BUB BOUNTY

IT NUMBER: IT22345332

NAME: G.P DINUJAYA THAMARA

WEEKEND BATCH

MALABE CAMPUS

Bug Bounty Platform – Hacker One

Bug Bounty Program - Booking.com

Scope

In Scope Assets

For in Scope Assets please refer to the Scope tab

Out-Of-Scope Applications Any application whether owned by Booking.com or third-party vendor **not included as an in-scope asset** will be mentioned on the scope tab as out of scope.

For Out Of Scope Assets please refer to the Scope tab

In-scope Vulnerabilities

Accepted, in-scope vulnerabilities include, but are not limited to:

- Disclosure of sensitive or personally identifiable information
- Cross-Site Scripting (XSS) - Please note, for XSS if the same issue is reported for the different subdomains but with the same root cause, it will be considered duplicate
- Cross-Site Request Forgery (CSRF) for sensitive functions in a privileged context
- Remote code execution (RCE)
- Authentication or authorization flaws, including insecure direct object references and authentication bypass
- Injection vulnerabilities, including SQL and XML injection
- Directory traversal
- Significant security misconfiguration with a verifiable vulnerability
- Account takeover by exploiting a vulnerability

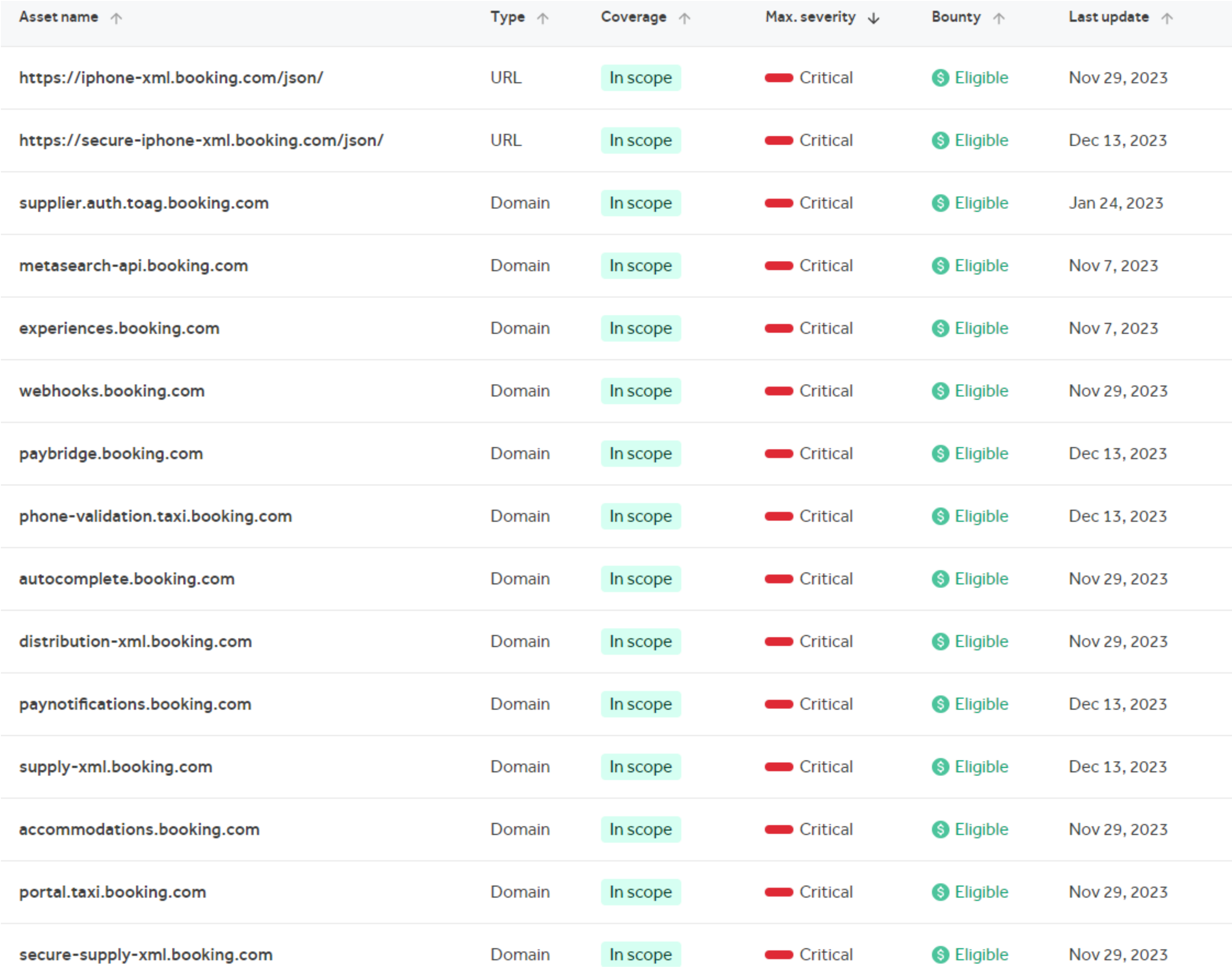
- SSRF
- XXE
- Subdomain takeover in *.booking.com domains

Out-Of-Scope Vulnerabilities Depending on their impact, not all reported issues may qualify for a monetary reward. However, all reports are reviewed on a case-by-case basis and any report that results in a change being made will at a minimum receive recognition. Please note that our **program terms and rules of engagement** still apply.

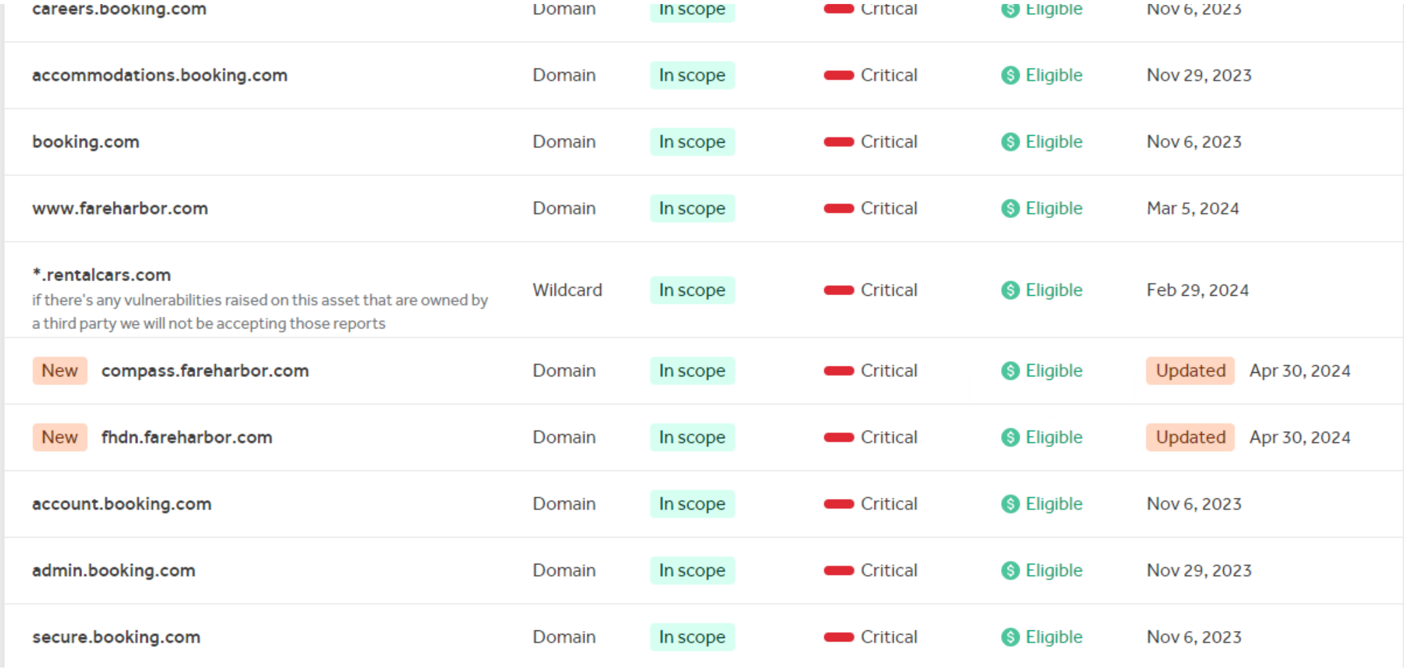
The following issues are outside the scope of our vulnerability rewards program:

- Any vulnerability which requires access to a compromised email account or Booking.com account for successful exploitation
- Vulnerabilities on Third Party Products
- Attacks requiring physical access to a user's device or network.
- Forms missing CSRF tokens (we require evidence of actual CSRF vulnerability)
- Login/Logout CSRF
- Missing security headers which do not lead directly to a vulnerability
- Use of a known-vulnerable library (without evidence of exploitability)
- Reports from automated tools or scans
- Social engineering of Booking staff or contractors
- Denial of Service attacks and/or reports on rate limiting issues
- Not enforcing certificate pinning
- Any issues that require a rooted or jailbroken device or a compromised device
- Clickjacking
- Improper session invalidation
- User enumeration
- Host header injections without a specific, demonstrable impact
- Self-XSS, which includes any payload entered by the victim

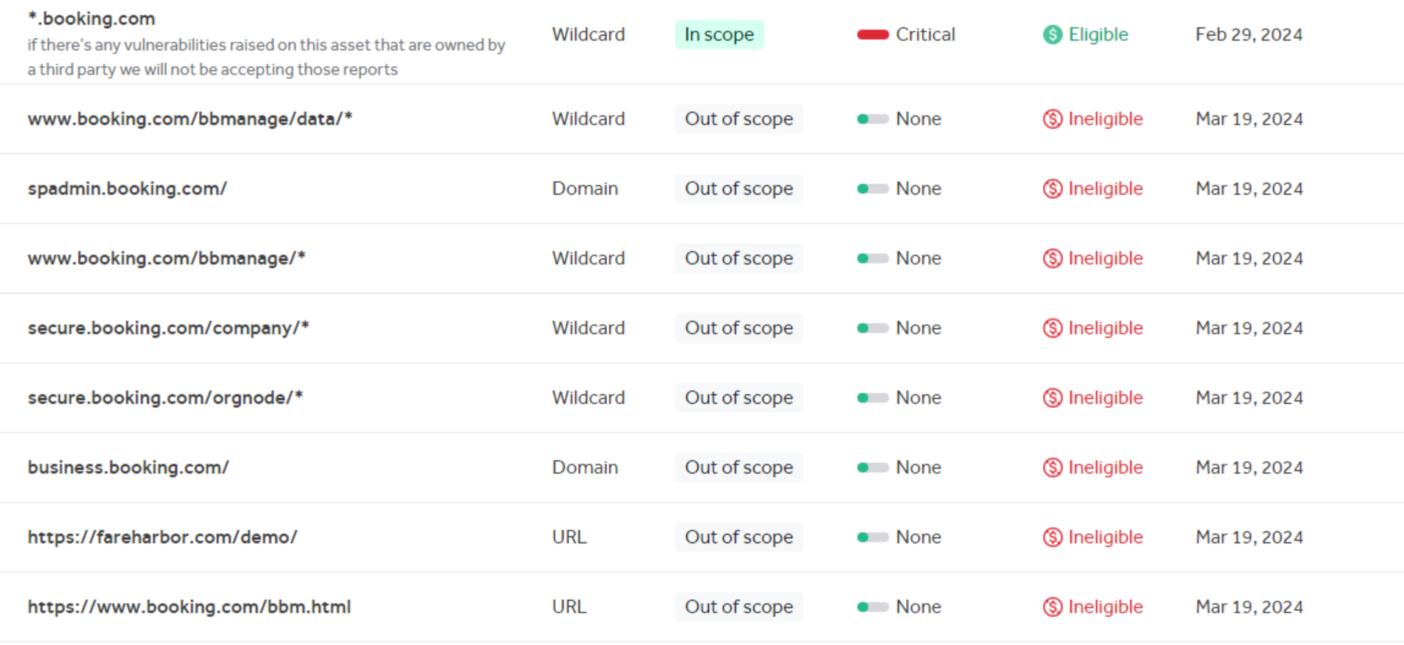
- Any vulnerabilities requiring significant and unlikely interaction by the victim, such as disabling browser controls
- Content spoofing without embedded HTML or JavaScript
- Hypothetical issues that do not have any practical impact
- Infrastructure vulnerabilities, including:
- Issues related to SSL certificates
- DNS configuration issues
- Server configuration issues (e.g. open ports, TLS versions, etc.)



