ZAP Scanning Report

Generated with ZAP on qui. 8 ago. 2024, at 10:47:37

ZAP Version: 2.15.0

ZAP is supported by the Crash Override Open Source Fellowship

Contents

About this report

Report parameters

Contexts

No contexts were selected, so all contexts were included by default.

Sites

The following sites were included:

- `https://i8z10k6wma.execute-api.us-east-1.amazonaws.com`

(If no sites were selected, all sites were included by default.)

An included site must also be within one of the included contexts for its data to be included in the report.

Risk levels

Included: `Alto`, `Médio`, `Baixo`, `Informativo`

Excluded: None

Confidence levels

Included: `User Confirmed`, `Alto`, `Médio`, `Baixo`

Excluded: `User Confirmed`, `Alto`, `Médio`, `Baixo`, `Falso Positivo`

Summaries

Alert counts by risk and confidence

This table shows the number of alerts for each level of risk and confidence included in the report.

(The percentages in brackets represent the count as a percentage of the total number of alerts included in the report, rounded to one decimal place.)

		Confidence				
		User Confirmed	Alto	Médio	Baixo	Total
Risk	Alto	0 (0,0%)	0 (0,0%)	0 (0,0%)	0 (0,0%)	0 (0,0%)
	Médio	0 (0,0%)	1 (10,0%)	2 (20,0%)	0 (0,0%)	3 (30,0%)
	Baixo	0 (0,0%)	2 (20,0%)	2 (20,0%)	1 (10,0%)	5 (50,0%)
	Informativo	0 (0,0%)	0 (0,0%)	1 (10,0%)	1 (10,0%)	2 (20,0%)
	Total	0 (0,0%)	3 (30,0%)	5 (50,0%)	2 (20,0%)	10 (100%)

Alert counts by site and risk

This table shows, for each site for which one or more alerts were raised, the number of alerts raised at each risk level.

Alerts with a confidence level of "False Positive" have been excluded from these counts.

(The numbers in brackets are the number of alerts raised for the site at or above that risk level.)

		Risk			
					Informativo
		Alto (= Alto)	Médio (>= Médio)	Baixo (>= Baixo)	(>= Informativo)
Site	https://i8z10k6wma.execute-api.us-east-1.amazonaws.com	0 (0)	3 (3)	5 (8)	2 (10)

Alert counts by alert type

This table shows the number of alerts of each alert type, together with the alert type's risk level.

(The percentages in brackets represent each count as a percentage, rounded to one decimal place, of the total number of alerts included in this report.)

Alert type	Risk	Count
Content Security Policy (CSP) Header Not Set	Médio	1 (10,0%)
Erro de Formato de String	Médio	1 (10,0%)
Missing Anti-clickjacking Header	Médio	1 (10,0%)
Fraqueza de script entre sites (persistente na resposta JSON)	Baixo	1 (10,0%)
O servidor vaza informações por meio dos campos de cabeçalho de resposta HTTP "X-Powered-By"	Baixo	9 (90,0%)
Server Leaks Version Information via "Server" HTTP Response Header Field	Baixo	1 (10,0%)
Strict-Transport-Security Header Not Set	Baixo	12
Total		10

Alert type	Risk	Count
		(120,0%)
X-Content-Type-Options Header Missing	Baixo	6 (60,0%)
Re-examine Cache-control Directives	Informativo	6 (60,0%)
User Agent Fuzzer	Informativo	42 (420,0%)
Total		10

Alerts

Risk=Médio, Confidence=Alto (1)

https://i8z10k6wma.execute-api.us-east-1.amazonaws.com (1)

Content Security Policy (CSP) Header Not Set (1)

▶ PUT https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/payment/toggle

Risk=Médio, Confidence=Médio (2)

https://i8z10k6wma.execute-api.us-east-1.amazonaws.com (2)

Erro de Formato de String (1)

▼ POST https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/categories

Alert tags	▪ OWASP_2017_A01

- CWE-134
- OWASP_2021_A03

Alert description

Um erro de Formato de String ocorre quando o dado enviado de uma string de entrada é avaliado como um comando pelo aplicativo.

