Burp Scanner Report

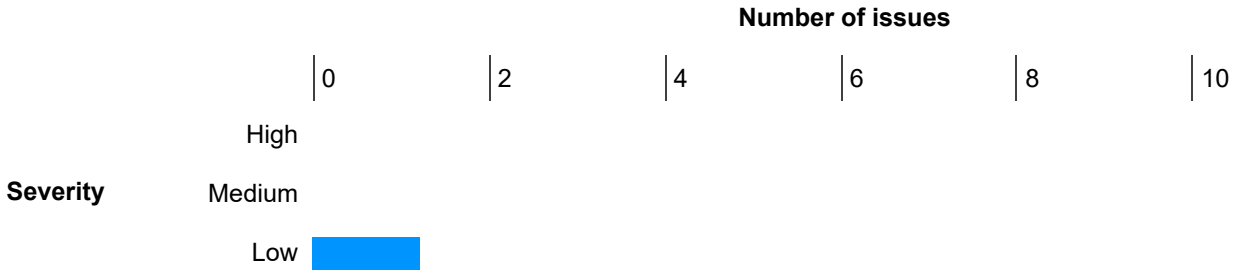
Burp Suite Professional

Summary

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

		Confidence			
		Certain	Firm	Tentative	Total
Severity	High	0	0	0	0
	Medium	0	0	0	0
	Low	1	0	0	1
	Information	9	2	0	11
	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. Unencrypted communications

Summary

Severity:	**Low**
Confidence:	**Certain**
Host:	**http://localhost:3000**
Path:	**/**

Issue description

The application allows users to connect to it over unencrypted connections. An attacker suitably positioned to view a legitimate user's network traffic could record and monitor their interactions with the application and obtain any information the user supplies. Furthermore, an attacker able to modify traffic could use the application as a platform for attacks against its users and third-party websites. Unencrypted connections have been exploited by ISPs and governments to track users, and to inject adverts and malicious JavaScript. Due to these concerns, web browser vendors are planning to visually flag unencrypted connections as hazardous.

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.

Please note that using a mixture of encrypted and unencrypted communications is an ineffective defense against active attackers, because they can easily remove references to encrypted resources when these references are transmitted over an unencrypted connection.

Issue remediation

Applications should use transport-level encryption (SSL/TLS) to protect all communications passing between the client and the server. The Strict-Transport-Security HTTP header should be used to ensure that clients refuse to access the server over an insecure connection.

References

- Marking HTTP as non-secure
- Configuring Server-Side SSL/TLS
- HTTP Strict Transport Security

Vulnerability classifications

- CWE-326: Inadequate Encryption Strength
- CAPEC-94: Man in the Middle Attack
- CAPEC-157: Sniffing Attacks

---

2. Cross-origin resource sharing

There are 4 instances of this issue:

- /
- /manifest.json
- /robots.txt
- /static/js/bundle.js

Issue background

An HTML5 cross-origin resource sharing (CORS) policy controls whether and how content running on other domains can perform two-way interaction with the domain that publishes the policy. The policy is fine-grained and can apply access controls per-request based on the URL and other features of the request.

If another domain is allowed by the policy, then that domain can potentially attack users of the application. If a user is logged in to the application, and visits a domain allowed by the policy, then any malicious content running on that domain can potentially retrieve content from the application, and sometimes carry out actions within the security context of the logged in user.

Even if an allowed domain is not overtly malicious in itself, security vulnerabilities within that domain could potentially be leveraged by an attacker to exploit the trust relationship and attack the application that allows access. CORS policies on pages containing sensitive information should be reviewed to determine whether it is appropriate for the application to trust both the intentions and security posture of any domains granted access.

Issue remediation

Any inappropriate domains should be removed from the CORS policy.

References

- Web Security Academy: Cross-origin resource sharing (CORS)
- Exploiting CORS Misconfigurations

Vulnerability classifications

- CWE-942: Overly Permissive Cross-domain Whitelist

2.1. http://localhost:3000/

Summary

Severity:	**Information**
Confidence:	**Certain**
Host:	**http://localhost:3000**
Path:	**/**

Issue detail

The application implements an HTML5 cross-origin resource sharing (CORS) policy for this request.

If the application relies on network firewalls or other IP-based access controls, this policy is likely to present a security risk.

Since the Vary: Origin header was not present in the response, reverse proxies and intermediate servers may cache it. This may enable an attacker to carry out cache poisoning attacks.