Asset name ↑	Type ↑	Coverage ↑	Max. severity ↓	Bounty ↑	Last update ↑
https://iphone-xml.booking.com/json/	URL	In scope	Critical	Eligible	Nov 29, 2023
https://secure-iphone-xml.booking.com/json/	URL	In scope	Critical	Eligible	Dec 13, 2023
supplier.auth.toag.booking.com	Domain	In scope	Critical	Eligible	Jan 24, 2023
metasearch-api.booking.com	Domain	In scope	Critical	Eligible	Nov 7, 2023
experiences.booking.com	Domain	In scope	Critical	Eligible	Nov 7, 2023
webhooks.booking.com	Domain	In scope	Critical	Eligible	Nov 29, 2023
paybridge.booking.com	Domain	In scope	Critical	Eligible	Dec 13, 2023
phone-validation.taxi.booking.com	Domain	In scope	Critical	Eligible	Dec 13, 2023
autocomplete.booking.com	Domain	In scope	Critical	Eligible	Nov 29, 2023
distribution-xml.booking.com	Domain	In scope	Critical	Eligible	Nov 29, 2023
paynotifications.booking.com	Domain	In scope	Critical	Eligible	Dec 13, 2023
supply-xml.booking.com	Domain	In scope	Critical	Eligible	Dec 13, 2023
accommodations.booking.com	Domain	In scope	Critical	Eligible	Nov 29, 2023
portal.taxi.booking.com	Domain	In scope	Critical	Eligible	Nov 29, 2023
secure-supply-xml.booking.com	Domain	In scope	Critical	Eligible	Nov 29, 2023

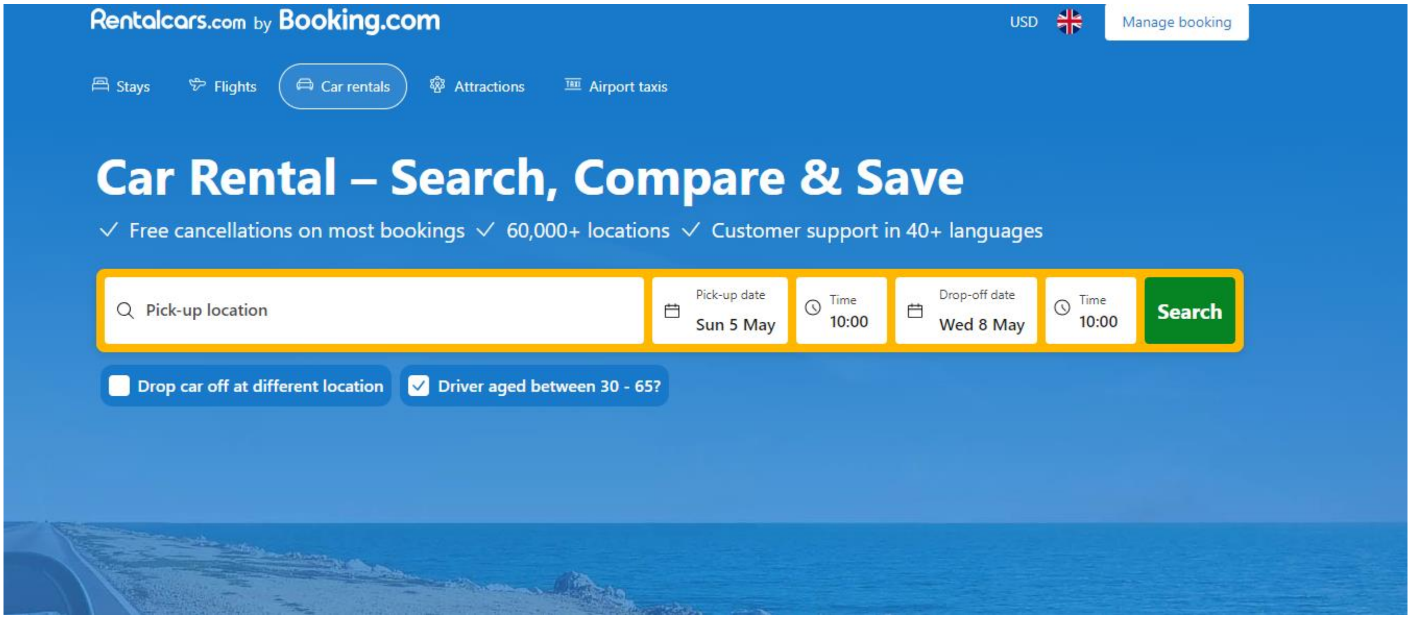


careers.booking.com	Domain	In scope	Critical	Eligible	Nov 6, 2023
accommodations.booking.com	Domain	In scope	Critical	Eligible	Nov 29, 2023
booking.com	Domain	In scope	Critical	Eligible	Nov 6, 2023
www.fareharbor.com	Domain	In scope	Critical	Eligible	Mar 5, 2024
*.rentalcars.com if there's any vulnerabilities raised on this asset that are owned by a third party we will not be accepting those reports	Wildcard	In scope	Critical	Eligible	Feb 29, 2024
New compass.fareharbor.com	Domain	In scope	Critical	Eligible	Updated Apr 30, 2024
New fhdn.fareharbor.com	Domain	In scope	Critical	Eligible	Updated Apr 30, 2024
account.booking.com	Domain	In scope	Critical	Eligible	Nov 6, 2023
admin.booking.com	Domain	In scope	Critical	Eligible	Nov 29, 2023
secure.booking.com	Domain	In scope	Critical	Eligible	Nov 6, 2023



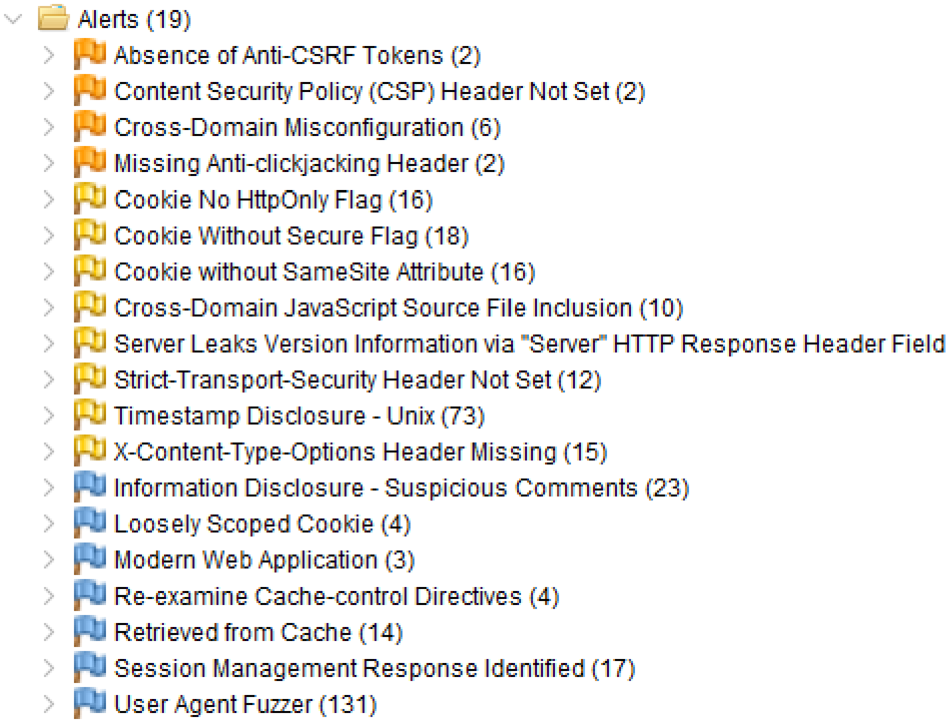
*.booking.com if there's any vulnerabilities raised on this asset that are owned by a third party we will not be accepting those reports	Wildcard	In scope	Critical	Eligible	Feb 29, 2024
www.booking.com/bbmanage/data/*	Wildcard	Out of scope	None	Ineligible	Mar 19, 2024
spadmin.booking.com/	Domain	Out of scope	None	Ineligible	Mar 19, 2024
www.booking.com/bbmanage/*	Wildcard	Out of scope	None	Ineligible	Mar 19, 2024
secure.booking.com/company/*	Wildcard	Out of scope	None	Ineligible	Mar 19, 2024
secure.booking.com/orgnode/*	Wildcard	Out of scope	None	Ineligible	Mar 19, 2024
business.booking.com/	Domain	Out of scope	None	Ineligible	Mar 19, 2024
https://fareharbor.com/demo/	URL	Out of scope	None	Ineligible	Mar 19, 2024
https://www.booking.com/bbm.html	URL	Out of scope	None	Ineligible	Mar 19, 2024

www.rentalcars.com (*rentalcars.com)



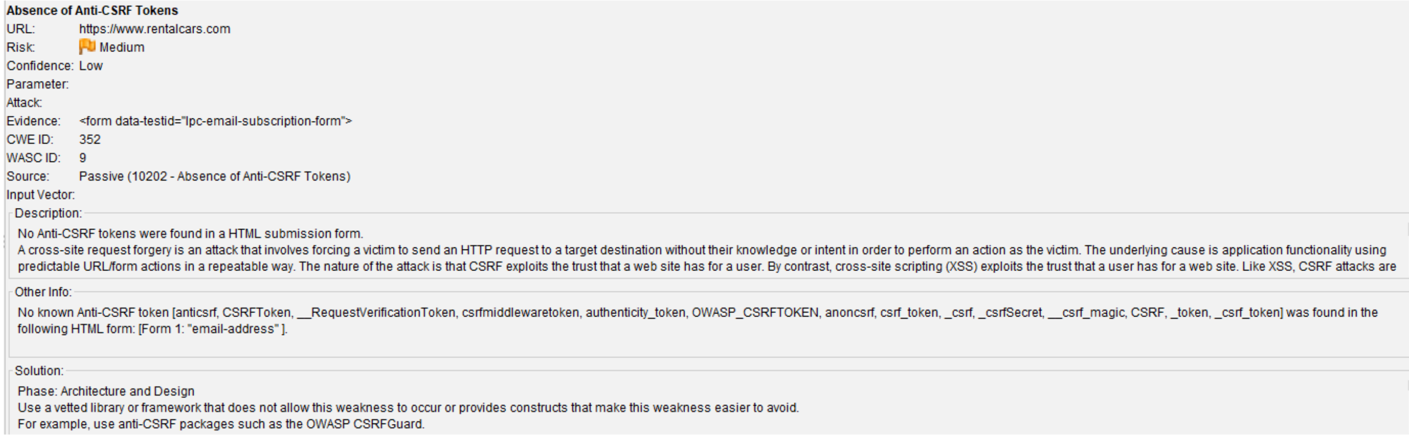
Rentalcars.com by Booking.com
USD
Manage booking
Stays
Flights
Car rentals
Attractions
Airport taxis
Car Rental – Search, Compare & Save
✓ Free cancellations on most bookings ✓ 60,000+ locations ✓ Customer support in 40+ languages
Pick-up location
Pick-up date Sun 5 May
Time 10:00
Drop-off date Wed 8 May
Time 10:00
Search
□ Drop car off at different location
☑ Driver aged between 30 - 65?

The ZAP automated scan gives the following results



Alerts (19)

- Absence of Anti-CSRF Tokens (2)
- Content Security Policy (CSP) Header Not Set (2)
- Cross-Domain Misconfiguration (6)
- Missing Anti-clickjacking Header (2)
- Cookie No HttpOnly Flag (16)
- Cookie Without Secure Flag (18)
- Cookie without SameSite Attribute (16)
- Cross-Domain JavaScript Source File Inclusion (10)
- Server Leaks Version Information via "Server" HTTP Response Header Field
- Strict-Transport-Security Header Not Set (12)
- Timestamp Disclosure - Unix (73)
- X-Content-Type-Options Header Missing (15)
- Information Disclosure - Suspicious Comments (23)
- Loosely Scoped Cookie (4)
- Modern Web Application (3)
- Re-examine Cache-control Directives (4)
- Retrieved from Cache (14)
- Session Management Response Identified (17)
- User Agent Fuzzer (131)



Absence of Anti-CSRF Tokens

URL: https://www.rentalcars.com
Risk: Medium
Confidence: Low
Parameter:
Attack:
Evidence: <form data-testid="lpc-email-subscription-form">
CWE ID: 352
WASC ID: 9
Source: Passive (10202 - Absence of Anti-CSRF Tokens)
Input Vector:

Description:

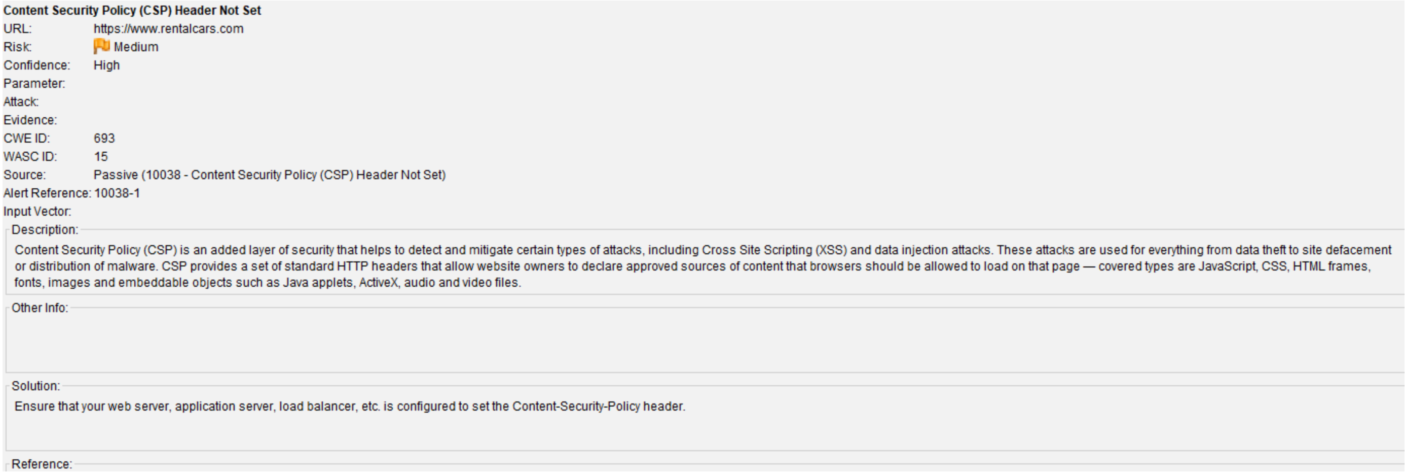
No Anti-CSRF tokens were found in a HTML submission form.
A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are

Other Info:

No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "email-address"].

Solution:

Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
For example, use anti-CSRF packages such as the OWASP CSRFGuard.



