

Beap Engine Report 2

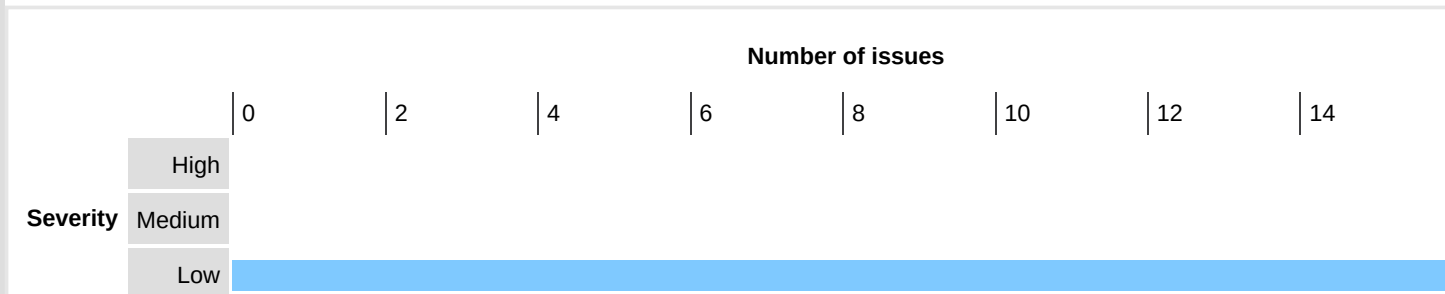


Summary

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	16	0	16
	Information	3	0	0	3
	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)

- 1.1. <http://localhost:3000/>
- 1.2. <http://localhost:3000/>
- 1.3. <http://localhost:3000/>
- 1.4. <http://localhost:3000/>
- 1.5. <http://localhost:3000/about-us>
- 1.6. <http://localhost:3000/about-us>
- 1.7. <http://localhost:3000/about-us>
- 1.8. <http://localhost:3000/about-us>
- 1.9. <http://localhost:3000/login>
- 1.10. <http://localhost:3000/login>
- 1.11. <http://localhost:3000/login>
- 1.12. <http://localhost:3000/login>
- 1.13. <http://localhost:3000/privacy-policy>
- 1.14. <http://localhost:3000/privacy-policy>
- 1.15. <http://localhost:3000/privacy-policy>

1.16. <http://localhost:3000/privacy-policy>

2. Email addresses disclosed

3. Private IP addresses disclosed

4. Credit card numbers disclosed

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

There are 16 instances of this issue:

- /
- /
- /
- /
- /about-us
- /about-us
- /about-us
- /about-us
- /login
- /login
- /login
- /login
- /privacy-policy
- /privacy-policy
- /privacy-policy
- /privacy-policy

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/

Summary

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

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 / 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-MJaGmruv93asyHCM4bLub6uFyDk"
Vary: Accept-Encoding
Date: Thu, 04 Apr 2024 02:47:22 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:

```
f9g12ayxz6%27%22`"/f9g12ayxz6/><f9g12ayxz6/\>gml5mt4mst&
```

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.gm0tj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SH0uv (<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/

Summary

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

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 / 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-MJaGmruv93asyHCM4bLub6uFyDk"
Vary: Accept-Encoding
Date: Thu, 04 Apr 2024 02:47:22 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:

```
ckgp6u7ql8%27%22`"/ckgp6u7ql8/><ckgp6u7ql8/\>ge6gamydeb&
```

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.gm0tj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SH0uv (<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/

Summary

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

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 / 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-MJaGmruv93asyHCM4bLub6uFyDk"
Vary: Accept-Encoding
Date: Thu, 04 Apr 2024 02:47:22 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:

```
xddpsqi5tl%27%22`"/>xddpsqi5tl/><xddpsqi5tl/\>ju97d0l815&
```


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.gm0tj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SH0uv (<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/

Summary



Severity:

Low

Confidence:	Firm
Host:	http://localhost:3000
Path:	/

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 / 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-MJaGmruv93asyHCM4bLub6uFyDk"
Vary: Accept-Encoding
Date: Thu, 04 Apr 2024 02:47:22 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:

pnnqge8pah%27%22` ' " /pnnqge8pah/><pnnqge8pah/\>wfl10uojar&

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)
```

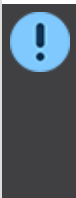
The stack trace at the sink was:

```
at Object.WRdsg (<anonymous>:1:184730)
at Object.gm0tj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SH0uv (<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.

