js:63454:22)
at mountIndeterminateComponent (http://localhost:3000/static/js/bundle.js:66796:23)
at beginWork (http://localhost:3000/static/js/bundle.js:68034:20)
at beginWork$1 (http://localhost:3000/static/js/bundle.js:72993:18)
at performUnitOfWork (http://localhost:3000/static/js/bundle.js:72263:16)
at workLoopSync (http://localhost:3000/static/js/bundle.js:72186:9)
at renderRootSync (http://localhost:3000/static/js/bundle.js:72159:11)
at performConcurrentWorkOnRoot (http://localhost:3000/static/js/bundle.js:71554:78)
at workLoop (http://localhost:3000/static/js/bundle.js:89985:38)
at flushWork (http://localhost:3000/static/js/bundle.js:89963:18)
at MessagePort.performWorkUntilDeadline (http://localhost:3000/static/js/bundle.js:90200:25)
```

This was triggered by a **message** event.

1.2. http://localhost:3000/signup

Summary

Severity:	**Low**
Confidence:	**Firm**
Host:	**http://localhost:3000**
Path:	**/signup**

Issue detail

The application may be vulnerable to DOM-based client-side JSON injection. Data is read from **document.cookie** and passed to **JSON.parse**.

Because the data originates from a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

Request

```
GET /signup HTTP/1.1
Host: localhost:3000
Accept-Encoding: gzip, deflate, br
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.58 Safari/537.36
Connection: close
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Sec-CH-UA: ".Not/A)Brand";v="99", "Google Chrome";v="123", "Chromium";v="123"
Sec-CH-UA-Platform: Windows
Sec-CH-UA-Mobile: ?0
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: *
Access-Control-Allow-Headers: *
Content-Type: text/html; charset=utf-8
Accept-Ranges: bytes
ETag: W/"36b9-l+lbYeloQuWUVcszoOjyolkMmMU"
Vary: Accept-Encoding
Date: Thu, 04 Apr 2024 03:55:15 GMT
Connection: close
Content-Length: 14009

<!DOCTYPE html>
<html lang="en">
<head><script>try{(0,eval)("globalThis._triedToInstallGlobalErrorHandler") || (0,eval)("/* https://github.com/wallabyjs/console-ninja#how-does-it-work */'use strict'
...[SNIP]...
```

Dynamic analysis

Data is read from **document.cookie** and passed to **JSON.parse**.

The previous value reached the sink as:

```
ownxdtqlz6%27%22`'"/ownxdtqlz6/><ownxdtqlz6/\>h5dmcvmfco&
```

The stack trace at the source was:

```
at Object.<computed>.get (<anonymous>:1:624755)
at Cookies.update (http://localhost:3000/static/js/bundle.js:148822:73)
at Cookies.getAll (http://localhost:3000/static/js/bundle.js:148862:12)
at http://localhost:3000/static/js/bundle.js:119529:98
at mountState (http://localhost:3000/static/js/bundle.js:64033:24)
at Object.useState (http://localhost:3000/static/js/bundle.js:64638:20)
at useState (http://localhost:3000/static/js/bundle.js:85850:25)
at useCookies (http://localhost:3000/static/js/bundle.js:119529:83)
at useLogout (http://localhost:3000/static/js/bundle.js:7918:85)
at Navbar (http://localhost:3000/static/js/bundle.js:1326:74)
at renderWithHooks (http://localhost:3000/static/js/bundle.js:63454:22)
at mountIndeterminateComponent (http://localhost:3000/static/js/bundle.js:66738:17)
at beginWork (http://localhost:3000/static/js/bundle.js:68034:20)
at beginWork$1 (http://localhost:3000/static/js/bundle.js:72993:18)
at performUnitOfWork (http://localhost:3000/static/js/bundle.js:72263:16)
at workLoopSync (http://localhost:3000/static/js/bundle.js:72186:9)
at renderRootSync (http://localhost:3000/static/js/bundle.js:72159:11)
at performConcurrentWorkOnRoot (http://localhost:3000/static/js/bundle.js:71554:78)
at workLoop (http://localhost:3000/static/js/bundle.js:89985:38)
at flushWork (http://localhost:3000/static/js/bundle.js:89963:18)
at MessagePort.performWorkUntilDeadline (http://localhost:3000/static/js/bundle.js:90200:25)
```

The stack trace at the sink was:

```
at Object.WRdsg (<anonymous>:1:184730)
at Object.gmOtj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SHOuv (<anonymous>:1:178262)
at Object.iPcCb (<anonymous>:1:505498)
at _0x464a84.PcEKj._0x2950e0.<computed> (<anonymous>:1:528870)
at readCookie (http://localhost:3000/static/js/bundle.js:148797:19)
at http://localhost:3000/static/js/bundle.js:148841:18
at Set.forEach (<anonymous>)
at Cookies._checkChanges (http://localhost:3000/static/js/bundle.js:148837:11)
at Cookies.update (http://localhost:3000/static/js/bundle.js:148823:12)
at Cookies.getAll (http://localhost:3000/static/js/bundle.js:148862:12)
at http://localhost:3000/static/js/bundle.js:119529:98
at mountState (http://localhost:3000/static/js/bundle.js:64033:24)
at Object.useState (http://localhost:3000/static/js/bundle.js:64638:20)
at useState (http://localhost:3000/static/js/bundle.js:85850:25)
at useCookies (http://localhost:3000/static/js/bundle.js:119529:83)
at useLogout (http://localhost:3000/static/js/bundle.js:7918:85)
at Navbar (http://localhost:3000/static/js/bundle.js:1326:74)
at renderWithHooks (http://localhost:3000/static/js/bundle.js:63454:22)
at mountIndeterminateComponent (http://localhost:3000/static/js/bundle.js:66738:17)
at beginWork (http://localhost:3000/static/js/bundle.js:68034:20)
at beginWork$1 (http://localhost:3000/static/js/bundle.js:72993:18)
at performUnitOfWork (http://localhost:3000/static/js/bundle.js:72263:16)
at workLoopSync (http://localhost:3000/static/js/bundle.js:72186:9)
at renderRootSync (http://localhost:3000/static/js/bundle.js:72159:11)
at performConcurrentWorkOnRoot (http://localhost:3000/static/js/bundle.js:71554:78)
at workLoop (http://localhost:3000/static/js/bundle.js:89985:38)
at flushWork (http://localhost:3000/static/js/bundle.js:89963:18)
at MessagePort.performWorkUntilDeadline (http://localhost:3000/static/js/bundle.js:90200:25)
```

This was triggered by a **message** event.

1.3. http://localhost:3000/signup

Summary

!	Severity:	**Low**
	Confidence:	**Firm**
	Host:	**http://localhost:3000**
	Path:	**/signup**

Issue detail

The application may be vulnerable to DOM-based client-side JSON injection. Data is read from **document.cookie** and passed to **JSON.parse**.

Because the data originates from a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

Request

```
GET /signup HTTP/1.1
Host: localhost:3000
Accept-Encoding: gzip, deflate, br
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.58 Safari/537.36
Connection: close
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Sec-CH-UA: ".Not/A)Brand";v="99", "Google Chrome";v="123", "Chromium";v="123"
Sec-CH-UA-Platform: Windows
Sec-CH-UA-Mobile: ?0
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: *
Access-Control-Allow-Headers: *
Content-Type: text/html; charset=utf-8
Accept-Ranges: bytes
ETag: W/"36b9-l+lbYeloQuWUVcszoOjyolkMmMU"
Vary: Accept-Encoding
Date: Thu, 04 Apr 2024 03:55:15 GMT
Connection: close
Content-Length: 14009

<!DOCTYPE html>
<html lang="en">
<head><script>try{(0,eval)("globalThis._triedToInstallGlobalErrorHandler") || (0,eval)("/* https://github.com/wallabyjs/console-
```

ninja#how-does-it-work */'use strict'
...[SNIP]...

Dynamic analysis

Data is read from **document.cookie** and passed to **JSON.parse**.

The previous value reached the sink as:

```
d43u3upzpa%27%22`'"/d43u3upzpa/><d43u3upzpa/\>hyp8kmjszd&
```

The stack trace at the source was:

```
at Object.<computed>.get (<anonymous>:1:624755)
at Cookies.update (http://localhost:3000/static/js/bundle.js:148822:73)
at Cookies.getAll (http://localhost:3000/static/js/bundle.js:148862:12)
at http://localhost:3000/static/js/bundle.js:119529:98
at mountState (http://localhost:3000/static/js/bundle.js:64033:24)
at Object.useState (http://localhost:3000/static/js/bundle.js:64638:20)
at useState (http://localhost:3000/static/js/bundle.js:85850:25)
at useCookies (http://localhost:3000/static/js/bundle.js:119529:83)
at useIsUserLoggedIn (http://localhost:3000/static/js/bundle.js:7737:85)
at AuthProvider (http://localhost:3000/static/js/bundle.js:416:101)
at renderWithHooks (http://localhost:3000/static/js/bundle.js:63454:22)
at mountIndeterminateComponent (http://localhost:3000/static/js/bundle.js:66796:23)
at beginWork (http://localhost:3000/static/js/bundle.js:68034:20)
at beginWork$1 (http://localhost:3000/static/js/bundle.js:72993:18)
at performUnitOfWork (http://localhost:3000/static/js/bundle.js:72263:16)
at workLoopSync (http://localhost:3000/static/js/bundle.js:72186:9)
at renderRootSync (http://localhost:3000/static/js/bundle.js:72159:11)
at performConcurrentWorkOnRoot (http://localhost:3000/static/js/bundle.js:71554:78)
at workLoop (http://localhost:3000/static/js/bundle.js:89985:38)
at flushWork (http://localhost:3000/static/js/bundle.js:89963:18)
at MessagePort.performWorkUntilDeadline (http://localhost:3000/static/js/bundle.js:90200:25)
```

The stack trace at the sink was:

```
at Object.WRdsg (<anonymous>:1:184730)
at Object.gmOtj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SHOuv (<anonymous>:1:178262)
at Object.iPcCb (<anonymous>:1:505498)
at _0x464a84.PcEKj._0x2950e0.<computed> (<anonymous>:1:528870)
at readCookie (http://localhost:3000/static/js/bundle.js:148797:19)
at http://localhost:3000/static/js/bundle.js:148841:18
at Set.forEach (<anonymous>)
at Cookies._checkChanges (http://localhost:3000/static/js/bundle.js:148837:11)
at Cookies.update (http://localhost:3000/static/js/bundle.js:148823:12)
at Cookies.getAll (http://localhost:3000/static/js/bundle.js:148862:12)
at http://localhost:3000/static/js/bundle.js:119529:98
at mountState (http://localhost:3000/static/js/bundle.js:64033:24)
at Object.useState (http://localhost:3000/static/js/bundle.js:64638:20)
at useState (http://localhost:3000/static/js/bundle.js:85850:25)
at useCookies (http://localhost:3000/static/js/bundle.js:119529:83)
at useIsUserLoggedIn (http://localhost:3000/static/js/bundle.js:7737:85)
at AuthProvider (http://localhost:3000/static/js/bundle.js:416:101)
at renderWithHooks (http://localhost:3000/static/js/bundle.js:63454:22)
at mountIndeterminateComponent (http://localhost:3000/static/js/bundle.js:66796:23)
at beginWork (http://localhost:3000/static/js/bundle.js:68034:20)
at beginWork$1 (http://localhost:3000/static/js/bundle.js:72993:18)
at performUnitOfWork (http://localhost:3000/static/js/bundle.js:72263:16)
at workLoopSync (http://localhost:3000/static/js/bundle.js:72186:9)
at renderRootSync (http://localhost:3000/static/js/bundle.js:72159:11)
at performConcurrentWorkOnRoot (http://localhost:3000/static/js/bundle.js:71554:78)
at workLoop (http://localhost:3000/static/js/bundle.js:89985:38)
at flushWork (http://localhost:3000/static/js/bundle.js:89963:18)
at MessagePort.performWorkUntilDeadline (http://localhost:3000/static/js/bundle.js:90200:25)
```

This was triggered by a **message** event.

1.4. http://localhost:3000/signup

Summary

Severity:	**Low**
Confidence:	**Firm**
Host:	**http://localhost:3000**
Path:	**/signup**

Issue detail

The application may be vulnerable to DOM-based client-side JSON injection. Data is read from **document.cookie** and passed to **JSON.parse**.

Because the data originates from a cookie, the application's behavior is not trivial to exploit in an attack against another user. Typically, you will need to find a means of setting an arbitrary cookie value in the victim's browser in order to exploit the vulnerability. Applications often contain "cookie-forcing" conditions which make this possible, and such a condition in any related domain or subdomain can potentially be used for this purpose. Nonetheless, this limitation somewhat mitigates the impact of the vulnerability.

Request

```
GET /signup HTTP/1.1
Host: localhost:3000
Accept-Encoding: gzip, deflate, br
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.58 Safari/537.36
Connection: close
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Sec-CH-UA: ".Not/A)Brand";v="99", "Google Chrome";v="123", "Chromium";v="123"
Sec-CH-UA-Platform: Windows
Sec-CH-UA-Mobile: ?0
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: *
Access-Control-Allow-Headers: *
Content-Type: text/html; charset=utf-8
Accept-Ranges: bytes
ETag: W/"36b9-I+lbYeloQuWUVcszoOjyoIkMmMU"
```

```
Vary: Accept-Encoding
Date: Thu, 04 Apr 2024 03:55:15 GMT
Connection: close
Content-Length: 14009