Content Security Policy (CSP) Header Not Set

URL: https://www.rentalcars.com
Risk: Medium
Confidence: High
Parameter:
Attack:
Evidence:
CWE ID: 693
WASC ID: 15
Source: Passive (10038 - Content Security Policy (CSP) Header Not Set)
Alert Reference: 10038-1
Input Vector:

Description:

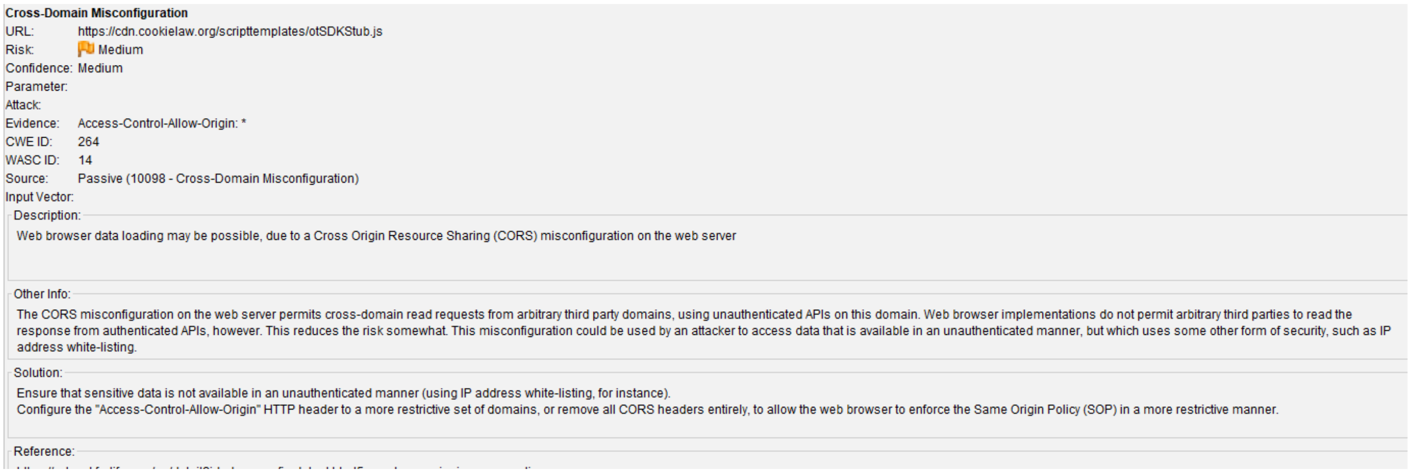
Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a 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.

Other Info:

Solution:

Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Reference:



Cross-Domain Misconfiguration

URL: https://cdn.cookielaw.org/scripttemplates/otSDKStub.js
Risk: Medium
Confidence: Medium
Parameter:
Attack:
Evidence: Access-Control-Allow-Origin: *
CWE ID: 264
WASC ID: 14
Source: Passive (10098 - Cross-Domain Misconfiguration)
Input Vector:

Description:

Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server

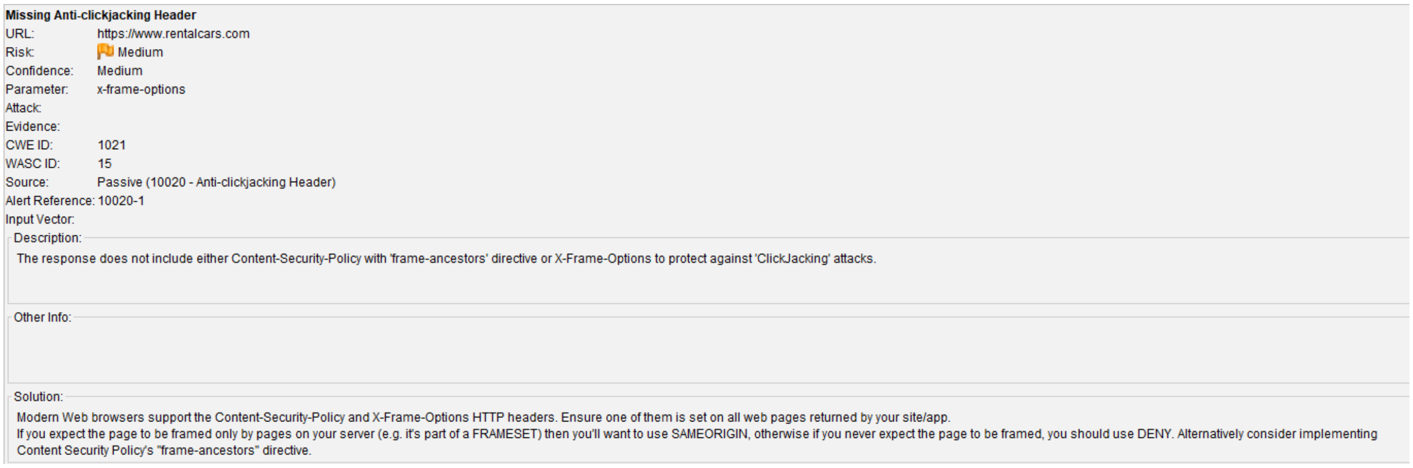
Other Info:

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.

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:



Missing Anti-clickjacking Header

URL: https://www.rentalcars.com
Risk: Medium
Confidence: Medium
Parameter: x-frame-options
Attack:
Evidence:
CWE ID: 1021
WASC ID: 15
Source: Passive (10020 - Anti-clickjacking Header)
Alert Reference: 10020-1
Input Vector:

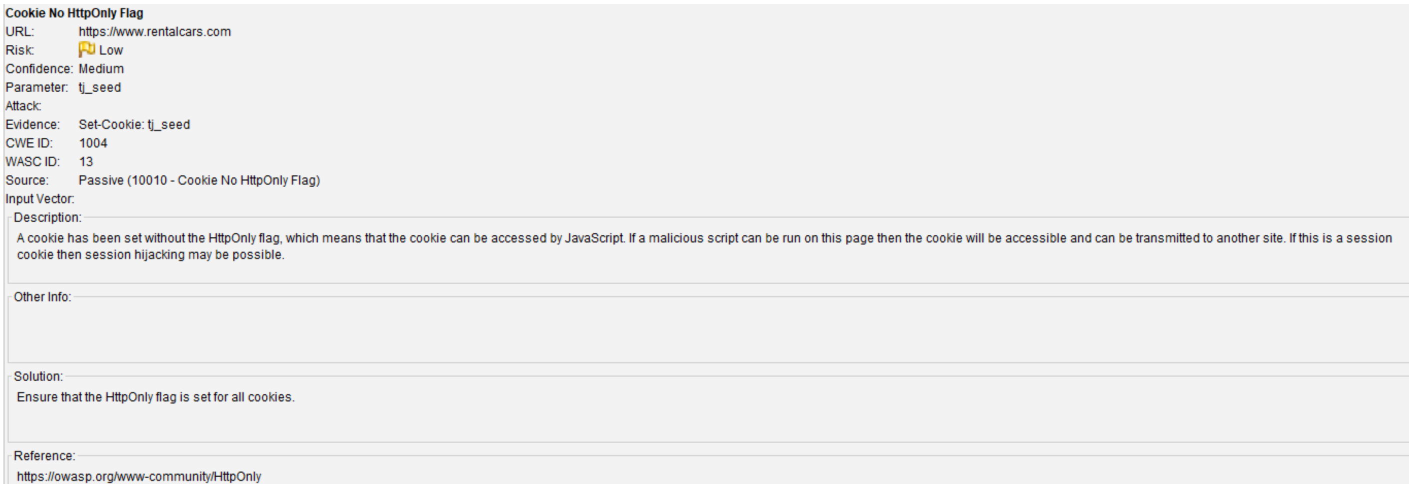
Description:

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

Other Info:

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.



Cookie No HttpOnly Flag

URL: https://www.rentalcars.com

Risk: 🚩 Low

Confidence: Medium

Parameter: tj_seed

Attack:

Evidence: Set-Cookie: tj_seed

CWE ID: 1004

WASC ID: 13

Source: Passive (10010 - Cookie No HttpOnly Flag)

Input Vector:

Description:

A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.

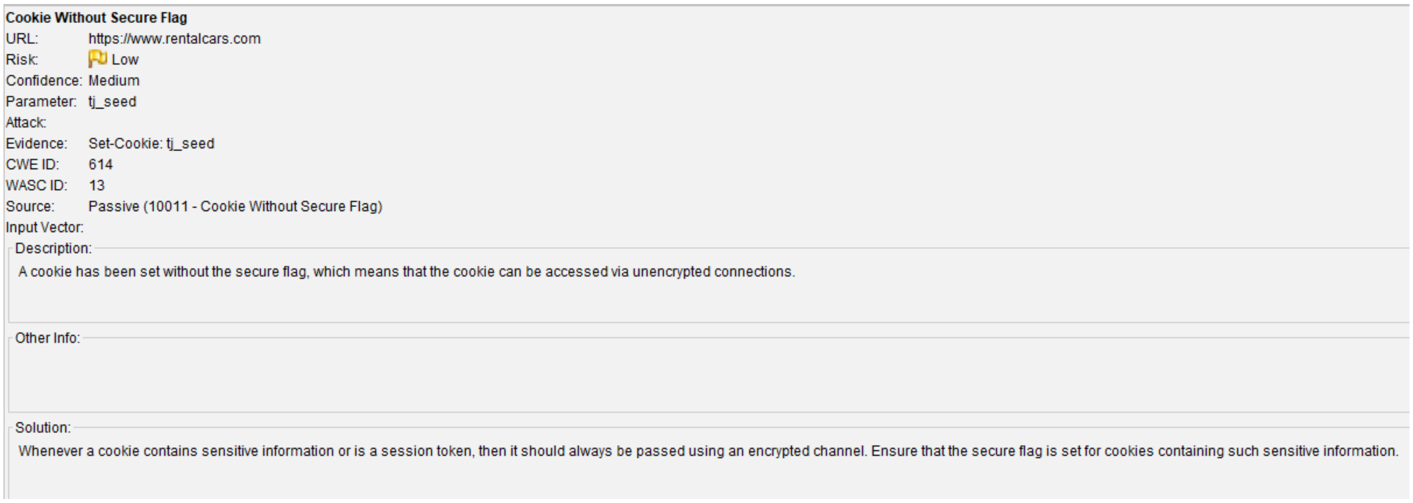
Other Info:

Solution:

Ensure that the HttpOnly flag is set for all cookies.

Reference:

https://owasp.org/www-community/HttpOnly



Cookie Without Secure Flag

URL: https://www.rentalcars.com

Risk: 🚩 Low

Confidence: Medium

Parameter: tj_seed

Attack:

Evidence: Set-Cookie: tj_seed

CWE ID: 614

WASC ID: 13

Source: Passive (10011 - Cookie Without Secure Flag)

Input Vector:

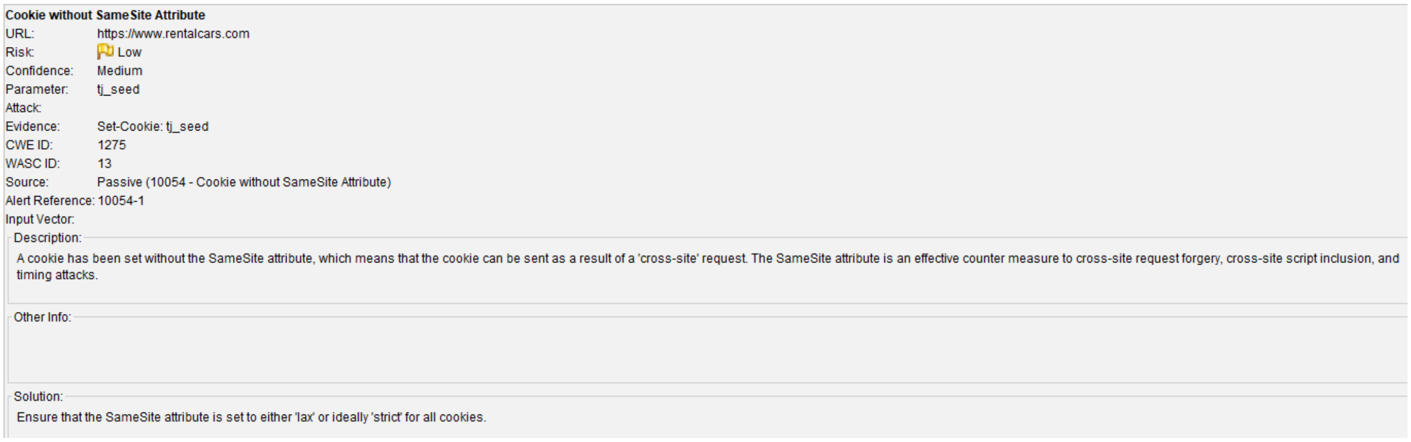
Description:

A cookie has been set without the secure flag, which means that the cookie can be accessed via unencrypted connections.

Other Info:

Solution:

Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such sensitive information.