1.5. http://localhost:3000/about-us

Summary

	Severity:	Low
	Confidence:	Firm
	Host:	http://localhost:3000
	Path:	/about-us

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 /about-us 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
Referer: http://localhost:3000/
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 02:56:51 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:

```
e5rneduxoh%27%22`"/e5rneduxoh/><e5rneduxoh/\>y5ilhpg79z&
```

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.gm0tj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SH0uv (<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.

1.6. http://localhost:3000/about-us

Summary

	Severity:	Low
	Confidence:	Firm
	Host:	http://localhost:3000
	Path:	/about-us

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 /about-us 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
Referer: http://localhost:3000/
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 02:56:51 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:

```
fwfwztnmjo%27%22`"/fwfwztnmjo/><fwfwztnmjo/\>efx17kv1hn&
```

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.gm0tj (<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.

1.7. http://localhost:3000/about-us

Summary

	Severity:	Low
	Confidence:	Firm
	Host:	http://localhost:3000
	Path:	/about-us

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 /about-us 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
Referer: http://localhost:3000/
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 02:56:51 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:

```
qv0abqq0hn%27%22` `"/qv0abqq0hn/><qv0abqq0hn/\>zefd49f6ki&
```

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.gm0tj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SH0uv (<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.

1.8. http://localhost:3000/about-us

Summary

	Severity:	Low
	Confidence:	Firm
	Host:	http://localhost:3000
	Path:	/about-us

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 /about-us 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
Referer: http://localhost:3000/
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 02:56:51 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:

```
eq5029ccvk%27%22`"/eq5029ccvk/><eq5029ccvk/\>wq7ivadsd&
```

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.gm0tj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SH0uv (<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.

1.9. http://localhost:3000/login

Summary

	Severity:	Low
	Confidence:	Firm
	Host:	http://localhost:3000
	Path:	/login

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 /login 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 02:56:56 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:

```
ibj987prlg%27%22`"/ibj987prlg/><ibj987prlg/\>t5ufze9sam&
```

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.gm0tj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SH0uv (<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.

1.10. http://localhost:3000/login

Summary

	Severity:	Low
	Confidence:	Firm
	Host:	http://localhost:3000
	Path:	/login

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 /login 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+IbYeloQuWUVcszoOjyolkMmMU"
Vary: Accept-Encoding
Date: Thu, 04 Apr 2024 02:56:56 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:

```
jfliec7a5s%27%22`'"/jfliec7a5s/><jfliec7a5s/\>o2a4sd2zm5&
```

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.gm0tj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
```




```
at Object.SH0uv (<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.

1.11. http://localhost:3000/login

Summary

	Severity:	Low
	Confidence:	Firm
	Host:	http://localhost:3000
	Path:	/login

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 /login 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 02:56:56 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:

```
cuu8hllyq7%27%22`'"/cuu8hllyq7/><cuu8hllyq7/\>xsamfw9wwu&
```

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.gm0tj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SH0uv (<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.

1.12. http://localhost:3000/login

Summary

	Severity:	Low
	Confidence:	Firm
	Host:	http://localhost:3000
	Path:	/login

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 /login 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 02:56:56 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:

```
ufpsr62lex%27%22`"/ufpsr62lex/><ufpsr62lex/\>fed1f50qg7&
```

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.gm0tj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SH0uv (<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.

1.13. http://localhost:3000/privacy-policy

Summary

	Severity:	Low
	Confidence:	Firm
	Host:	http://localhost:3000
	Path:	/privacy-policy

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 /privacy-policy 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
Referer: http://localhost:3000/
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-MJaGmruv93asyHCM4bLub6uFyDk"
Vary: Accept-Encoding
Date: Thu, 04 Apr 2024 01:33:24 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:

```
twzi7veikp%27%22`"/twzi7veikp/><twzi7veikp/\>e6hjm21o3b&
```

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.gm0tj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SH0uv (<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.

1.14. http://localhost:3000/privacy-policy

Summary

	Severity:	Low
	Confidence:	Firm
	Host:	http://localhost:3000
	Path:	/privacy-policy

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 /privacy-policy 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
Referer: http://localhost:3000/
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-MJaGmruv93asyHCM4bLub6uFyDk"
Vary: Accept-Encoding
Date: Thu, 04 Apr 2024 01:33:24 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:

```
k85n6luxv1%27%22`"/k85n6luxv1/><k85n6luxv1/\>cajb1m1dpk&
```

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.gm0tj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at _Object.SH0uv (<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.