Other info

Possível Erro de Formato de String. O script fechou a conexão devido a um erro de formato de string da Microsoft

Request

▼ Request line and header section (338 bytes)

```
POST https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/categories HTTP/1.1
host: i8z10k6wma.execute-api.us-east-1.amazonaws.com
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
pragma: no-cache
cache-control: no-cache
content-type: application/json
content-length: 283
```

▼ Request body (283 bytes)

```
{
    "name": "xpto5",
    "slug": "ZAP %1!s%2!s%3!s%4!s%5!s%6!s%7!s%8!s%9!s%10!s%11!s%12!s%13!s%14!s%15!s%16!s%17!s%18!s%19!s%20!s%21!n%22!n%23!n%24!n%25!n%26!n%27!n%28!n%29!n%30!n%31!n%32!n%33!n%34!n%35!n%36!n%37!n%38!n%39!n%40!n\n",
    "description": "
```

	{{$randomLoremSentence}}" }

Response	▼ Status line and header section (266 bytes)
	`HTTP/1.1 500 Internal Server Error`
	`Date: Thu, 08 Aug 2024 13:46:38 GMT`
	`Content-Type: application/json; charset=utf-8`
	`Content-Length: 52`
	`Connection: keep-alive`
	`apigw-requestid: cMTZ0jaKIAMEbEw=`
	`X-Powered-By: Express`
	`etag: W/"34-rlKccw1E+/fV8niQk4oFitDfPro"`
	▼ Response body (52 bytes)
	`{"statusCode":500,"message":"Internal server error"}`

Parameter	slug
Attack	ZAP %1!s%2!s%3!s%4!s%5!s%6!s%7!s%8!s%9!s%10!s%11!s%12!s%13!s%14!s%15!s%16!s%17!s%18!s%19!s%20!s%21!n%22!n%23!n%24!n%25!n%26!n%27!n%28!n%29!n%30!n%31!n%32!n%33!n%34!n%35!n%36!n%37!n%38!n%39!n%40!n
Solution	Reescreva o programa de plano de fundo utilizando o apagamento apropriado das bad character strings. Isso irá requerer uma recompilação do executável do plano de fundo.

Missing Anti-clickjacking Header (1)

▼ PUT https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/payment/toggle

Alert tags

- WSTG-v42-CLNT-09
- OWASP_2021_A05
- OWASP_2017_A06
- CWE-1021

Alert description

The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.

Request

▼ Request line and header section (288 bytes)

```
PUT https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/payment/toggle HTTP/1.1
host: i8z10k6wma.execute-api.us-east-1.amazonaws.com
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
pragma: no-cache
cache-control: no-cache
```

▼ Request body (0 bytes)

Response

▼ Status line and header section (214 bytes)

```
HTTP/1.1 200 OK
Date: Thu, 08 Aug 2024 13:43:20 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 33
Connection: keep-alive
server: Werkzeug/2.2.2 Python/3.10.14
```

	`apigw-requestid: cMS64jHoIAMEVRw=`
	▼ Response body (33 bytes)
	`Now the payments will be canceled`

Parameter	`x-frame-options`
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.

Risk=Baixo, Confidence=Alto (2)

https://i8z10k6wma.execute-api.us-east-1.amazonaws.com (2)

Server Leaks Version Information via "Server" HTTP Response Header Field (1)

▼ PUT https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/payment/toggle

Alert tags	OWASP_2021_A05 OWASP_2017_A06 WSTG-v42-INFO-02 CWE-200

Alert 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.

Request

▼ Request line and header section (288 bytes)

```
PUT https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/payment/toggle HTTP/1.1
host: i8z10k6wma.execute-api.us-east-1.amazonaws.com
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
pragma: no-cache
cache-control: no-cache
```

▼ Request body (0 bytes)

Response

▼ Status line and header section (214 bytes)

```
HTTP/1.1 200 OK
Date: Thu, 08 Aug 2024 13:43:20 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 33
Connection: keep-alive
server: Werkzeug/2.2.2 Python/3.10.14
apigw-requestid: cMS64jHoIAMEVRw=
```

▼ Response body (33 bytes)

```
Now the payments will be canceled
```

Evidence

`Werkzeug/2.2.2 Python/3.10.14`

Solution

Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Strict-Transport-Security Header Not Set (1)

▼ `POST https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/auth`

Alert tags

- OWASP_2021_A05
- OWASP_2017_A06
- CWE-319

Alert description

HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an IETF standards track protocol and is specified in RFC 6797.