Cookie without SameSite Attribute

URL: https://www.rentalcars.com

Risk: 🚩 Low

Confidence: Medium

Parameter: tj_seed

Attack:

Evidence: Set-Cookie: tj_seed

CWE ID: 1275

WASC ID: 13

Source: Passive (10054 - Cookie without SameSite Attribute)

Alert Reference: 10054-1

Input Vector:

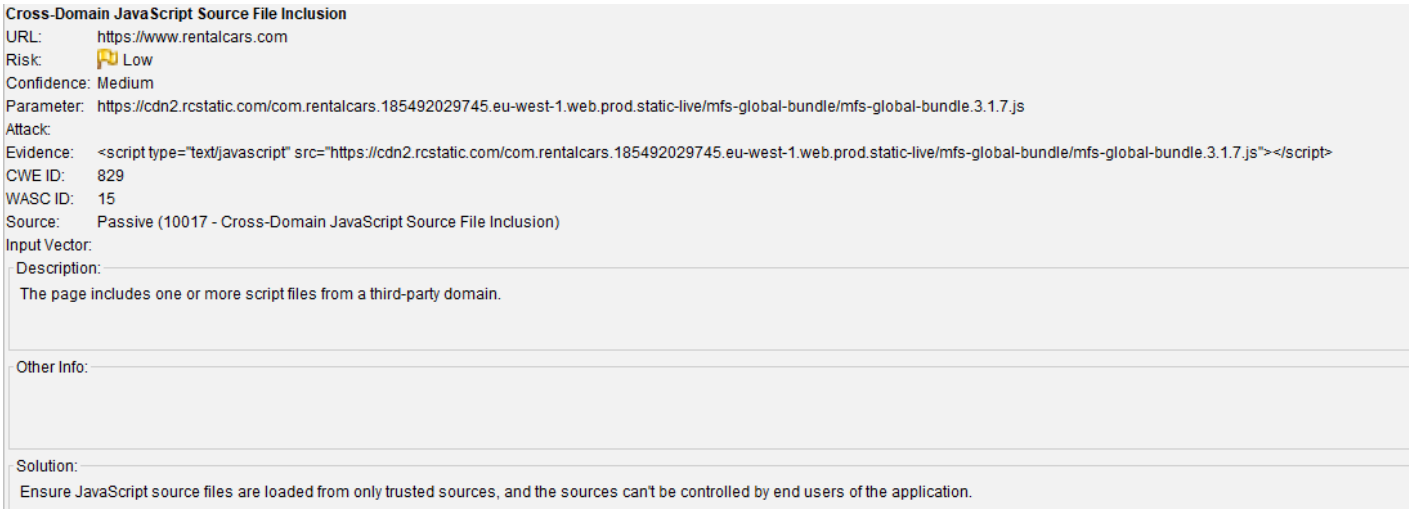
Description:

A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.

Other Info:

Solution:

Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.



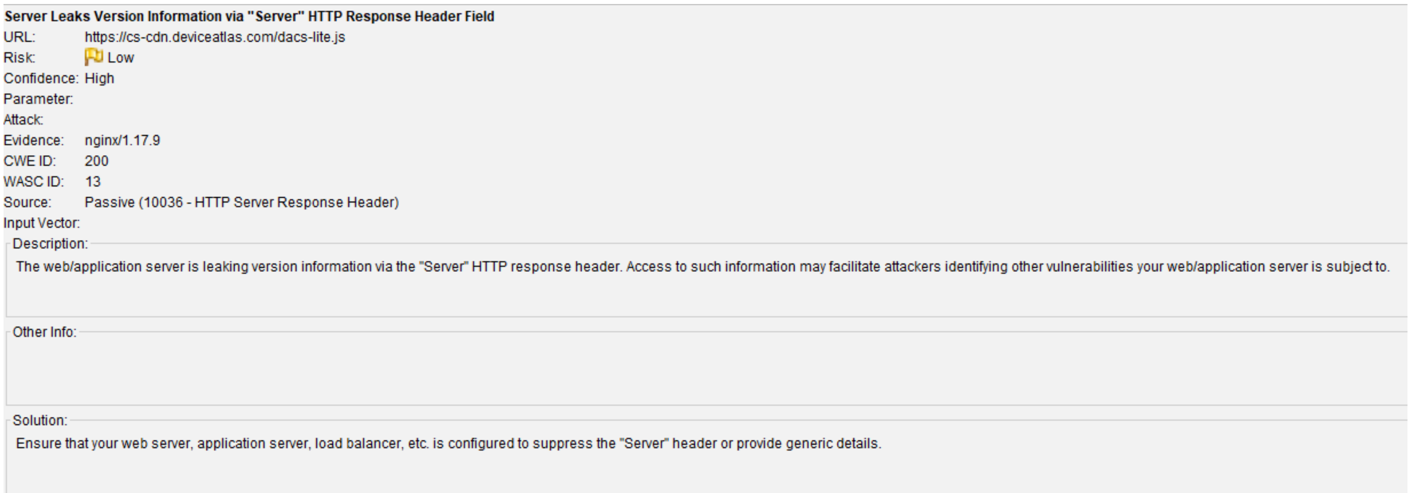
Cross-Domain JavaScript Source File Inclusion

URL: https://www.rentalcars.com
Risk: ⚑ Low
Confidence: Medium
Parameter: https://cdn2.rcstatic.com/com.rentalcars.185492029745.eu-west-1.web.prod.static-live/mfs-global-bundle/mfs-global-bundle.3.1.7.js
Attack:
Evidence: <script type="text/javascript" src="https://cdn2.rcstatic.com/com.rentalcars.185492029745.eu-west-1.web.prod.static-live/mfs-global-bundle/mfs-global-bundle.3.1.7.js"></script>
CWE ID: 829
WASC ID: 15
Source: Passive (10017 - Cross-Domain JavaScript Source File Inclusion)
Input Vector:

Description:
The page includes one or more script files from a third-party domain.

Other Info:

Solution:
Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.



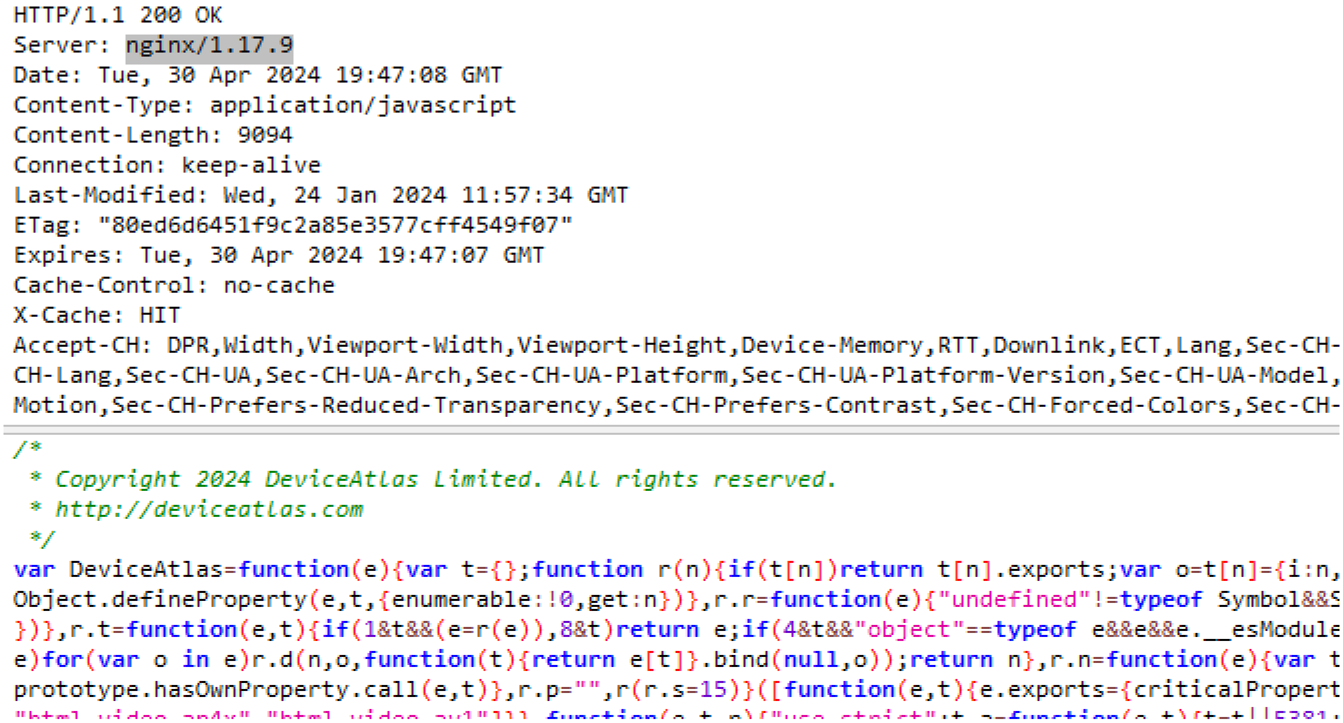
Server Leaks Version Information via "Server" HTTP Response Header Field

URL: https://cs-cdn.deviceatlas.com/dacs-lite.js
Risk: ⚑ Low
Confidence: High
Parameter:
Attack:
Evidence: nginx/1.17.9
CWE ID: 200
WASC ID: 13
Source: Passive (10036 - HTTP Server Response Header)
Input Vector:

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.

Other Info:

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



```
HTTP/1.1 200 OK
Server: nginx/1.17.9
Date: Tue, 30 Apr 2024 19:47:08 GMT
Content-Type: application/javascript
Content-Length: 9094
Connection: keep-alive
Last-Modified: Wed, 24 Jan 2024 11:57:34 GMT
ETag: "80ed6d6451f9c2a85e3577cff4549f07"
Expires: Tue, 30 Apr 2024 19:47:07 GMT
Cache-Control: no-cache
X-Cache: HIT
Accept-CH: DPR,Width,Viewport-Width,Viewport-Height,Device-Memory,RTT,Downlink,ECT,Lang,Sec-CH-
CH-Lang,Sec-CH-UA,Sec-CH-UA-Arch,Sec-CH-UA-Platform,Sec-CH-UA-Platform-Version,Sec-CH-UA-Model,
Motion,Sec-CH-Prefers-Reduced-Transparency,Sec-CH-Prefers-Contrast,Sec-CH-Forced-Colors,Sec-CH-
```

```
/*
 * Copyright 2024 DeviceAtlas Limited. All rights reserved.
 * http://deviceatlas.com
 */
var DeviceAtlas=function(e){var t={};function r(n){if(t[n])return t[n].exports;var o=t[n]={i:n,
Object.defineProperty(e,t,{enumerable:!0,get:n})},r.r=function(e){"undefined"!=typeof Symbol&&S
})},r.t=function(e,t){if(1&t&&(e=r(e)),8&t)return e;if(4&t&&"object"==typeof e&&e&&e.__esModule
e)for(var o in e)r.d(n,o,function(t){return e[t]}.bind(null,o));return n},r.n=function(e){var t
prototype.hasOwnProperty.call(e,t)},r.p="",r(r.s=15)}([function(e,t){e.exports={criticalPropert
"html_video_ap4x","html_video_av1"]]},function(e,t,n){"use strict";t.a=function(e,t){t=t||5381;
```

According to the HTTP response header the server is nginx/1.17.9

The are the vulnerabilities which are related to nginx/1.17.9

Remote code execution in nginx -But the patch is available
https://www.cybersecurity-help.cz/vdb/SB2021052543

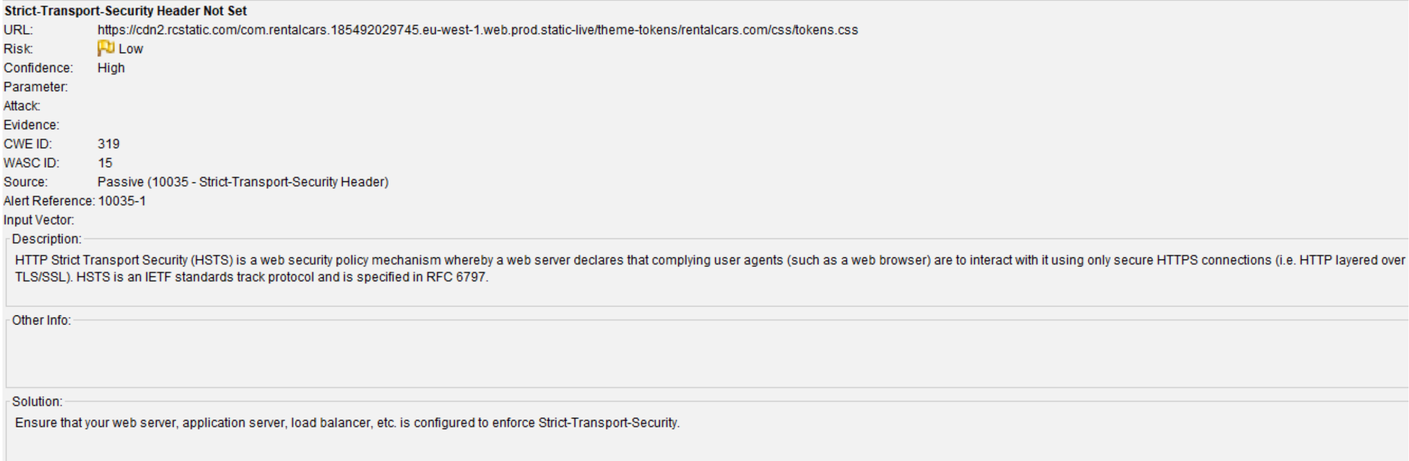
https://www.cybersecurity-help.cz/vdb/SB2022101941

Security restrictions bypass in nginx- The patch is available.