1.15. http://localhost:3000/privacy-policy

Summary

	Severity:	Low
	Confidence:	Firm
	Host:	http://localhost:3000
	Path:	/privacy-policy

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 /privacy-policy 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
Referer: http://localhost:3000/
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-MJaGmruv93asyHCM4bLub6uFyDk"
```

```
Vary: Accept-Encoding
Date: Thu, 04 Apr 2024 01:33:24 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:

```
f06jrutdkf%27%22`'"/f06jrutdkf/><f06jrutdkf/\>osumzxc0sr&
```

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:

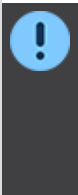
```
at Object.WRdsg (<anonymous>:1:184730)
at Object.gm0tj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SH0uv (<anonymous>:1:178262)
at Object.iPcCb (<anonymous>:1:505498)
at _0x464a84.PcEKj._0x2950e0.<computed> (<anonymous>:1:528870)
at readCookie (http://localhost:3000/static/js/bundle.js:148797:19)
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.

1.16. http://localhost:3000/privacy-policy

Summary

	Severity:	Low
	Confidence:	Firm
	Host:	http://localhost:3000
	Path:	/privacy-policy

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 /privacy-policy 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
Referer: http://localhost:3000/
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-MJaGmruv93asyHCM4bLub6uFyDk"
Vary: Accept-Encoding
Date: Thu, 04 Apr 2024 01:33:24 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:

```
nwtlcxjufu%27%22`"/nwtlcxjufu/><nwtlcxjufu/\>zgkmxxv8nq&
```

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.gm0tj (<anonymous>:1:612095)
at _0x2f9cd6 (<anonymous>:1:627659)
at Object.SH0uv (<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.

2. Email addresses disclosed

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost:3000
	Path:	/static/js/bundle.js

Issue detail

The following email addresses were disclosed in the response:

- daniel.fuller@usask.ca
- icehr@mun.ca

Issue background

The presence of email addresses within application responses does not necessarily constitute a security vulnerability. Email addresses may appear intentionally within contact information, and many applications (such as web mail) include arbitrary third-party email addresses within their core content.

However, email addresses of developers and other individuals (whether appearing on-screen or hidden within page source) may disclose information that is useful to an attacker; for example, they may represent usernames that can be used at the application's login, and they may be used in social engineering attacks against the organization's personnel. Unnecessary or excessive disclosure of email addresses may also lead to an increase in the volume of spam email received.

Issue remediation

Consider removing any email addresses that are unnecessary, or replacing personal addresses with anonymous mailbox addresses (such as helpdesk@example.com).

To reduce the quantity of spam sent to anonymous mailbox addresses, consider hiding the email address and instead providing a form that generates the email server-side, protected by a CAPTCHA if necessary.

References

- Web Security Academy: Information disclosure

Vulnerability classifications

- CWE-200: Information Exposure
- CAPEC-37: Retrieve Embedded Sensitive Data

Request

```
GET /static/js/bundle.js 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: application/javascript; charset=utf-8
Accept-Ranges: bytes
ETag: W/"6c0a27-9NMUiBziV/bym7NTvcbC7oxGIXU"
Vary: Accept-Encoding
Date: Thu, 04 Apr 2024 02:57:24 GMT
Connection: close
Content-Length: 7080487

```

/*****/ (() => { // webpackBootstrap
/*****/   var __webpack_modules__ = ({

/***/ ".src/App.tsx":
/*!*****!*\
!*** ./src/App.tsx ***!
\******/
/***/ ((module, __w
...[SNIP]...
PACK_IMPORTED_MODULE_2__["default"].text,
children: ["Researcher(s): Dr. Daniel Fuller (",
/*# __PURE__ */(0,react_jsx_dev_runtime__WEBPACK_IMPORTED_MODULE_4___jsxDEV)("a", {
href: "mailto:daniel.fuller@usask.ca",
className: _PrivacyPolicy_module_css__WEBPACK_IMPORTED_MODULE_2__["default"].link,
children: "daniel.fuller@usask.ca"
}, void 0, false, {
fileName: _jsxFileName,
lineNumber: 26,
columnNumber: 17
}, this), ";", /*# __PURE__ */(0,react_jsx_dev_runtime__WEBPACK_IMPORTED_MODULE_4___jsxDEV
...[SNIP]...
site to improve functionality. If you experience an error please contact Dr. Daniel Fuller (",
/*# __PURE__ */(0,react_jsx_dev_runtime__WEBPACK_IMPORTED_MODULE_3___jsxDEV)("a", {
href: "mailto:daniel.fuller@usask.ca",
className: _AboutUs_module_css__WEBPACK_IMPORTED_MODULE_1__["default"].link,
children: "daniel.fuller@usask.ca"
}, void 0, false, {
fileName: _jsxFileName,
lineNumber: 155,
columnNumber: 21
}, this), ") with a screenshot and description of the error."
}, voi
...[SNIP]...
```