<!DOCTYPE html>
<html lang="en">
<head><script>try{(0,eval)("globalThis._triedToInstallGlobalErrorHandler") || (0,eval)("/* https://github.com/wallabyjs/console-ninja#how-does-it-work */'use strict'
...[SNIP]...
```

Dynamic analysis

Data is read from **document.cookie** and passed to **JSON.parse**.

The previous value reached the sink as:

```
se6opr90vs%27%22`'"/se6opr90vs/><se6opr90vs/\>x6eekpxytf&
```

The stack trace at the source was:

```
at Object.<computed>.get (<anonymous>:1:624755)
at Cookies.update (http://localhost:3000/static/js/bundle.js:148822:73)
at Cookies.getAll (http://localhost:3000/static/js/bundle.js:148862:12)
at http://localhost:3000/static/js/bundle.js:119529:98
at mountState (http://localhost:3000/static/js/bundle.js:64033:24)
at Object.useState (http://localhost:3000/static/js/bundle.js:64638:20)
at useState (http://localhost:3000/static/js/bundle.js:85850:25)
at useCookies (http://localhost:3000/static/js/bundle.js:119529:83)
at useIsUserLoggedIn (http://localhost:3000/static/js/bundle.js:7737:85)
at AuthProvider (http://localhost:3000/static/js/bundle.js:416:101)
at renderWithHooks (http://localhost:3000/static/js/bundle.js:63454:22)
at mountIndeterminateComponent (http://localhost:3000/static/js/bundle.js:66738:17)
at beginWork (http://localhost:3000/static/js/bundle.js:68034:20)
at beginWork$1 (http://localhost:3000/static/js/bundle.js:72993:18)
at performUnitOfWork (http://localhost:3000/static/js/bundle.js:72263:16)
at workLoopSync (http://localhost:3000/static/js/bundle.js:72186:9)
at renderRootSync (http://localhost:3000/static/js/bundle.js:72159:11)
at performConcurrentWorkOnRoot (http://localhost:3000/static/js/bundle.js:71554:78)
at workLoop (http://localhost:3000/static/js/bundle.js:89985:38)
at flushWork (http://localhost:3000/static/js/bundle.js:89963:18)
at MessagePort.performWorkUntilDeadline (http://localhost:3000/static/js/bundle.js:90200:25)
```

The stack trace at the sink was:

```
at Object.WRdsg (<anonymous>:1:184730)
at Object.gmOtj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SHOuv (<anonymous>:1:178262)
at Object.iPcCb (<anonymous>:1:505498)
at _0x464a84.PcEKj._0x2950e0.<computed> (<anonymous>:1:528870)
at readCookie (http://localhost:3000/static/js/bundle.js:148797:19)
at http://localhost:3000/static/js/bundle.js:148841:18
at Set.forEach (<anonymous>)
at Cookies._checkChanges (http://localhost:3000/static/js/bundle.js:148837:11)
at Cookies.update (http://localhost:3000/static/js/bundle.js:148823:12)
at Cookies.getAll (http://localhost:3000/static/js/bundle.js:148862:12)
at http://localhost:3000/static/js/bundle.js:119529:98
at mountState (http://localhost:3000/static/js/bundle.js:64033:24)
at Object.useState (http://localhost:3000/static/js/bundle.js:64638:20)
at useState (http://localhost:3000/static/js/bundle.js:85850:25)
at useCookies (http://localhost:3000/static/js/bundle.js:119529:83)
at useIsUserLoggedIn (http://localhost:3000/static/js/bundle.js:7737:85)
at AuthProvider (http://localhost:3000/static/js/bundle.js:416:101)
at renderWithHooks (http://localhost:3000/static/js/bundle.js:63454:22)
at mountIndeterminateComponent (http://localhost:3000/static/js/bundle.js:66738:17)
at beginWork (http://localhost:3000/static/js/bundle.js:68034:20)
at beginWork$1 (http://localhost:3000/static/js/bundle.js:72993:18)
at performUnitOfWork (http://localhost:3000/static/js/bundle.js:72263:16)
at workLoopSync (http://localhost:3000/static/js/bundle.js:72186:9)
at renderRootSync (http://localhost:3000/static/js/bundle.js:72159:11)
at performConcurrentWorkOnRoot (http://localhost:3000/static/js/bundle.js:71554:78)
at workLoop (http://localhost:3000/static/js/bundle.js:89985:38)
at flushWork (http://localhost:3000/static/js/bundle.js:89963:18)
at MessagePort.performWorkUntilDeadline (http://localhost:3000/static/js/bundle.js:90200:25)
```

This was triggered by a **message** event.

Report generated by Burp Suite web vulnerability scanner v2024.2.1.3, at Wed Apr 03 22:02:18 CST 2024.