https://www.cybersecurity-help.cz/vdb/SB2022010903

In the following site it contains some vulnerabilities related to this server

https://snyk.io/test/docker/nginx%3A1.17.9-alpine



```
Strict-Transport-Security Header Not Set
URL:            https://cdn2.rcstatic.com/com.rentalcars.185492029745.eu-west-1.web.prod.static-live/theme-tokens/rentalcars.com/css/tokens.css
Risk:           Low
Confidence:     High
Parameter:
Attack:
Evidence:
CWE ID:         319
WASC ID:        15
Source:         Passive (10035 - Strict-Transport-Security Header)
Alert Reference: 10035-1
Input Vector:
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.

Other Info:

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

It's widely accepted that HTTPS is far more secure than HTTP. However, if you're encountering the "HSTS missing from HTTPS server" message, then this protocol could be putting your site at risk.

Fortunately, it is possible to close this serious security loophole. Even if you haven't encountered this error message, any site that redirects from HTTP to HTTPS is vulnerable to this exploit. Therefore, it's still wise to take a proactive approach and fix this flaw.

To help keep visitors safe, it's not uncommon for sites to perform HTTPS redirection. This redirection forwards visitors from an HTTP to an HTTPS version of the website.

A user may explicitly enter HTTP into their browser's address bar, or follow a link that points to an HTTP version of the site. In these scenarios, a redirect can prevent malicious third parties from stealing the visitor's data.

However, no technology is perfect. If your site does use HTTPS redirects, then you may be susceptible to a Man-In-The-Middle (MITM) attack known as Secure Sockets Layer (SSL) Stripping. As part of this attack, the hacker will block the redirection request and prevent the browser from loading your site over the HTTPS protocol. As a result, the visitor will access your website via HTTP, which makes it much easier for hackers to steal data.

Alternatively, the attacker might intercept the redirect and forward visitors to a clone version of your site. At this point, the hacker can steal any data that the user shares, including passwords and payment information. Some hackers might also try to trick visitors into downloading malicious software.

It's also possible for hackers to steal a session cookie over an unsecured connection, in an attack known as cookie hijacking. These cookies can contain a wealth of information, including usernames, passwords, and even credit card details.

To protect your visitors against these attacks, we recommend enabling HTTP Strict Transport Security (HSTS). This protocol forces the browser to ignore any direct requests and load your site over HTTPS.

For further information visit this site :
https://kinsta.com/knowledgebase/hsts-missing-from-https-server/

How to fix Strict-Transport-Security Header Not Set

Your server should be configured to include the header, e.g.

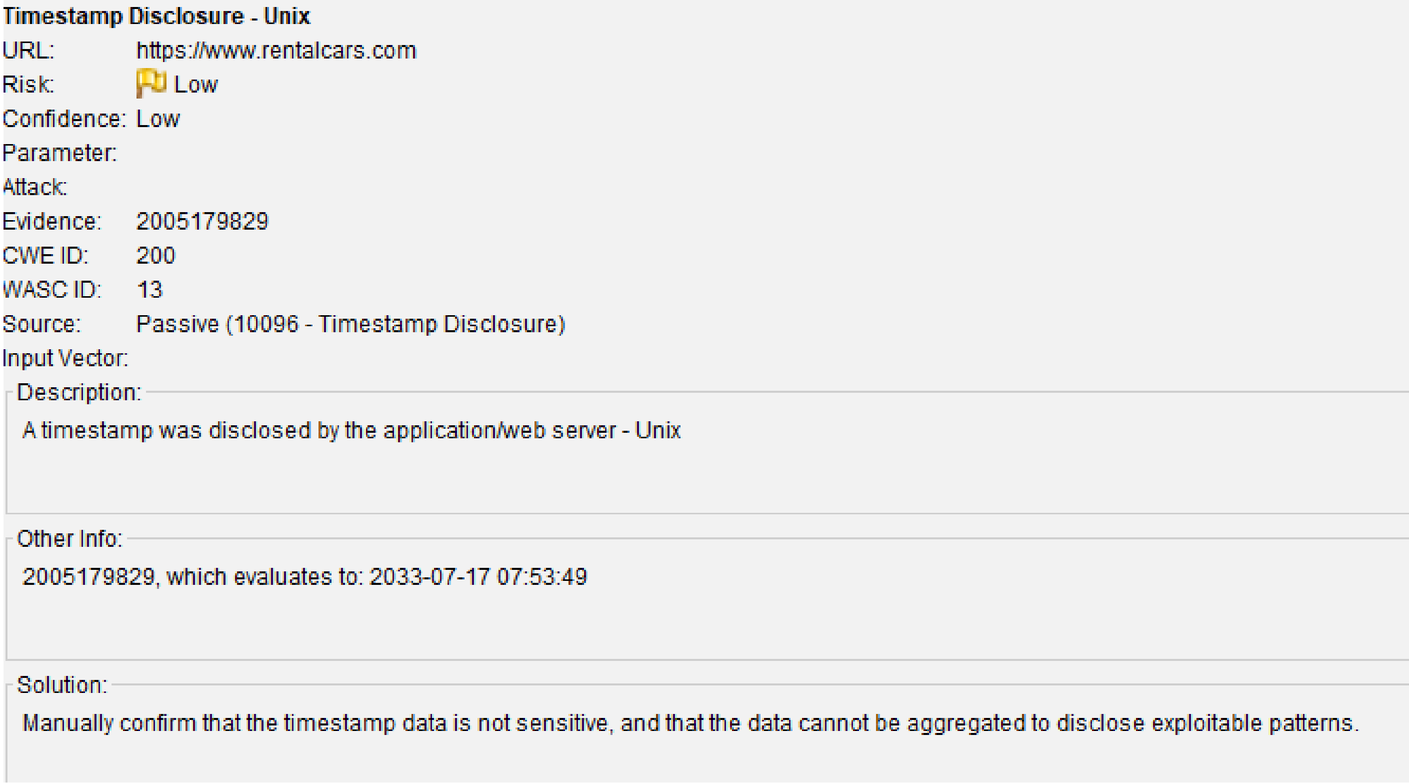
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload

max-age is the time in seconds indicating how long the browser should remember that a site is accessible via HTTPS only. The time is refreshed (set again to max-age) after each request to the domain. In the example the time is equal to one year.

includeSubDomains indicates that HTTPS restriction is valid for subdomains too (optional, but recommended).

preload is optional (and not the part of the official specification) and allows you to add your website to a preload list maintained by Google. This means that your domain will be hardcoded in the list and browsers will never try to connect using an insecure connection. Note that it will have permanent consequences and switching back to HTTP may be troublesome.

https://scanrepeat.com/web-security-knowledge-base/strict-transport-security-header-not-set



Timestamp Disclosure - Unix

URL:	https://www.rentalcars.com
Risk:	Low
Confidence:	Low
Parameter:	
Attack:	
Evidence:	2005179829
CWE ID:	200
WASC ID:	13
Source:	Passive (10096 - Timestamp Disclosure)
Input Vector:	

Description:

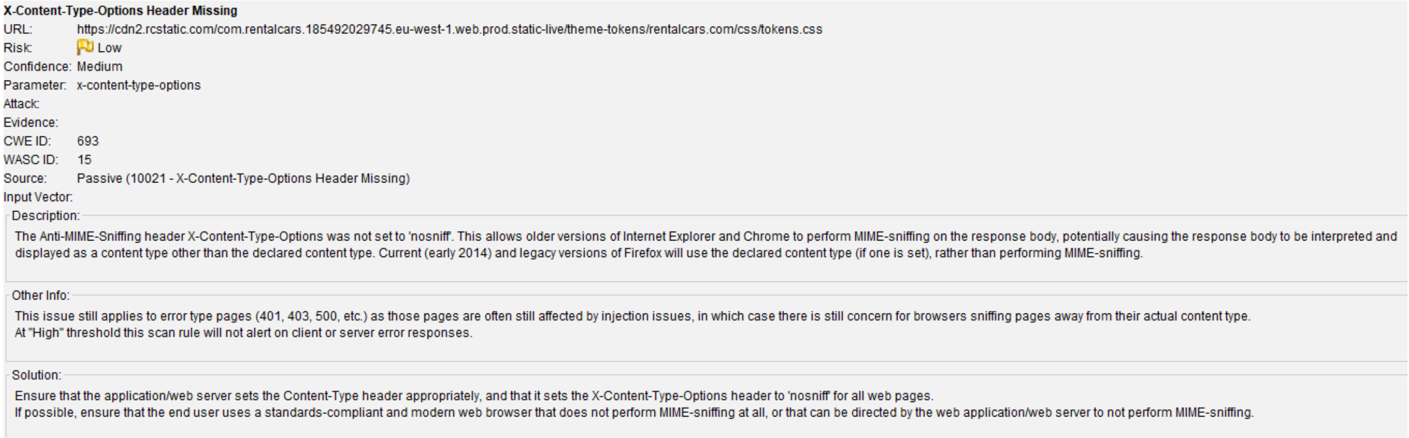
A timestamp was disclosed by the application/web server - Unix

Other Info:

2005179829, which evaluates to: 2033-07-17 07:53:49

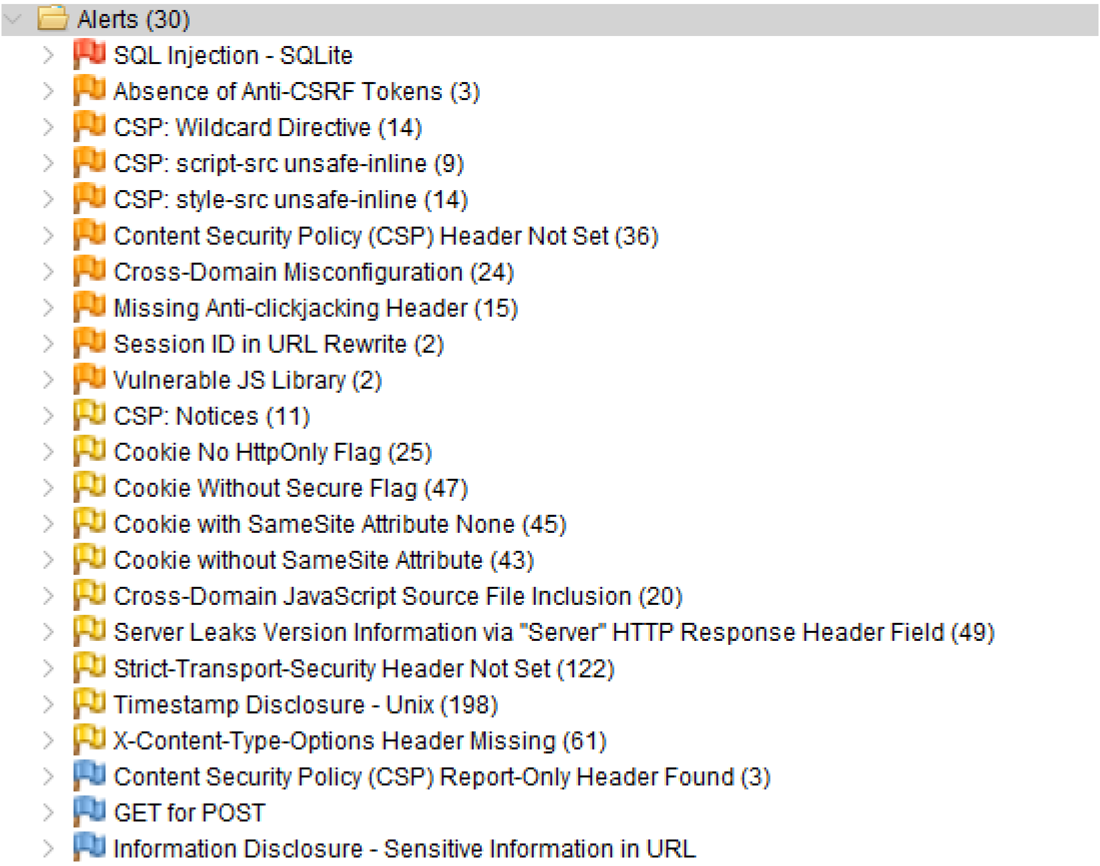
Solution:

Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.

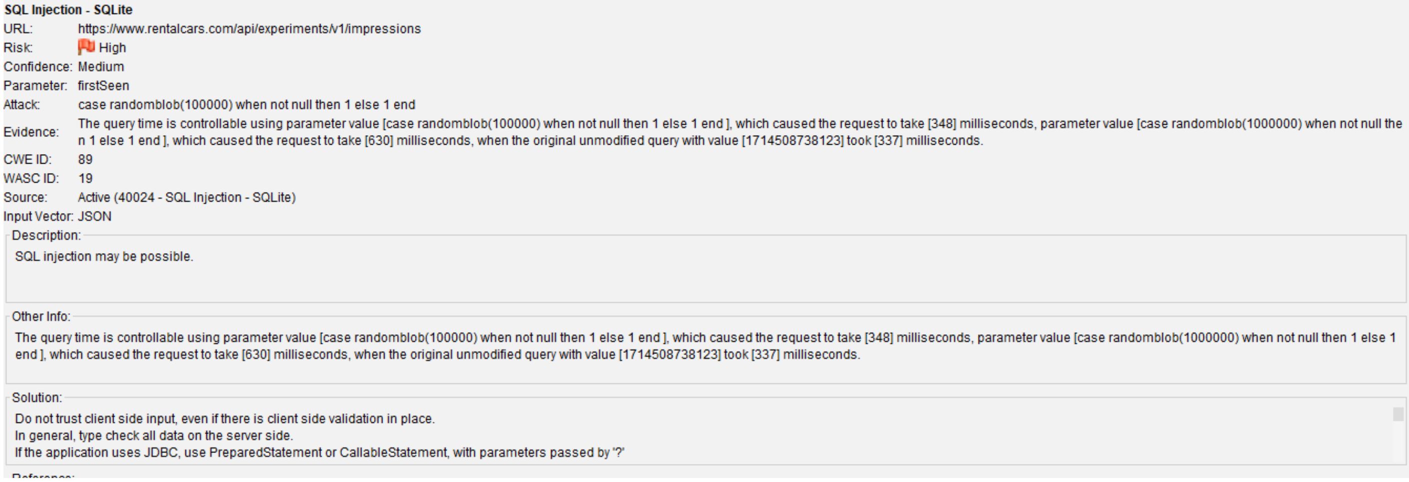


```
X-Content-Type-Options Header Missing
URL:        https://cdn2.rcstatic.com/com.rentalcars.185492029745.eu-west-1.web.prod.static-live/theme-tokens/rentalcars.com/css/tokens.css
Risk:       Low
Confidence: Medium
Parameter:  x-content-type-options
Attack:
Evidence:
CWE ID:     693
WASC ID:    15
Source:     Passive (10021 - X-Content-Type-Options Header Missing)
Input Vector:
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.