Request

▼ Request line and header section (331 bytes)

```
POST https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/auth HTTP/1.1
host: i8z10k6wma.execute-api.us-east-1.amazonaws.com
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
pragma: no-cache
cache-control: no-cache
content-type: application/json
```

```
content-length: 28
```

▼ Request body (28 bytes)

```
{
    "cpf": "12345678913"
}
```

Response

▼ Status line and header section (177 bytes)

```
HTTP/1.1 200 OK
Date: Thu, 08 Aug 2024 13:43:15 GMT
Content-Type: text/plain;
charset=utf-8
Content-Length: 485
Connection: keep-alive
Apigw-Requestid: cMS5vgiUIAMEVVA=
```

▼ Response body (485 bytes)

```
{
  "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
.eyJkYXRhIjp7Il9pZCI6IjY2YjRjYTA4NTY3
M2Q0MmMzNjQxMjY1YSIsIm5hbWUiOiJNb25hI
EJydWVuIiwiZW1haWwiOiJMYXVyZWw1MUBnbW
FpbC5jb20iLCJjcGYiOiIxMjM0NTY3ODkxMyI
sInBob25lIjoiNjAxLTI4OS05NjI5IiwiYWRk
cmVzcyI6IjIzNCBadWxhdWYgUmlkZ2VzIiwiY
3JlYXRlZEF0IjoiMjAyNC0wOC0wOFQxMzozNz
oxMi43MjZaIiwidXBkYXRlZEF0IjoiMjAyNC0
wOC0wOFQxMzozNzoxMi43MjZaIiwiX192Ijow
fSwiaWF0IjoxNzIzMTI0NTk1LCJleHAiOjE3M
jMxMjgxOTV9.OkmooDT879OiEmuk3lKsWAXHj
FCt9NRx5RbKdpcNojI"
}
```

Solution

Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.

Risk=Baixo, Confidence=Médio (2)

https://i8z10k6wma.execute-api.us-east-1.amazonaws.com (2)

O servidor vaza informações por meio dos campos de cabeçalho de resposta HTTP "X-Powered-By" (1)

▼ POST https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/customers

Alert tags	- OWASP_2021_A01 - OWASP_2017_A03 - WSTG-v42-INFO-08 - CWE-200
Alert description	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.

Request	▼ Request line and header section (337 bytes)
	`POST https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/customers HTTP/1.1`
	`host: i8z10k6wma.execute-api.us-east-1.amazonaws.com`
	`user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0`
	`pragma: no-cache`
	`cache-control: no-cache`

```
content-type: application/json
content-length: 178
```

▼ Request body (178 bytes)

```
{
    "name": "{{$randomFullName}}",
    "email": "{{$randomEmail}}",
    "phone": "
{{$randomPhoneNumber}}",
    "address": "
{{$randomStreetAddress}}",
    "cpf": "12345678913"
}
```

Response

▼ Status line and header section (256 bytes)

```
HTTP/1.1 400 Bad Request
Date: Thu, 08 Aug 2024 13:43:19 GMT
Content-Type: application/json;
charset=utf-8
Content-Length: 77
Connection: keep-alive
apigw-requestid: cMS6sgJNIAMEbqQ=
X-Powered-By: Express
etag: W/"4d-
ST/RBgnauwvfnfsKLs387lWfM+4"
```

▼ Response body (77 bytes)

```
{"message":["Email must be an
email"],"error":"Bad
Request","statusCode":400}
```

Evidence

`X-Powered-By: Express`

Solution

Certifique-se de que seu servidor web, servidor de aplicativos, balanceador de carga, etc. esteja configurado para suprimir cabeçalhos "X-Powered-By".

X-Content-Type-Options Header Missing (1)

▼ POST https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/auth

Alert tags	▪ CWE-693 ▪ OWASP_2021_A05 ▪ OWASP_2017_A06
Alert 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.
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.

Request	▼ Request line and header section (331 bytes)
	
	`POST https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/auth HTTP/1.1`
	`host: i8z10k6wma.execute-api.us-east-1.amazonaws.com`
	`user-agent: Mozilla/5.0 (Windows NT`

```
10.0; Win64; x64; rv:125.0)
Gecko/20100101 Firefox/125.0
pragma: no-cache
cache-control: no-cache
content-type: application/json
content-length: 28
```

▼ Request body (28 bytes)

```
{
    "cpf": "12345678913"
}
```

Response

▼ Status line and header section (177 bytes)

```
HTTP/1.1 200 OK
Date: Thu, 08 Aug 2024 13:43:15 GMT
Content-Type: text/plain;
charset=utf-8
Content-Length: 485
Connection: keep-alive
Apigw-Requestid: cMS5vgiUIAMEVVA=
```