Request

```
GET / HTTP/1.1
Host: localhost:3000
Accept-Encoding: gzip, deflate, br
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
Connection: close
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Sec-CH-UA: ".Not/A)Brand";v="99", "Google Chrome";v="130", "Chromium";v="130"
Sec-CH-UA-Platform: Windows
Sec-CH-UA-Mobile: ?0
Origin: http://localhost:3000
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: *
Access-Control-Allow-Headers: *
Content-Type: text/html; charset=utf-8
Accept-Ranges: bytes
ETag: W/"6af-+M4OSPFNZpwKBdFEydrj+1+V5xo"
Vary: Accept-Encoding
Date: Wed, 20 Nov 2024 11:36:22 GMT
Connection: close
Content-Length: 1711

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<link rel="icon" href="/favicon.ico" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<meta
...[SNIP]...
```

2.2. http://localhost:3000/manifest.json

Summary

Severity:	**Information**
Confidence:	**Certain**
Host:	**http://localhost:3000**
Path:	**/manifest.json**

Issue detail

The application implements an HTML5 cross-origin resource sharing (CORS) policy for this request.

If the application relies on network firewalls or other IP-based access controls, this policy is likely to present a security risk.

Since the Vary: Origin header was not present in the response, reverse proxies and intermediate servers may cache it. This may enable an attacker to carry out cache poisoning attacks.

Request

```
GET /manifest.json HTTP/1.1
Host: localhost:3000
Accept-Encoding: gzip, deflate, br
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
Connection: close
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Sec-CH-UA: ".Not/A)Brand";v="99", "Google Chrome";v="130", "Chromium";v="130"
Sec-CH-UA-Platform: Windows
Sec-CH-UA-Mobile: ?0
Content-Length: 0
Origin: http://localhost:3000
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: *
Access-Control-Allow-Headers: *
Accept-Ranges: bytes
Cache-Control: public, max-age=0
Last-Modified: Mon, 18 Nov 2024 09:47:08 GMT
ETag: W/"1ec-1933eab3704"
Content-Type: application/json; charset=UTF-8
Content-Length: 492
Vary: Accept-Encoding
Date: Wed, 20 Nov 2024 11:36:22 GMT
Connection: close

{
"short_name": "React App",
"name": "Create React App Sample",
"icons": [
{
"src": "favicon.ico",
"sizes": "64x64 32x32 24x24 16x16",
"type": "image/x-icon"
},
{

...[SNIP]...
```

2.3. http://localhost:3000/robots.txt

Summary

Severity:	**Information**
Confidence:	**Certain**

Host: **http://localhost:3000**

Path: **/robots.txt**

Issue detail

The application implements an HTML5 cross-origin resource sharing (CORS) policy for this request.

If the application relies on network firewalls or other IP-based access controls, this policy is likely to present a security risk.

Since the Vary: Origin header was not present in the response, reverse proxies and intermediate servers may cache it. This may enable an attacker to carry out cache poisoning attacks.

Request

```
GET /robots.txt HTTP/1.1
Host: localhost:3000
Accept-Encoding: gzip, deflate, br
Accept: */*
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70
Safari/537.36
Connection: close
Cache-Control: max-age=0
Origin: http://localhost:3000
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: *
Access-Control-Allow-Headers: *
Accept-Ranges: bytes
Cache-Control: public, max-age=0
Last-Modified: Mon, 18 Nov 2024 09:47:08 GMT
ETag: W/"43-1933eab3710"
Content-Type: text/plain; charset=UTF-8
Content-Length: 67
Vary: Accept-Encoding
Date: Wed, 20 Nov 2024 11:36:21 GMT
Connection: close

# https://www.robotstxt.org/robotstxt.html
User-agent: *
Disallow:
```

2.4. http://localhost:3000/static/js/bundle.js

Summary

Severity: **Information**

Confidence: **Certain**

Host: **http://localhost:3000**

Path: **/static/js/bundle.js**

Issue detail

The application implements an HTML5 cross-origin resource sharing (CORS) policy for this request.

If the application relies on network firewalls or other IP-based access controls, this policy is likely to present a security risk.

Since the Vary: Origin header was not present in the response, reverse proxies and intermediate servers may cache it. This may enable an attacker to carry out cache poisoning attacks.