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.
```

Once the automated scanner is finished then I moved into manual explore tab and check the site manually the got 30 alerts



```
Alerts (30)
  SQL Injection - SQLite
  Absence of Anti-CSRF Tokens (3)
  CSP: Wildcard Directive (14)
  CSP: script-src unsafe-inline (9)
  CSP: style-src unsafe-inline (14)
  Content Security Policy (CSP) Header Not Set (36)
  Cross-Domain Misconfiguration (24)
  Missing Anti-clickjacking Header (15)
  Session ID in URL Rewrite (2)
  Vulnerable JS Library (2)
  CSP: Notices (11)
  Cookie No HttpOnly Flag (25)
  Cookie Without Secure Flag (47)
  Cookie with SameSite Attribute None (45)
  Cookie without SameSite Attribute (43)
  Cross-Domain JavaScript Source File Inclusion (20)
  Server Leaks Version Information via "Server" HTTP Response Header Field (49)
  Strict-Transport-Security Header Not Set (122)
  Timestamp Disclosure - Unix (198)
  X-Content-Type-Options Header Missing (61)
  Content Security Policy (CSP) Report-Only Header Found (3)
  GET for POST
  Information Disclosure - Sensitive Information in URL
```



SQL Injection - SQLite
URL: https://www.rentalcars.com/api/experiments/v1/impressions
Risk: 🚩 High
Confidence: Medium
Parameter: firstSeen
Attack: case randomblob(100000) when not null then 1 else 1 end
Evidence: The query time is controllable using parameter value [case randomblob(100000) when not null then 1 else 1 end], which caused the request to take [348] milliseconds, parameter value [case randomblob(1000000) when not null then 1 else 1 end], which caused the request to take [630] milliseconds, when the original unmodified query with value [1714508738123] took [337] milliseconds.
CWE ID: 89
WASC ID: 19
Source: Active (40024 - SQL Injection - SQLite)
Input Vector: JSON

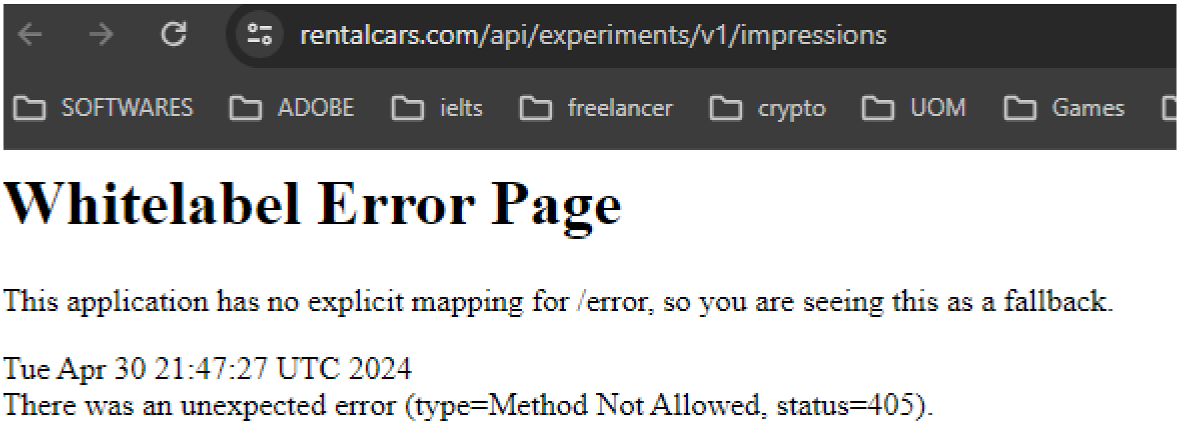
Description:
SQL injection may be possible.

Other Info:
The query time is controllable using parameter value [case randomblob(100000) when not null then 1 else 1 end], which caused the request to take [348] milliseconds, parameter value [case randomblob(1000000) when not null then 1 else 1 end], which caused the request to take [630] milliseconds, when the original unmodified query with value [1714508738123] took [337] milliseconds.

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 '?'

Reference:

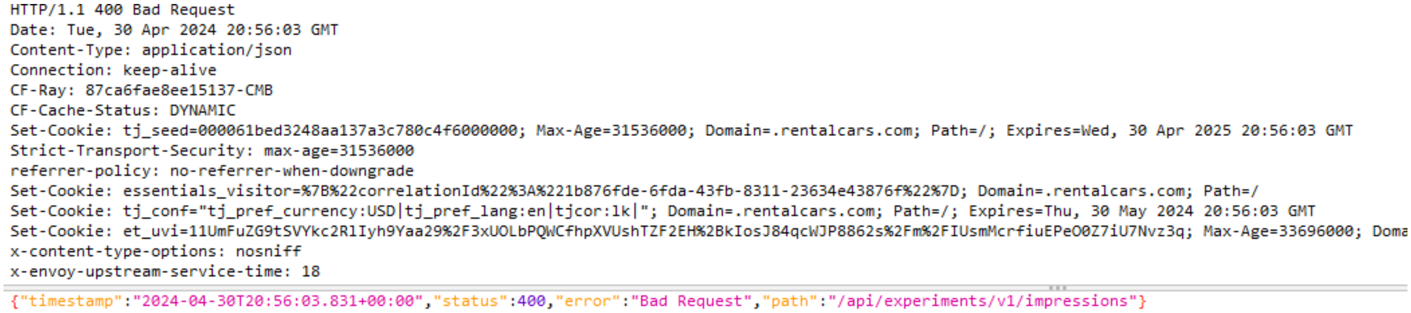
In here there is SQL injection -SQLite high risk alert now here is that page



Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Tue Apr 30 21:47:27 UTC 2024
There was an unexpected error (type=Method Not Allowed, status=405).

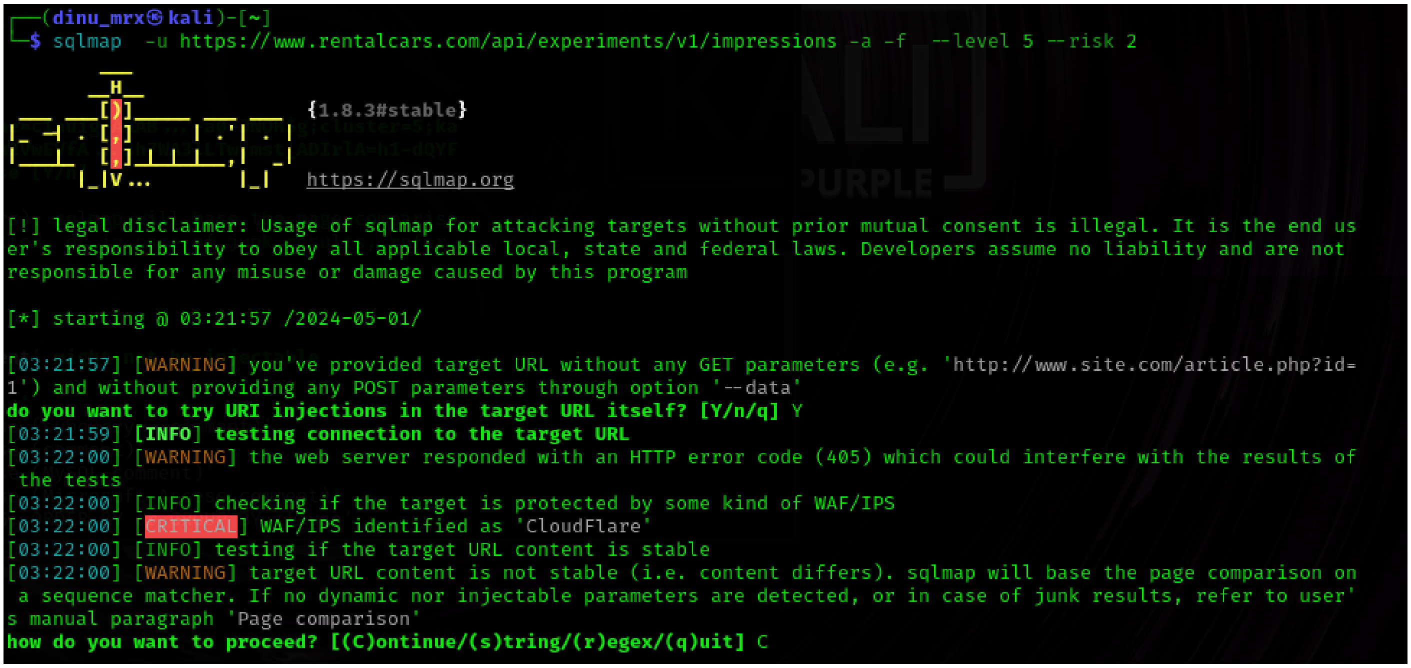


```
HTTP/1.1 400 Bad Request
Date: Tue, 30 Apr 2024 20:56:03 GMT
Content-Type: application/json
Connection: keep-alive
CF-Ray: 87ca6fae8ee15137-CMB
CF-Cache-Status: DYNAMIC
Set-Cookie: tj_seed=000061bed3248aa137a3c780c4f6000000; Max-Age=31536000; Domain=.rentalcars.com; Path=/; Expires=Wed, 30 Apr 2025 20:56:03 GMT
Strict-Transport-Security: max-age=31536000
referrer-policy: no-referrer-when-downgrade
Set-Cookie: essentials_visitor=%7B%22correlationId%22%3A%221b876fde-6fda-43fb-8311-23634e43876f%22%7D; Domain=.rentalcars.com; Path=/
Set-Cookie: tj_conf="tj_pref_currency:USD|tj_pref_lang:en|tjcor:lk|"; Domain=.rentalcars.com; Path=/; Expires=Thu, 30 May 2024 20:56:03 GMT
Set-Cookie: et_uvi=11UmFuZG9tSVYkc2RlIyh9Yaa29%2F3xUOLbPQWCfhpXVUshTZF2EH%2BkIosJ84qcWJP8862s%2Fm%2FIUsmMcrfiuEPeO0Z7iU7Nvz3q; Max-Age=33696000; Doma
x-content-type-options: nosniff
x-envoy-upstream-service-time: 18
```

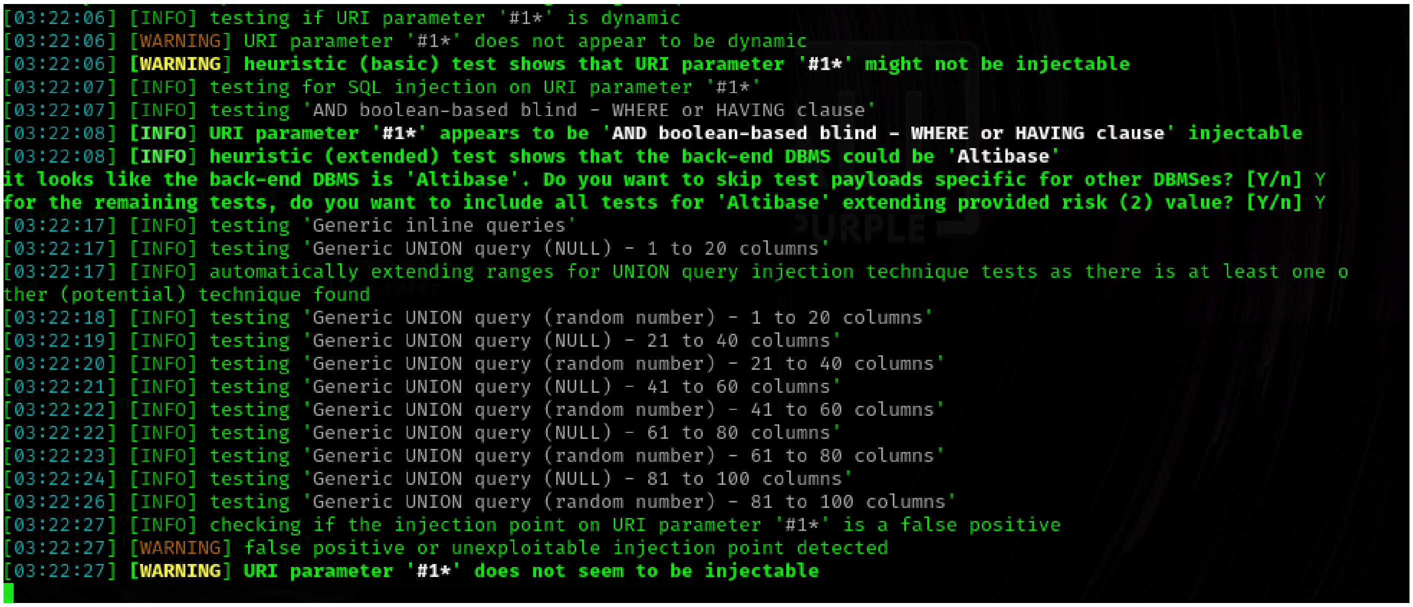
```
{"timestamp":"2024-04-30T20:56:03.831+00:00","status":400,"error":"Bad Request","path":"/api/experiments/v1/impressions"}
```

Other errors are quite similar to previous scan.

Results that were obtained when scanning through the SQLmap.In SQLmap I did the injections to the site that gives the above SQL Injection error.