▼ Response body (485 bytes)

```
{
  "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
.eyJkYXRhIjp7Il9pZCI6IjY2YjRjYTA4NTY3
M2Q0MmMzNjQxMjY1YSIsIm5hbWUiOiJNb25hI
EJydWVuIiwiZW1haWwiOiJMYXVyZWw1MUBnbW
FpbC5jb20iLCJjcGYiOiIxMjM0NTY3ODkxMyI
sInBob25lIjoiNjAxLTI4OS05NjI5IiwiYWRk
cmVzcyI6IjIzNCBadWxhdWYgUmlkZ2VzIiwiY
3JlYXRlZEF0IjoiMjAyNC0wOC0wOFQxMzozNz
oxMi43MjZaIiwidXBkYXRlZEF0IjoiMjAyNC0
wOC0wOFQxMzozNzoxMi43MjZaIiwiX192Ijow
fSwiaWF0IjoxNzIzMTI0NTk1LCJleHAiOjE3M
jMxMjgxOTV9.OkmooDT8790iEmuk3lKsWAXHj
```

	`FCt9NRx5RbKdpcNojI"`
	`}`

Parameter	`x-content-type-options`
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.

Risk=Baixo, Confidence=Baixo (1)

https://i8z10k6wma.execute-api.us-east-1.amazonaws.com (1)

Fraqueza de script entre sites (persistente na resposta JSON) (1)

▼ GET https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/categories

Alert tags	▪ CWE-79 ▪ WSTG-v42-INPV-02 ▪ OWASP_2021_A03 ▪ OWASP_2017_A07
Alert description	Um ataque XSS foi encontrado em uma resposta JSON, isso pode deixar os consumidores de conteúdo vulneráveis a ataques se eles não manipularem os dados (resposta) de forma adequada.

Other info	Gerado com BAIXA confiança, pois o Tipo de conteúdo não é HTML

Request	▼ Request line and header section (284 bytes)
	`GET https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/categories HTTP/1.1`
	`host: i8z10k6wma.execute-api.us-east-1.amazonaws.com`
	`user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0`
	`pragma: no-cache`
	`cache-control: no-cache`
	▼ Request body (0 bytes)

Response	▼ Status line and header section (252 bytes)
	`HTTP/1.1 200 OK`
	`Date: Thu, 08 Aug 2024 13:44:39 GMT`
	`Content-Type: application/json; charset=utf-8`
	`Content-Length: 10894`
	`Connection: keep-alive`
	`apigw-requestid: cMTHOioOoAMEVSQ=`
	`X-Powered-By: Express`
	`etag: W/"2a8e-OhCkmiD0FrvAz+mnyTQI3KnrrTU"`
	▶ Response body (10894 bytes)

Parameter	`slug`
Attack	`<script>alert(1);</script>`

Solution

Fase: Arquitetura e Design.

Use uma biblioteca verificada ou framework que não permita que essa vulnerabilidade ocorra, ou forneça construções/implementações que tornem essa vulnerabilidade mais fácil de evitar.

Exemplos de bibliotecas e frameworks que facilitam a geração de saída codificada adequadamente incluem a biblioteca Anti-XSS da Microsoft, o módulo de codificação OWASP ESAPI e o Apache Wicket.

Fases: Implementação | Arquitetura e Design.

Compreenda o contexto no qual seus dados serão usados e a codificação que será esperada. Isso é especialmente importante ao transmitir dados entre componentes diferentes ou ao gerar saídas que podem conter várias codificações ao mesmo tempo, como páginas da web ou mensagens de e-mail com várias partes. Estude todos os protocolos de comunicação e representações de dados esperados para determinar as estratégias de codificação necessárias.

Para quaisquer dados que serão enviados para outra página da web, especialmente quaisquer dados recebidos de entradas externas, use a codificação apropriada em todos os caracteres não alfanuméricos.

Consulte a Página de Dicas de Prevenção de XSS para obter mais detalhes sobre os tipos de codificação e escape que são necessários.

Fase: Arquitetura e Design.

Para todas as verificações de segurança realizadas no lado do cliente, certifique-se de que essas verificações sejam duplicadas no lado do servidor, a fim de evitar a CWE-602. Invasores podem ignorar as verificações do lado do cliente, modificando os valores após a realização das verificações ou alterando o cliente para remover as verificações do lado do cliente completamente. Em seguida, esses valores modificados poderiam ser enviados ao servidor.