Request

```
GET /static/js/bundle.js HTTP/1.1
Host: localhost:3000
Accept-Encoding: gzip, deflate, br
Accept: */*
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70
Safari/537.36
Connection: close
Cache-Control: max-age=0
Referer: http://localhost:3000/
Sec-CH-UA: ".Not/A)Brand";v="99", "Google Chrome";v="130", "Chromium";v="130"
Sec-CH-UA-Platform: Windows
Sec-CH-UA-Mobile: ?0
Origin: http://localhost:3000
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: *
Access-Control-Allow-Headers: *
Content-Type: application/javascript; charset=utf-8
Accept-Ranges: bytes
ETag: W/"46035c-tot+eCqkM0N5iSYRAexJ8p61Mzc"
Vary: Accept-Encoding
Date: Wed, 20 Nov 2024 11:36:25 GMT
Connection: close
Content-Length: 4588380

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

/***/ "./src/App.js":
/*!********************!*\
!*** ./src/App.js ***!
\********************/
/***/ ((module, __webpa
...[SNIP]...
```

---

3. Cross-origin resource sharing: arbitrary origin trusted

There are 4 instances of this issue:

- /
- /manifest.json
- /robots.txt
- /static/js/bundle.js

Issue background

An HTML5 cross-origin resource sharing (CORS) policy controls whether and how content running on other domains can perform two-way interaction with the domain that publishes the policy. The policy is fine-grained and can apply access controls per-request based on the URL and other features of the request.

Trusting arbitrary origins effectively disables the same-origin policy, allowing two-way interaction by third-party web sites. Unless the response consists only of unprotected public content, this policy is likely to present a security risk.

If the site specifies the header Access-Control-Allow-Credentials: true, third-party sites may be able to carry out privileged actions and retrieve sensitive information. Even if it does not, attackers may be able to bypass any IP-based access controls by proxying through users' browsers.

Issue remediation

Rather than using a wildcard or programmatically verifying supplied origins, use a whitelist of trusted domains.

References

- Web Security Academy: Cross-origin resource sharing (CORS)
- Exploiting CORS Misconfigurations

Vulnerability classifications

- CWE-942: Overly Permissive Cross-domain Whitelist

3.1. http://localhost:3000/

Summary

Severity:	**Information**
Confidence:	**Certain**
Host:	**http://localhost:3000**
Path:	**/**

Issue detail

The application implements an HTML5 cross-origin resource sharing (CORS) policy for this request that allows access from any domain.

The application allowed access from the requested origin **http://uiwlwihtuikj.com**

If the application relies on network firewalls or other IP-based access controls, this policy is likely to present a security risk.

Since the Vary: Origin header was not present in the response, reverse proxies and intermediate servers may cache it. This may enable an attacker to carry out cache poisoning attacks.

Request

```
GET / HTTP/1.1
Host: localhost:3000
Accept-Encoding: gzip, deflate, br
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
Connection: close
```

```
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Sec-CH-UA: ".Not/A)Brand";v="99", "Google Chrome";v="130", "Chromium";v="130"
Sec-CH-UA-Platform: Windows
Sec-CH-UA-Mobile: ?0
Origin: http://uiwlwihtuikj.com
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: *
Access-Control-Allow-Headers: *
Content-Type: text/html; charset=utf-8
Accept-Ranges: bytes
ETag: W/"6af-+M4OSPFNZpwKBdFEydrj+1+V5xo"
Vary: Accept-Encoding
Date: Wed, 20 Nov 2024 11:36:22 GMT
Connection: close
Content-Length: 1711

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<link rel="icon" href="/favicon.ico" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<meta
...[SNIP]...
```

3.2. http://localhost:3000/manifest.json

Summary

Severity:	**Information**
Confidence:	**Certain**
Host:	**http://localhost:3000**
Path:	**/manifest.json**

Issue detail

The application implements an HTML5 cross-origin resource sharing (CORS) policy for this request that allows access from any domain.

The application allowed access from the requested origin **http://ciwbebqiqmqd.com**

If the application relies on network firewalls or other IP-based access controls, this policy is likely to present a security risk.

Since the Vary: Origin header was not present in the response, reverse proxies and intermediate servers may cache it. This may enable an attacker to carry out cache poisoning attacks.