```
┌──(dinu_mrx㉿kali)-[~]
└─$ sqlmap  -u https://www.rentalcars.com/api/experiments/v1/impressions -a -f  --level 5 --risk 2
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.8.3#stable}
|_ -| . [,]     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end us
er's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not
responsible for any misuse or damage caused by this program

[*] starting @ 03:21:57 /2024-05-01/

[03:21:57] [WARNING] you've provided target URL without any GET parameters (e.g. 'http://www.site.com/article.php?id=
1') and without providing any POST parameters through option '--data'
do you want to try URI injections in the target URL itself? [Y/n/q] Y
[03:21:59] [INFO] testing connection to the target URL
[03:22:00] [WARNING] the web server responded with an HTTP error code (405) which could interfere with the results of
 the tests
[03:22:00] [INFO] checking if the target is protected by some kind of WAF/IPS
[03:22:00] [CRITICAL] WAF/IPS identified as 'CloudFlare'
[03:22:00] [INFO] testing if the target URL content is stable
[03:22:00] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the page comparison on
 a sequence matcher. If no dynamic nor injectable parameters are detected, or in case of junk results, refer to user'
s manual paragraph 'Page comparison'
how do you want to proceed? [(C)ontinue/(s)tring/(r)egex/(q)uit] C
```



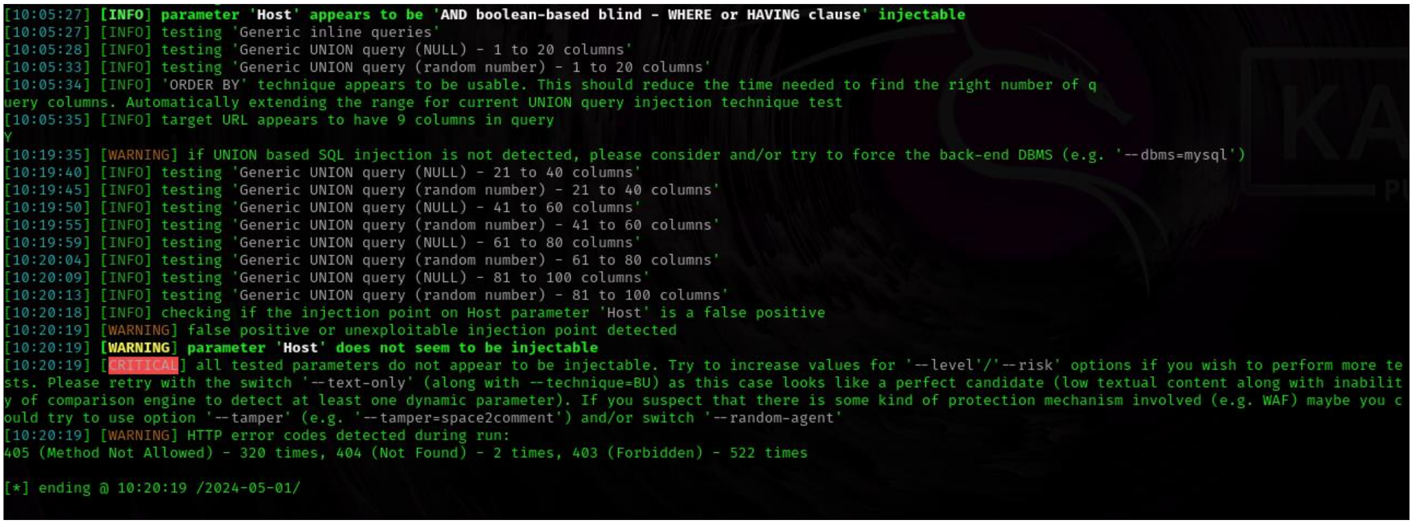
```
[03:22:06] [INFO] testing if URI parameter '#1*' is dynamic
[03:22:06] [WARNING] URI parameter '#1*' does not appear to be dynamic
[03:22:06] [WARNING] heuristic (basic) test shows that URI parameter '#1*' might not be injectable
[03:22:07] [INFO] testing for SQL injection on URI parameter '#1*'
[03:22:07] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[03:22:08] [INFO] URI parameter '#1*' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable
[03:22:08] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'Altibase'
it looks like the back-end DBMS is 'Altibase'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'Altibase' extending provided risk (2) value? [Y/n] Y
[03:22:17] [INFO] testing 'Generic inline queries'
[03:22:17] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[03:22:17] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one o
ther (potential) technique found
[03:22:18] [INFO] testing 'Generic UNION query (random number) - 1 to 20 columns'
[03:22:19] [INFO] testing 'Generic UNION query (NULL) - 21 to 40 columns'
[03:22:20] [INFO] testing 'Generic UNION query (random number) - 21 to 40 columns'
[03:22:21] [INFO] testing 'Generic UNION query (NULL) - 41 to 60 columns'
[03:22:22] [INFO] testing 'Generic UNION query (random number) - 41 to 60 columns'
[03:22:22] [INFO] testing 'Generic UNION query (NULL) - 61 to 80 columns'
[03:22:23] [INFO] testing 'Generic UNION query (random number) - 61 to 80 columns'
[03:22:24] [INFO] testing 'Generic UNION query (NULL) - 81 to 100 columns'
[03:22:26] [INFO] testing 'Generic UNION query (random number) - 81 to 100 columns'
[03:22:27] [INFO] checking if the injection point on URI parameter '#1*' is a false positive
[03:22:27] [WARNING] false positive or unexploitable injection point detected
[03:22:27] [WARNING] URI parameter '#1*' does not seem to be injectable
```

In the SQL map says that **AND Boolean-based blind -WHERE or HAVING Clause’** is injectable www.rentalcars.com/api/experiments/v1/impressions this is using the backend Altibase.

- Type: Enterprise-grade Open Source database.
- License: Free. As this is a free database, you do not need to purchase any license to use Altibase.
- Subscription: Subscription fees are lower than all mainstream DBMS providers.
- Industry: Enterprise Software
- Headquarters: The company manufacturing this product is known as 'Altibase'. It has two headquarters i.e. Greater New York City, Seoul, South Korea.
- Major Clients: Altibase has its customers in the Telecom, Financial Services, Manufacturing, and Utility Industry. Major clients include China Unicom, Posco, Samsung, HP, Hyundai, Toshiba Medical, and many other world-famous companies.
- Technical Support: 24/7/365 customer service is available globally.
- Scalability: Scales vertically and horizontally.
- User Size: This is suitable for all i.e. Small (<50 employees), Medium (50 to 1000 employees) and Big Enterprises (>1000 employees).

This database does have any direct vulnerability up to now. But there can be indirect vulnerabilities.

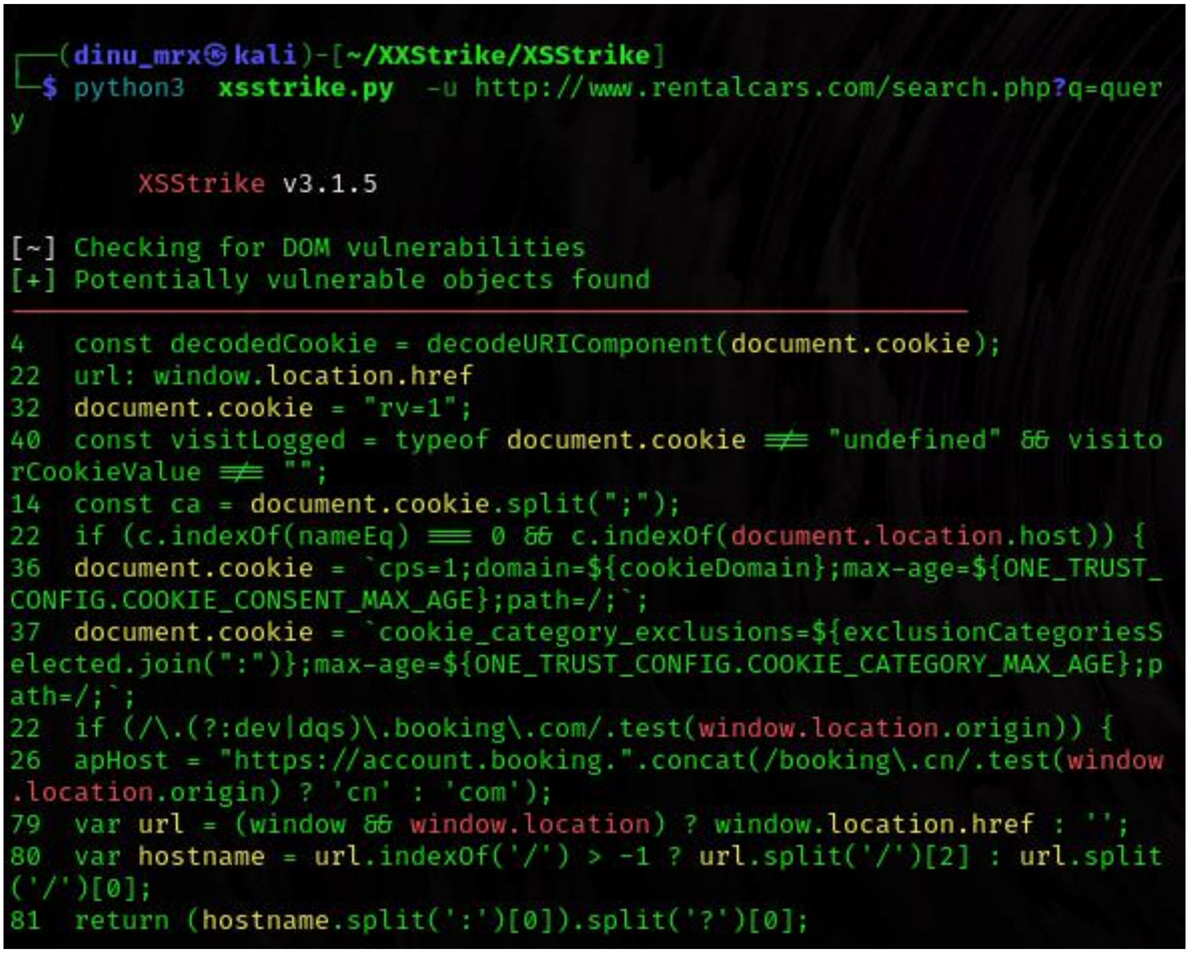
https://www.softwaretestinghelp.com/altibase-database-tutorial/



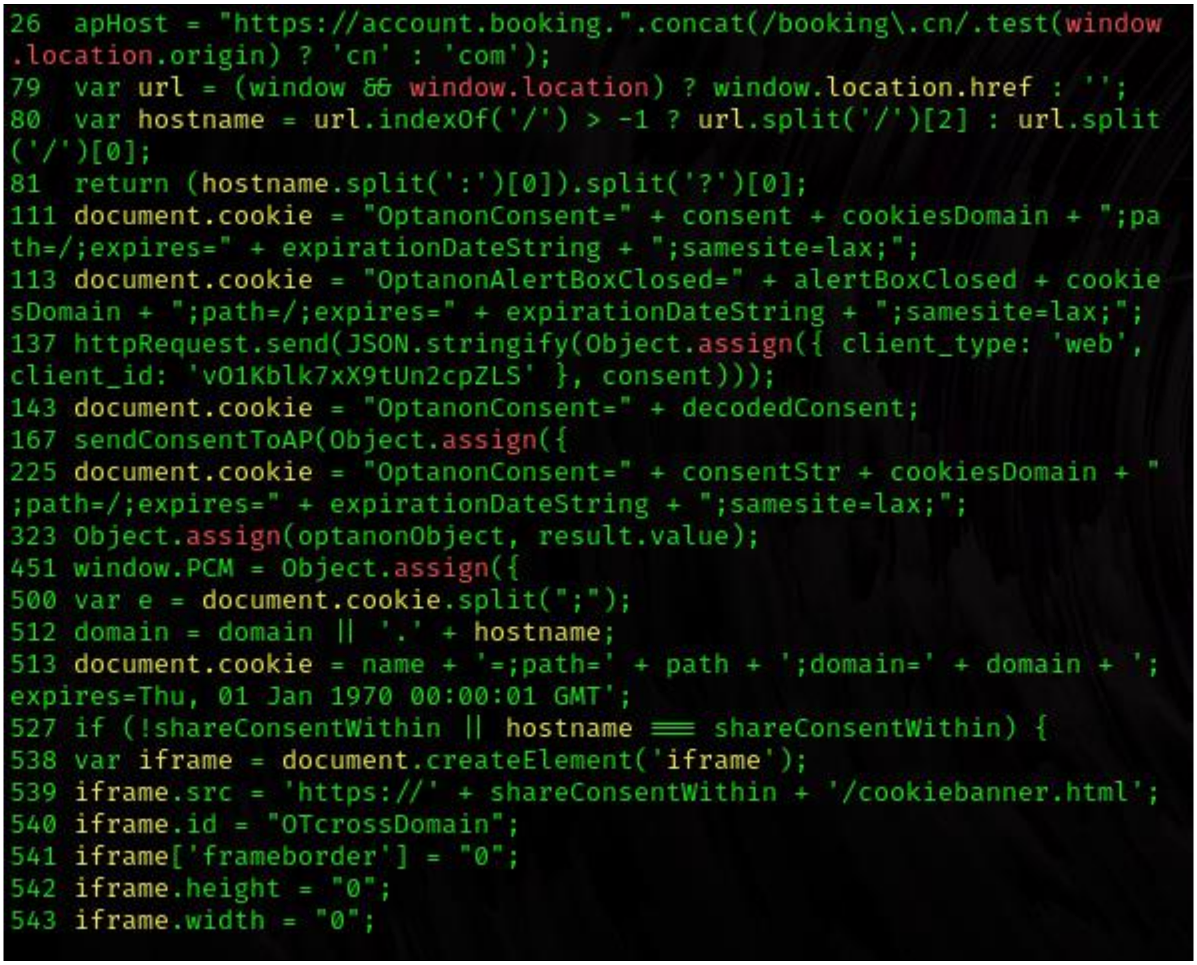
```
[10:05:27] [INFO] parameter 'Host' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable
[10:05:27] [INFO] testing 'Generic inline queries'
[10:05:28] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[10:05:33] [INFO] testing 'Generic UNION query (random number) - 1 to 20 columns'
[10:05:34] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of q
uery columns. Automatically extending the range for current UNION query injection technique test
[10:05:35] [INFO] target URL appears to have 9 columns in query
Y
[10:19:35] [WARNING] if UNION based SQL injection is not detected, please consider and/or try to force the back-end DBMS (e.g. '--dbms=mysql')
[10:19:40] [INFO] testing 'Generic UNION query (NULL) - 21 to 40 columns'
[10:19:45] [INFO] testing 'Generic UNION query (random number) - 21 to 40 columns'
[10:19:50] [INFO] testing 'Generic UNION query (NULL) - 41 to 60 columns'
[10:19:55] [INFO] testing 'Generic UNION query (random number) - 41 to 60 columns'
[10:19:59] [INFO] testing 'Generic UNION query (NULL) - 61 to 80 columns'
[10:20:04] [INFO] testing 'Generic UNION query (random number) - 61 to 80 columns'
[10:20:09] [INFO] testing 'Generic UNION query (NULL) - 81 to 100 columns'
[10:20:13] [INFO] testing 'Generic UNION query (random number) - 81 to 100 columns'
[10:20:18] [INFO] checking if the injection point on Host parameter 'Host' is a false positive
[10:20:19] [WARNING] false positive or unexploitable injection point detected
[10:20:19] [WARNING] parameter 'Host' does not seem to be injectable
[10:20:19] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more te
sts. Please retry with the switch '--text-only' (along with --technique=BU) as this case looks like a perfect candidate (low textual content along with inabilit
y of comparison engine to detect at least one dynamic parameter). If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you c
ould try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
[10:20:19] [WARNING] HTTP error codes detected during run:
405 (Method Not Allowed) - 320 times, 404 (Not Found) - 2 times, 403 (Forbidden) - 522 times