Se disponível, use mecanismos estruturados que impõem automaticamente a separação entre dados e código. Esses mecanismos podem ser capazes de fornecer citação, codificação e validação relevantes automaticamente, em vez de depender do desenvolvedor para fornecer esse recurso em cada ponto onde a saída é gerada.

Fase: Implementação.

Para cada página web gerada, use e especifique uma codificação de caracteres, como ISO-8859-1 ou UTF-8. Quando uma codificação não é especificada, o navegador pode escolher uma codificação diferente, tentando adivinhar por eliminação qual codificação está realmente sendo usada pela página da web. Isso pode fazer com que o navegador da web trate certas sequências como especiais, abrindo o cliente para ataques XSS sutis. Consulte a CWE-116 para obter mais informações sobre mitigações relacionadas à codificação/escape.

Para ajudar a mitigar os ataques XSS contra cookie de sessão do usuário, defina o cookie de sessão como HttpOnly. Em navegadores que suportam o recurso HttpOnly (como versões mais recentes do Internet Explorer e Firefox), esse atributo pode impedir que o cookie de sessão do usuário seja acessível a scripts mal-intencionados do lado do cliente que usam document.cookie. Esta não é uma solução completa, já que HttpOnly não é compatível com todos os navegadores. Mais importante ainda, XMLHTTPRequest e outras poderosas tecnologias de navegadores fornecem acesso de leitura a cabeçalhos HTTP, incluindo o cabeçalho Set-Cookie no qual o sinalizador HttpOnly é definido.

Presuma que toda a entrada de dados é maliciosa. Use uma estratégia de validação de entrada "aceita como boa", ou seja, use uma lista de permissões de entradas aceitáveis que estejam estritamente em conformidade com as especificações. Rejeite quaisquer entradas que não estejam estritamente de acordo com as especificações ou transforme-as em algo que esteja. Não confie exclusivamente na procura de entradas maliciosas ou malformadas (ou seja, não confie em uma lista de negação). No entanto, as listas de negação podem ser úteis para detectar ataques em potencial ou determinar quais entradas estão tão malformadas que devem ser rejeitadas imediatamente.

	Ao executar a validação de entradas de dados, considere todas as propriedades potencialmente relevantes, incluindo comprimento, tipo de entrada, a gama completa de valores aceitáveis, entradas ausentes ou extras, sintaxe, consistência entre campos relacionados e conformidade com as regras de negócios. Como um exemplo de lógica de regra de negócios, "barco" pode ser sintaticamente válido porque contém apenas caracteres alfanuméricos, mas não é válido se você estiver esperando cores como "vermelho" ou "azul". Certifique-se de realizar a validação de entrada em interfaces bem definidas dentro do aplicativo. Isso ajudará a proteger o aplicativo, mesmo se um componente for reutilizado ou movido para outro lugar.

Risk=Informativo, Confidence=Médio (1)

https://i8z10k6wma.execute-api.us-east-1.amazonaws.com (1)

User Agent Fuzzer (1)

▼ POST https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/customers/delete

Alert tags	
Alert description	Check for differences in response based on fuzzed User Agent (eg. mobile sites, access as a Search Engine Crawler). Compares the response statuscode and the hashcode of the response body with the original response.

Request

▼ Request line and header section (313 bytes)

```
POST https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/customers/delete HTTP/1.1
host: i8z10k6wma.execute-api.us-east-1.amazonaws.com
user-agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1)
pragma: no-cache
cache-control: no-cache
content-type: application/json
content-length: 28
```

▼ Request body (28 bytes)

```
{
    "cpf": "12345678913"
}
```

Response

▼ Status line and header section (183 bytes)

```
HTTP/1.1 404 Not Found
Date: Thu, 08 Aug 2024 13:46:56 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 45
Connection: keep-alive
Apigw-Requestid: cMTcngfxIAMEVsg=
```

▼ Response body (45 bytes)

```
{"error":"User not found or already deleted"}
```

Parameter

Cabeçalho do Agente de Usuário

Attack	`Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1)`

Risk=`Informativo`, Confidence=`Baixo` (1)

`https://i8z10k6wma.execute-api.us-east-1.amazonaws.com` (1)

Re-examine Cache-control Directives (1)

▼ `POST https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/auth`

Alert tags	▪ WSTG-v42-ATHN-06 ▪ CWE-525
Alert description	The cache-control header has not been set properly or is missing, allowing the browser and proxies to cache content. For static assets like css, js, or image files this might be intended, however, the resources should be reviewed to ensure that no sensitive content will be cached.