3. Private IP addresses disclosed

Summary



Severity:

Information

Confidence:	Certain
Host:	http://localhost:3000
Path:	/static/js/bundle.js

Issue detail

The following RFC 1918 IP addresses were disclosed in the response:

- 10.0.0.248
- 10.1.9.34
- 10.8.1.1
- 172.18.0.1

Issue background

RFC 1918 specifies ranges of IP addresses that are reserved for use in private networks and cannot be routed on the public Internet. Although various methods exist by which an attacker can determine the public IP addresses in use by an organization, the private addresses used internally cannot usually be determined in the same ways.

Discovering the private addresses used within an organization can help an attacker in carrying out network-layer attacks aiming to penetrate the organization's internal infrastructure.

Issue remediation

There is not usually any good reason to disclose the internal IP addresses used within an organization's infrastructure. If these are being returned in service banners or debug messages, then the relevant services should be configured to mask the private addresses. If they are being used to track back-end servers for load balancing purposes, then the addresses should be rewritten with innocuous identifiers from which an attacker cannot infer any useful information about the infrastructure.

References

- [Web Security Academy: Information disclosure](#)

Vulnerability classifications

- [CWE-200: Information Exposure](#)
- [CAPEC-37: Retrieve Embedded Sensitive Data](#)

Request

GET /static/js/bundle.js 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: application/javascript; charset=utf-8
Accept-Ranges: bytes


```
ETag: W/"6c0a27-9NMUiBziV/bym7NTvcbC7oxGIXU"
Vary: Accept-Encoding
Date: Thu, 04 Apr 2024 02:57:24 GMT
Connection: close
Content-Length: 7080487


/*****/ (() => { // webpackBootstrap
/*****/   var __webpack_modules__ = ({

/***/ "./src/App.tsx":
/*!*****!*\
!*** ./src/App.tsx ***!
\*****!
/***/ (module, __w
...[SNIP]...
2ff['_console_ninja'];})
(globalThis,_0x24ae32(0x1b0),_0x24ae32(0xd6),_0x24ae32(0x11d),_0x24ae32(0x10e),_0x24ae32(0x17a),'1712198927630',
[["localhost","127.0.0.1","example.cypress.io","GlennOS","10.0.0.248","172.18.0.1"],_0x24ae32(0x1ad),_0x24ae32(0x11e));
} catch (e) {}
}
; /* istanbul ignore next */
function oo_oo(i, ...v) {
try {
oo_cm().consoleLog(i, v);
} catch (e) {}
return v;
}

...[SNIP]...
.47 2 7.2 3.26 8.28 2.13 1.89-1.94-3.48-9.39-12.12-13.09a31.44 31.44 0 0 0-30.61 3.68c-3 2.18-5.81 5.22-5.41 7.06.85 3.74 10-2.71
22.6-3.48 7-.44 12.8 1.75 17.26 3.71zm-9 5.13c-9.07 1.42-15 6.53-13.47 10.1.9.34 1.17.81 5.21-.81a37 37 0 0 1 18.72-1.95c2.92.34
4.31.52 4.94-.49 1.46-2.22-5.71-8-15.39-6.85zm54.17 17.1c3.38-6.87-10.9-13.93-14.3-7s10.92 13.88 14.32 6.97zm15.66-20.47c-
7.66-.13-7.95 15.8-.26 15.93s7
...[SNIP]...
6 6.3-63.4-21.3-79.4-17.8-10.2-.6-.4-18.6-10.6-24.6-14.2-3.4-51.9 21.6-37.5 18.6 10.8-.1-.1 18.5 10.7 48.4 28 65.1 90.3 37.2
138.5zm21.8-208.8c-17 29.5-16.3 28.8-19 31.5-6.5 6.5-16.3 8.7-26.5 3.6-18.6-10.8.1.1-18.5-10.7-27.6-15.9-63.4-6.3-79.4 21.3s-6.3 63.4
21.3 79.4c0 0 18.5 10.6 18.6 10.6 24.6 14.2 3.4 51.9-21.6 37.5-18.6-10.8.1.1-18.5-10.7-48.2-27.8-64.9-90.1-37.1-138.4 27.9-48.2 89.9-
64.9 138.2-37.2l4.8-8.4c14.3-24.9 52-3.3 37.7 21.5z"
},
"child": []
})
})(props);
}
;
function FaSellsy(props) {
return (0,_
...[SNIP]...
```