Request

```
GET /manifest.json HTTP/1.1
Host: localhost:3000
Accept-Encoding: gzip, deflate, br
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-
```

```
exchange;v=b3;q=0.7
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70
Safari/537.36
Connection: close
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Sec-CH-UA: ".Not/A)Brand";v="99", "Google Chrome";v="130", "Chromium";v="130"
Sec-CH-UA-Platform: Windows
Sec-CH-UA-Mobile: ?0
Content-Length: 0
Origin: http://ciwbebqiqmqd.com
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: *
Access-Control-Allow-Headers: *
Accept-Ranges: bytes
Cache-Control: public, max-age=0
Last-Modified: Mon, 18 Nov 2024 09:47:08 GMT
ETag: W/"1ec-1933eab3704"
Content-Type: application/json; charset=UTF-8
Content-Length: 492
Vary: Accept-Encoding
Date: Wed, 20 Nov 2024 11:36:23 GMT
Connection: close

{
"short_name": "React App",
"name": "Create React App Sample",
"icons": [
{
"src": "favicon.ico",
"sizes": "64x64 32x32 24x24 16x16",
"type": "image/x-icon"
},
{

...[SNIP]...
```

3.3. http://localhost:3000/robots.txt

Summary

Severity:	**Information**
Confidence:	**Certain**
Host:	**http://localhost:3000**
Path:	**/robots.txt**

Issue detail

The application implements an HTML5 cross-origin resource sharing (CORS) policy for this request that allows access from any domain.

The application allowed access from the requested origin **http://fanwxlzejucl.com**

If the application relies on network firewalls or other IP-based access controls, this policy is likely to present a security risk.

Since the Vary: Origin header was not present in the response, reverse proxies and intermediate servers may cache it. This may enable an attacker to carry out cache poisoning attacks.

Request

```
GET /robots.txt HTTP/1.1
Host: localhost:3000
Accept-Encoding: gzip, deflate, br
Accept: */*
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70
Safari/537.36
Connection: close
Cache-Control: max-age=0
Origin: http://fanwxlzejucl.com
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: *
Access-Control-Allow-Headers: *
Accept-Ranges: bytes
Cache-Control: public, max-age=0
Last-Modified: Mon, 18 Nov 2024 09:47:08 GMT
ETag: W/"43-1933eab3710"
Content-Type: text/plain; charset=UTF-8
Content-Length: 67
Vary: Accept-Encoding
Date: Wed, 20 Nov 2024 11:36:21 GMT
Connection: close

# https://www.robotstxt.org/robotstxt.html
User-agent: *
Disallow:
```

3.4. http://localhost:3000/static/js/bundle.js

Summary

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

Issue detail

The application implements an HTML5 cross-origin resource sharing (CORS) policy for this request that allows access from any domain.

The application allowed access from the requested origin **http://qxvtdagimkpl.com**

If the application relies on network firewalls or other IP-based access controls, this policy is likely to present a security risk.

Since the Vary: Origin header was not present in the response, reverse proxies and intermediate servers may cache it. This may enable an attacker to carry out cache poisoning attacks.

Request

```
GET /static/js/bundle.js HTTP/1.1
Host: localhost:3000
Accept-Encoding: gzip, deflate, br
Accept: */*
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70
Safari/537.36
Connection: close
Cache-Control: max-age=0
Referer: http://localhost:3000/
Sec-CH-UA: ".Not/A)Brand";v="99", "Google Chrome";v="130", "Chromium";v="130"
Sec-CH-UA-Platform: Windows
Sec-CH-UA-Mobile: ?0
Origin: http://qxvtdagimkpl.com
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: *
Access-Control-Allow-Headers: *
Content-Type: application/javascript; charset=utf-8
Accept-Ranges: bytes
ETag: W/"46035c-tot+eCqkM0N5iSYRAexJ8p61Mzc"
Vary: Accept-Encoding
Date: Wed, 20 Nov 2024 11:36:26 GMT
Connection: close
Content-Length: 4588380

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

/***/ "./src/App.js":
/*!********************!*\
!*** ./src/App.js ***!
\********************/
/***/ ((module, __webpa
...[SNIP]...
```

---

4. Frameable response (potential Clickjacking)

There are 2 instances of this issue:

- /
- /courses

Issue description

If a page fails to set an appropriate X-Frame-Options or Content-Security-Policy HTTP header, it might be possible for a page controlled by an attacker to load it within an iframe. This may enable a clickjacking attack, in which the attacker's page overlays the target application's interface with a different interface provided by the attacker. By inducing victim users to perform actions such as mouse clicks and keystrokes, the attacker can cause them to unwittingly carry out actions within the application that is being targeted. This technique allows the attacker to circumvent defenses against cross-site request forgery, and may result in unauthorized actions.

Note that some applications attempt to prevent these attacks from within the HTML page itself, using "framebusting" code. However, this type of defense is normally ineffective and can usually be circumvented by a skilled attacker.

