ZAP Scanning Report

Sites: http://localhost:8084 http://localhost:8083 http://localhost:8082 http://localhost:8081 http://localhost:8080

Generated on Sun, 7 Jul 2024 14:11:01

ZAP Version: 2.15.0

ZAP is supported by the Crash Override Open Source Fellowship

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	1
Low	3
Informational	1

Alerts

Name	Risk Level	Number of Instances
Cross-Domain Misconfiguration	Medium	10
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	10
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	10
X-Content-Type-Options Header Missing	Low	10
Authentication Request Identified	Informational	2

Alert Detail

Medium	Cross-Domain Misconfiguration
Description	Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server
URL	http://localhost:8081/api/products/search?id%5B%5D=1&id%5B%5D=2
Method	GET
Attack	
Evidence	Access-Control-Allow-Origin: *
Other Info	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.
URL	http://localhost:8082/api/orders

Method	GET
Attack	
Evidence	Access-Control-Allow-Origin: *
Other Info	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.
URL	http://localhost:8084/api/orders
Method	GET
Attack	
Evidence	Access-Control-Allow-Origin: *
Other Info	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.
URL	http://localhost:8080/api/login
Method	POST
Attack	
Evidence	Access-Control-Allow-Origin: *
Other Info	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.
URL	http://localhost:8080/api/users/create
Method	POST
Attack	
Evidence	Access-Control-Allow-Origin: *
Other Info	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.
URL	http://localhost:8081/api/categories
Method	POST
Attack	
Evidence	Access-Control-Allow-Origin: *
Other Info	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.
URL	http://localhost:8081/api/products
Method	POST

Attack	
Evidence	Access-Control-Allow-Origin: *
Other Info	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.
URL	http://localhost:8082/api/orders
Method	POST
Attack	
Evidence	Access-Control-Allow-Origin: *
Other Info	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.
URL	http://localhost:8083/api/webhook/ml/pix
Method	POST
Attack	
Evidence	Access-Control-Allow-Origin: *
Other Info	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.
URL	http://localhost:8084/api/orders/666caee86f2fe5907b03a832
Method	PUT
Attack	
Evidence	Access-Control-Allow-Origin: *
Other Info	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.
Instances	10
Solution	Ensure that sensitive data is not available in an unauthenticated manner (using IP address white-listing, for instance). Configure the "Access-Control-Allow-Origin" HTTP header to a more restrictive set of domains, or remove all CORS headers entirely, to allow the web browser to enforce the Same Origin Policy (SOP) in a more restrictive manner.
Reference	https://vulncat.fortify.com/en/detail?id=desc.config.dotnet.html5_overly_permissive_cors_policy
CWE Id	264
WASC Id	14
Plugin Id	10098

Low	**Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)**

Description	The web/application server is leaking information via one or more "X-Powered-By" HTTP response headers. Access to such information may facilitate attackers identifying other frameworks/components your web application is reliant upon and the vulnerabilities such components may be subject to.
URL	http://localhost:8081/api/products/search?id%5B%5D=1&id%5B%5D=2
Method	GET
Attack	
Evidence	X-Powered-By: PHP/8.2.20
Other Info	
URL	http://localhost:8082/api/orders
Method	GET
Attack	
Evidence	X-Powered-By: PHP/8.2.20
Other Info	
URL	http://localhost:8084/api/orders
Method	GET
Attack	
Evidence	X-Powered-By: PHP/8.2.20
Other Info	
URL	http://localhost:8080/api/login
Method	POST
Attack	
Evidence	X-Powered-By: PHP/8.2.20
Other Info	
URL	http://localhost:8080/api/users/create
Method	POST
Attack	
Evidence	X-Powered-By: PHP/8.2.20
Other Info	
URL	http://localhost:8081/api/categories
Method	POST
Attack	
Evidence	X-Powered-By: PHP/8.2.20
Other Info	
URL	http://localhost:8081/api/products
Method	POST
Attack	
Evidence	X-Powered-By: PHP/8.2.20

Other Info	
URL	http://localhost:8082/api/orders
Method	POST
Attack	
Evidence	X-Powered-By: PHP/8.2.20
Other Info	
URL	http://localhost:8083/api/webhook/ml/pix
Method	POST
Attack	
Evidence	X-Powered-By: PHP/8.2.20
Other Info	
URL	http://localhost:8084/api/orders/666caee86f2fe5907b03a832
Method	PUT
Attack	
Evidence	X-Powered-By: PHP/8.2.20
Other Info	
Instances	10
Solution	Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.
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	200
WASC Id	13
Plugin Id	10037

Low	**Server Leaks Version Information via "Server" HTTP Response Header Field**
Description	The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.
URL	http://localhost:8081/api/products/search?id%5B%5D=1&id%5B%5D=2
Method	GET
Attack	
Evidence	Apache/2.4.59 (Debian)
Other Info	
URL	http://localhost:8082/api/orders
Method	GET
Attack	
Evidence	Apache/2.4.59 (Debian)
Other Info	

URL	http://localhost:8084/api/orders
Method	GET
Attack	
Evidence	Apache/2.4.59 (Debian)
Other Info	
URL	http://localhost:8080/api/login
Method	POST
Attack	
Evidence	Apache/2.4.59 (Debian)
Other Info	
URL	http://localhost:8080/api/users/create
Method	POST
Attack	
Evidence	Apache/2.4.59 (Debian)
Other Info	
URL	http://localhost:8081/api/categories
Method	POST
Attack	
Evidence	Apache/2.4.59 (Debian)
Other Info	
URL	http://localhost:8081/api/products
Method	POST
Attack	
Evidence	Apache/2.4.59 (Debian)
Other Info	
URL	http://localhost:8082/api/orders
Method	POST
Attack	
Evidence	Apache/2.4.59 (Debian)
Other Info	
URL	http://localhost:8083/api/webhook/ml/pix
Method	POST
Attack	
Evidence	Apache/2.4.59 (Debian)
Other Info	
URL	http://localhost:8084/api/orders/666caee86f2fe5907b03a832
Method	PUT

Attack	
Evidence	Apache/2.4.59 (Debian)
Other Info	
Instances	10
Solution	Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.
Reference	https://httpd.apache.org/docs/current/mod/core.html#servertokens https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff648552(v=pandp.10) https://www.troyhunt.com/shhh-dont-let-your-response-headers/
CWE Id	200
WASC Id	13
Plugin Id	10036

Low	**X-Content-Type-Options Header Missing**
Description	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://localhost:8081/api/products/search?id%5B%5D=1&id%5B%5D=2
Method	GET
Attack	
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:8082/api/orders
Method	GET
Attack	
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:8084/api/orders
Method	GET
Attack	
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:8080/api/login
Method	POST
Attack	
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:8080/api/users/create
Method	POST
Attack	
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:8081/api/categories
Method	POST
Attack	
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:8081/api/products
Method	POST
Attack	
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:8082/api/orders
Method	POST
Attack	
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:8083/api/webhook/ml/pix
Method	POST
Attack	
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:8084/api/orders/666caee86f2fe5907b03a832
Method	PUT
Attack	
Evidence	
	This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still

Other Info	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	10
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.
Reference	https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/compatibility/gg622941(v=vs.85) https://owasp.org/www-community/Security_Headers
CWE Id	693
WASC Id	15
Plugin Id	10021

Informational	**Authentication Request Identified**
Description	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:8080/api/users/create
Method	POST
Attack	
Evidence	password
Other Info	userParam=email userValue=user2@example.com passwordParam=password
URL	http://localhost:8080/api/login
Method	POST
Attack	
Evidence	password
Other Info	userParam=email userValue=mario@mario.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