4. Credit card numbers disclosed

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	http://localhost:3000
	Path:	/static/js/bundle.js

Issue detail

The following credit card numbers were disclosed in the response:

- 5224224224224100

- 4312002000110
- 5601056565656000
- 4114414414414464

Issue background

Applications sometimes disclose sensitive financial information such as credit card numbers. Responses containing credit card numbers may not represent any security vulnerability - for example, a number may belong to the logged-in user to whom it is displayed. If a credit card number is identified during a security assessment it should be verified, then application logic reviewed to identify whether its disclosure within the application is necessary and appropriate.

References

- [Web Security Academy: Information disclosure](#)

Vulnerability classifications

- [CWE-200: Information Exposure](#)
- [CWE-388: Error Handling](#)
- [CAPEC-37: Retrieve Embedded Sensitive Data](#)

Request

```
GET /static/js/bundle.js 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: application/javascript; charset=utf-8
Accept-Ranges: bytes
ETag: W/"6c0a27-9NMUiBziV/bym7NTvcB7oxGIXU"
Vary: Accept-Encoding
Date: Thu, 04 Apr 2024 02:57:24 GMT
Connection: close
Content-Length: 7080487

/*****/ (() => { // webpackBootstrap
/*****/  var __webpack_modules__ = ({

/***/ "./src/App.tsx":
/*!*****!*\
!*** ./src/App.tsx ***!
\*****/
/***/ ((module, __w
...[SNIP]...
28.9 11 40 0 6.3-6.3 9-14.9 8.1-23.1175.2-88.8c6.3-6.5-3.3-15.9-9.5-9.6zm-83.8 111.5c-5.6 5.5-14.6 5.5-20.2 0-5.6-5.6-5.6-14.6 0-
20.2s14.6-5.6 20.2 0 5.6 14.7 0 20.2zM224 32C100.5 32 0 132.5 0 256s100.5 224 224 224-100.5 224-224S347.5 32 224 32zm0
384c-88.2 0-160-71.8-160-160S135.8 96 224 96s160 71.8 160 160-71.8 160-160 160z"
},
"child": []
```

```
}}
})(props);
}
;
function FaCloudsmith(props) {
return
...[SNIP]...
"tag": "path",
"attr": {
"d": "M512 256c0 141.2-114.7 256-256 256C114.8 512 0 397.3 0 256S114.7 0 256 0s256 114.7 256 256zm-32 0c0-123.2-100.3-224-224-224C132.5 32 32 132.5 32 256s100.5 224 224 224 224-100.5 224-224zM160.9 124.6l86.9 37.1-37.1 86.9-86.9-37.1 37.1-86.9zm110 169.1l46.6 94h-14.6l-50-100-48.9 100h-14l51.1-106.9-22.3-9.4 6-14 68.6 29.1-6 14.3-16.5-7.1zm-11.8-116.3l68.6 29.4-29.4 68.3L230 246
...[SNIP]...
"child": [{
"tag": "path",
"attr": {
"d": "M256 8C119.043 8 8 119.083 8 256c0 136.997 111.043 248 248 248s248-111.003 248-248C504 119.083 392.957 8 256 8zm0 448c-110.532 0-200-89.431-200-89.431-200-89.431 0-200-89.431 200 200 0 110.53-89.431 200-200 200zm107.244-255.2c0 67.052-72.421 68.084-72.421 92.863V300c0 6.627-5.373 12-12 12h-45.647c-6.627 0-12-5.373-12-12v-8.659c0-
...[SNIP]...
```

Report generated by Burp Suite [web vulnerability scanner](#) v2024.2.1.3, at Wed Apr 03 21:47:44 CST 2024.