You should determine whether any functions accessible within frameable pages can be used by application users to perform any sensitive actions within the application.

Issue remediation

To effectively prevent framing attacks, the application should return a response header with the name **X-Frame-Options** and the value **DENY** to prevent framing altogether, or the value **SAMEORIGIN** to allow framing only by pages on the same origin as the response itself. Note that the SAMEORIGIN header can be partially bypassed if the application itself can be made to frame untrusted websites.

References

- Web Security Academy: Clickjacking
- X-Frame-Options

Vulnerability classifications

- CWE-693: Protection Mechanism Failure
- CWE-1021: Improper Restriction of Rendered UI Layers or Frames
- CAPEC-103: Clickjacking

4.1. http://localhost:3000/

Summary

Severity:	**Information**
Confidence:	**Firm**
Host:	**http://localhost:3000**
Path:	**/**

Request

```
GET / HTTP/1.1
Host: localhost:3000
Accept-Encoding: gzip, deflate, br
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
Connection: close
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Sec-CH-UA: ".Not/A)Brand";v="99", "Google Chrome";v="130", "Chromium";v="130"
Sec-CH-UA-Platform: Windows
Sec-CH-UA-Mobile: ?0
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: *
```

```
Access-Control-Allow-Headers: *
Content-Type: text/html; charset=utf-8
Accept-Ranges: bytes
ETag: W/"6af-+M4OSPFNZpwKBdFEydrj+1+V5xo"
Vary: Accept-Encoding
Date: Wed, 20 Nov 2024 11:36:10 GMT
Connection: close
Content-Length: 1711

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<link rel="icon" href="/favicon.ico" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<meta
...[SNIP]...
```

4.2. http://localhost:3000/courses

Summary

Severity:	**Information**
Confidence:	**Firm**
Host:	**http://localhost:3000**
Path:	**/courses**

Request

```
GET /courses HTTP/1.1
Host: localhost:3000
sec-ch-ua: "Not?A_Brand";v="99", "Chromium";v="130"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70
Safari/537.36
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
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: *
Access-Control-Allow-Headers: *
Content-Type: text/html; charset=utf-8
Accept-Ranges: bytes
ETag: W/"6af-+M4OSPFNZpwKBdFEydrj+1+V5xo"
Vary: Accept-Encoding
```

```
Date: Wed, 20 Nov 2024 10:53:52 GMT
Connection: keep-alive
Keep-Alive: timeout=5
Content-Length: 1711

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<link rel="icon" href="/favicon.ico" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<meta
...[SNIP]...
```

---

5. Robots.txt file

Summary

Severity:	**Information**
Confidence:	**Certain**
Host:	**http://localhost:3000**
Path:	**/robots.txt**

Issue detail

The web server contains a robots.txt file.

Issue background

The file robots.txt is used to give instructions to web robots, such as search engine crawlers, about locations within the web site that robots are allowed, or not allowed, to crawl and index.

The presence of the robots.txt does not in itself present any kind of security vulnerability. However, it is often used to identify restricted or private areas of a site's contents. The information in the file may therefore help an attacker to map out the site's contents, especially if some of the locations identified are not linked from elsewhere in the site. If the application relies on robots.txt to protect access to these areas, and does not enforce proper access control over them, then this presents a serious vulnerability.

Issue remediation

The robots.txt file is not itself a security threat, and its correct use can represent good practice for non-security reasons. You should not assume that all web robots will honor the file's instructions. Rather, assume that attackers will pay close attention to any locations identified in the file. Do not rely on robots.txt to provide any kind of protection over unauthorized access.

Vulnerability classifications

- CWE-200: Information Exposure

Request

```
GET /robots.txt HTTP/1.1
Host: localhost:3000
Accept-Encoding: gzip, deflate, br
Accept: */*
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70
Safari/537.36
Connection: close
```

```
Cache-Control: max-age=0
```

Response

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: *
Access-Control-Allow-Headers: *
Accept-Ranges: bytes
Cache-Control: public, max-age=0
Last-Modified: Mon, 18 Nov 2024 09:47:08 GMT
ETag: W/"43-1933eab3710"
Content-Type: text/plain; charset=UTF-8
Content-Length: 67
Vary: Accept-Encoding
Date: Wed, 20 Nov 2024 11:36:20 GMT
Connection: close

# https://www.robotstxt.org/robotstxt.html
User-agent: *
Disallow:
```
