Relatório de Testes de Segurança - auth/login

Este relatório apresenta os resultados da análise de segurança realizada na API desenvolvida em NestJS. O teste foi realizado utilizando a ferramenta OWASP ZAP, aplicando varreduras ativas nos endpoints /auth/login e /users. O objetivo foi identificar vulnerabilidades como SQL Injection, XSS, problemas de autenticação, autorização e configuração.

Sites: http://example.com. http://localhost:3000

Generated on ter., 27 mai. 2025 01:02:21

ZAP Version: 2.16.1

ZAP by Checkmarx

Summary of Alerts

Nível de Risco	Number of Alerts
Alto	1
Médio	2
Baixo	2
Informativo	2

Alertas

Nome	Nível de Risco	Number of Instances
Injeção SQL	Alto	1
Content Security Policy (CSP) Header Not Set	Médio	1
Missing Anti-clickjacking Header	Médio	1
O servidor vaza informações por meio dos campos de cabeçalho de resposta HTTP "X-Powered-By"	Baixo	2
X-Content-Type-Options Header Missing	Baixo	3
Authentication Request Identified	Informativo	2
Session Management Response Identified	Informativo	1

Alert Detail

Alto	Injeção SQL
Descrição	SQL injection may be possible.
URL	http://localhost:3000/auth/login
Método	POST
Ataque	localhost:3000 OR 1=1 --

Evidence	
Other Info	The page results were successfully manipulated using the boolean conditions [localhost: 3000 AND 1=1 --] and [localhost:3000 OR 1=1 --] The parameter value being modified was stripped from the HTML output for the purposes of the comparison. Data was NOT returned for the original parameter. The vulnerability was detected by successfully retrieving more data than originally returned, by manipulating the parameter.
Instances	1
Solution	Do not trust client side input, even if there is client side validation in place. In general, type check all data on the server side. If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?' If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries. If database Stored Procedures can be used, use them. Do *not* concatenate strings into queries in the stored procedure, or use 'exec', 'exec immediate', or equivalent functionality! Do not create dynamic SQL queries using simple string concatenation. Escape all data received from the client. Apply an 'allow list' of allowed characters, or a 'deny list' of disallowed characters in user input. Apply the principle of least privilege by using the least privileged database user possible. In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact. Grant the minimum database access that is necessary for the application.
Reference	https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html
CWE Id	89
WASC Id	19
Plugin Id	40018

Médio	**Content Security Policy (CSP) Header Not Set**
Descrição	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
URL	http://example.com./
Método	GET
Ataque	
Evidence	
Other Info	
Instances	1
Solution	Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Reference	https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html https://www.w3.org/TR/CSP/ https://w3c.github.io/webappsec-csp/ https://web.dev/articles/csp https://caniuse.com/#feat=contentsecuritypolicy https://content-security-policy.com/
CWE Id	693
WASC Id	15
Plugin Id	10038

Médio	**Missing Anti-clickjacking Header**
Descrição	The response does not protect against 'ClickJacking' attacks. It should include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options.
URL	http://example.com./
Método	GET
Ataque	
Evidence	
Other Info	
Instances	1
Solution	Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app. If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.
Reference	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
CWE Id	1021
WASC Id	15
Plugin Id	10020

Baixo	**O servidor vaza informações por meio dos campos de cabeçalho de resposta HTTP "X-Powered-By"**
Descrição	O servidor da web/aplicativo está vazando informações por meio de um ou mais cabeçalhos de resposta HTTP "X-Powered-By". O acesso a essas informações pode facilitar que os invasores identifiquem outras estruturas/componentes dos quais seu aplicativo da web depende e as vulnerabilidades às quais esses componentes podem estar sujeitos.
URL	http://localhost:3000/users
Método	GET
Ataque	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/auth/login
Método	POST
Ataque	
Evidence	X-Powered-By: Express

Other Info	
Instances	2
Solution	Certifique-se de que seu servidor web, servidor de aplicativos, balanceador de carga, etc. esteja configurado para suprimir cabeçalhos "X-Powered-By".
Reference	https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/01-Information_Gathering/08-Fingerprint_Web_Application_Framework https://www.troyhunt.com/2012/02/shhh-dont-let-your-response-headers.html
CWE Id	497
WASC Id	13
Plugin Id	10037

Baixo	**X-Content-Type-Options Header Missing**
Descrição	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.
URL	http://example.com./
Método	GET
Ataque	
Evidence	
Other Info	This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scan rule will not alert on client or server error responses.
URL	http://localhost:3000/users
Método	GET
Ataque	
Evidence	
Other Info	This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scan rule will not alert on client or server error responses.
URL	http://localhost:3000/auth/login
Método	POST
Ataque	
Evidence	
Other Info	This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scan rule will not alert on client or server error responses.
Instances	3
Solution	Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages. If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application /web server to not perform MIME-sniffing.
	https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer

Reference	/compatibility/gg622941(v=vs.85) https://owasp.org/www-community/Security_Headers
CWE Id	693
WASC Id	15
Plugin Id	10021

Informativo	Authentication Request Identified
Descrição	The given request has been identified as an authentication request. The 'Other Info' field contains a set of key=value lines which identify any relevant fields. If the request is in a context which has an Authentication Method set to "Auto-Detect" then this rule will change the authentication to match the request identified.
URL	http://localhost:3000/auth/login
Método	POST
Ataque	
Evidence	password
Other Info	userParam=email userValue=lucas@email.com passwordParam=password
URL	http://localhost:3000/auth/login
Método	POST
Ataque	
Evidence	password
Other Info	userParam=email userValue=teste@email.com passwordParam=password
Instances	2
Solution	This is an informational alert rather than a vulnerability and so there is nothing to fix.
Reference	https://www.zaproxy.org/docs/desktop/addons/authentication-helper/auth-req-id/
CWE Id	
WASC Id	
Plugin Id	10111

Informativo	Session Management Response Identified
Descrição	The given response has been identified as containing a session management token. The 'Other a set of header tokens that can be used in the Header Based Session Management Method. If t context which has a Session Management Method set to "Auto-Detect" then this rule will chang management to use the tokens identified.
URL	http://localhost:3000/auth/login
Método	GET
Ataque	
Evidence	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9. eyJzdWIiOjMsImVtYWlsIjoibHVjYXNAZW1haWwuY29tIiwiaWF0IjoxNzQ4MzEzNjE5LCJleHAiO JT5SO7OMa4JzWtVeJ805IyuExEg60dMTNNlkRJK5kNI
Other Info	json:access_token
Instances	1
Solution	This is an informational alert rather than a vulnerability and so there is nothing to fix.
Reference	https://www.zaproxy.org/docs/desktop/addons/authentication-helper/session-mgmt-id
CWE Id	
WASC Id	

Plugin Id	10112