Request	▼ Request line and header section (331 bytes)
	`POST https://i8z10k6wma.execute-api.us-east-1.amazonaws.com/lanchonete/auth HTTP/1.1`
	`host: i8z10k6wma.execute-api.us-east-1.amazonaws.com`
	`user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0`
	`pragma: no-cache`
	`cache-control: no-cache`
	`content-type: application/json`

```
content-length: 28
```

▼ Request body (28 bytes)

```
{
    "cpf": "12345678913"
}
```

Response

▼ Status line and header section (177 bytes)

```
HTTP/1.1 200 OK
Date: Thu, 08 Aug 2024 13:43:15 GMT
Content-Type: text/plain;
charset=utf-8
Content-Length: 485
Connection: keep-alive
Apigw-Requestid: cMS5vgiUIAMEVVA=
```

▼ Response body (485 bytes)

```
{
  "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
.eyJkYXRhIjp7Il9pZCI6IjY2YjRjYTA4NTY3
M2Q0MmMzNjQxMjY1YSIsIm5hbWUiOiJNb25hI
EJydWVuIiwiZW1haWwiOiJMYXVyZWw1MUBnbW
FpbC5jb20iLCJjcGYiOiIxMjM0NTY3ODkxMyI
sInBob25lIjoiNjAxLTI4OS05NjI5IiwiYWRk
cmVzcyI6IjIzNCBadWxhdWYgUmlkZ2VzIiwiY
3JlYXRlZEF0IjoiMjAyNC0wOC0wOFQxMzozNz
oxMi43MjZaIiwidXBkYXRlZEF0IjoiMjAyNC0
wOC0wOFQxMzozNzoxMi43MjZaIiwiX192Ijow
fSwiaWF0IjoxNzIzMTI0NTk1LCJleHAiOjE3M
jMxMjgxOTV9.OkmooDT8790iEmuk3lKsWAXHj
FCt9NRx5RbKdpcNojI"
}
```

Parameter

```
cache-control
```

Solution	For secure content, ensure the cache-control HTTP header is set with "no-cache, no-store, must-revalidate". If an asset should be cached consider setting the directives "public, max-age, immutable".

Appendix

Alert types

This section contains additional information on the types of alerts in the report.

Content Security Policy (CSP) Header Not Set

Source	raised by a passive scanner (Content Security Policy (CSP) Header Not Set)
CWE ID	693
WASC ID	15
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/

Erro de Formato de String

Source	raised by an active scanner (Erro de Formato de String)
CWE ID	134
WASC ID	6
Reference	▪ https://owasp.org/www-community/attacks/Format_string_attack

Missing Anti-clickjacking Header

Source	raised by a passive scanner (Anti-clickjacking Header)
CWE ID	1021
WASC ID	15
Reference	▪ https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

Fraqueza de script entre sites (persistente na resposta JSON)

Source	raised by an active scanner (Cross Site Scripting (Persistente))
CWE ID	79
WASC ID	8

Reference	▪ https://owasp.org/www-community/attacks/xss/ ▪ https://cwe.mitre.org/data/definitions/79.html

O servidor vaza informações por meio dos campos de cabeçalho de resposta HTTP "X-Powered-By"

Source	raised by a passive scanner (O servidor vaza informações por meio dos campos de cabeçalho de resposta HTTP "X-Powered-By")
CWE ID	200
WASC ID	13
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

Server Leaks Version Information via "Server" HTTP Response Header Field

Source	raised by a passive scanner (HTTP Server Response Header)
CWE ID	200
WASC ID	13
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/

Strict-Transport-Security Header Not Set

Source	raised by a passive scanner (Strict-Transport-Security Header)
CWE ID	319
WASC ID	15
Reference	https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html https://owasp.org/www-community/Security_Headers https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security https://caniuse.com/stricttransportsecurity https://datatracker.ietf.org/doc/html/rfc6797

X-Content-Type-Options Header Missing

Source	raised by a passive scanner (X-Content-Type-Options Header Missing)
CWE ID	693
WASC ID	15

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

Re-examine Cache-control Directives

Source	raised by a passive scanner (Re-examine Cache-control Directives)
CWE ID	525
WASC ID	13
Reference	- https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html#web-content-caching - https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cache-Control - https://grayduck.mn/2021/09/13/cache-control-recommendations/

User Agent Fuzzer

Source	raised by an active scanner (User Agent Fuzzer)
Reference	- https://owasp.org/wstg