[*] ending @ 10:20:19 /2024-05-01/
```

The query has 9 columns that data found through this scan.

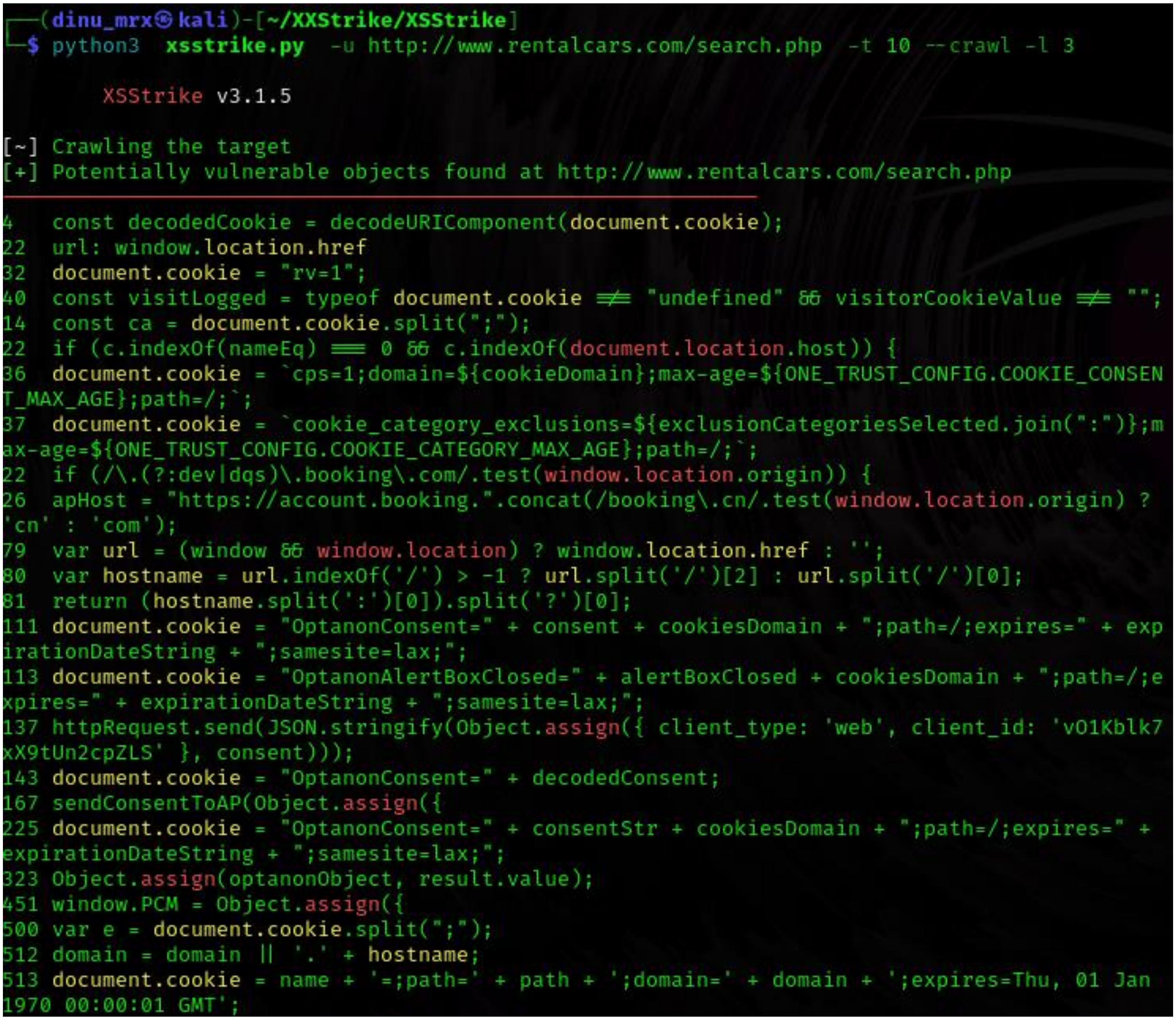
The results that were obtained from XSStrike



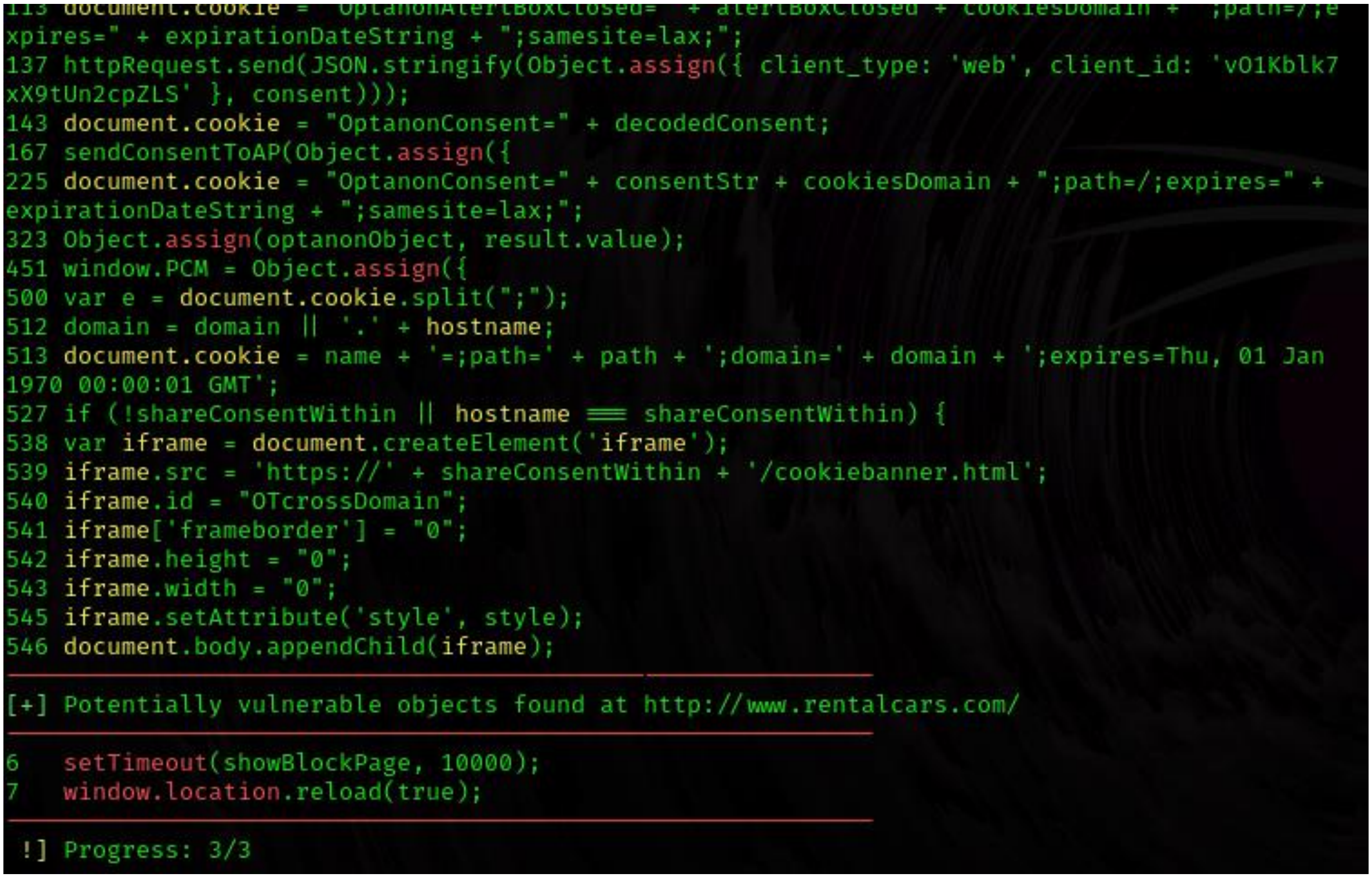
```
┌──(dinu_mrx㉿kali)-[~/XXStrike/XSStrike]
└─$ python3  xsstrike.py  -u http://www.rentalcars.com/search.php?q=quer
y

        XSStrike v3.1.5

[~] Checking for DOM vulnerabilities
[+] Potentially vulnerable objects found
────────────────────────────────────────────────────────────
4   const decodedCookie = decodeURIComponent(document.cookie);
22  url: window.location.href
32  document.cookie = "rv=1";
40  const visitLogged = typeof document.cookie !== "undefined" && visito
rCookieValue !== "";
14  const ca = document.cookie.split(";");
22  if (c.indexOf(nameEq) === 0 && c.indexOf(document.location.host)) {
36  document.cookie = `cps=1;domain=${cookieDomain};max-age=${ONE_TRUST_
CONFIG.COOKIE_CONSENT_MAX_AGE};path=/;`;
37  document.cookie = `cookie_category_exclusions=${exclusionCategoriesS
elected.join(":")};max-age=${ONE_TRUST_CONFIG.COOKIE_CATEGORY_MAX_AGE};p
ath=/;`;
22  if (/\.(?:dev|dqs)\.booking\.com/.test(window.location.origin)) {
26  apHost = "https://account.booking.".concat(/booking\.cn/.test(window
.location.origin) ? 'cn' : 'com');
79  var url = (window && window.location) ? window.location.href : '';
80  var hostname = url.indexOf('/') > -1 ? url.split('/')[2] : url.split
('/')[0];
81  return (hostname.split(':')[0]).split('?')[0];
```



```
26  apHost = "https://account.booking.".concat(/booking\.cn/.test(window.location.origin) ? 'cn' : 'com');
79  var url = (window && window.location) ? window.location.href : '';
80  var hostname = url.indexOf('/') > -1 ? url.split('/')[2] : url.split('/')[0];
81  return (hostname.split(':')[0]).split('?')[0];
111 document.cookie = "OptanonConsent=" + consent + cookiesDomain + ";path=/;expires=" + expirationDateString + ";samesite=lax;";
113 document.cookie = "OptanonAlertBoxClosed=" + alertBoxClosed + cookiesDomain + ";path=/;expires=" + expirationDateString + ";samesite=lax;";
137 httpRequest.send(JSON.stringify(Object.assign({ client_type: 'web', client_id: 'vO1Kblk7xX9tUn2cpZLS' }, consent)));
143 document.cookie = "OptanonConsent=" + decodedConsent;
167 sendConsentToAP(Object.assign({
225 document.cookie = "OptanonConsent=" + consentStr + cookiesDomain + ";path=/;expires=" + expirationDateString + ";samesite=lax;";
323 Object.assign(optanonObject, result.value);
451 window.PCM = Object.assign({
500 var e = document.cookie.split(";");
512 domain = domain || '.' + hostname;
513 document.cookie = name + '=;path=' + path + ';domain=' + domain + ';expires=Thu, 01 Jan 1970 00:00:01 GMT';
527 if (!shareConsentWithin || hostname === shareConsentWithin) {
538 var iframe = document.createElement('iframe');
539 iframe.src = 'https://' + shareConsentWithin + '/cookiebanner.html';
540 iframe.id = "OTcrossDomain";
541 iframe['frameborder'] = "0";
542 iframe.height = "0";
543 iframe.width = "0";
```



```
┌──(dinu_mrx㉿kali)-[~/XXStrike/XSStrike]
└─$ python3  xsstrike.py  -u http://www.rentalcars.com/search.php  -t 10 --crawl -l 3

        XSStrike v3.1.5

[~] Crawling the target
[+] Potentially vulnerable objects found at http://www.rentalcars.com/search.php
------------------------------------------------------------------------------------
4   const decodedCookie = decodeURIComponent(document.cookie);
22  url: window.location.href
32  document.cookie = "rv=1";
40  const visitLogged = typeof document.cookie !== "undefined" && visitorCookieValue !== "";
14  const ca = document.cookie.split(";");
22  if (c.indexOf(nameEq) === 0 && c.indexOf(document.location.host)) {
36  document.cookie = `cps=1;domain=${cookieDomain};max-age=${ONE_TRUST_CONFIG.COOKIE_CONSEN
T_MAX_AGE};path=/;`;
37  document.cookie = `cookie_category_exclusions=${exclusionCategoriesSelected.join(":")};m
ax-age=${ONE_TRUST_CONFIG.COOKIE_CATEGORY_MAX_AGE};path=/;`;
22  if (/\.(?:dev|dqs)\.booking\.com/.test(window.location.origin)) {
26  apHost = "https://account.booking.".concat(/booking\.cn/.test(window.location.origin) ?
'cn' : 'com');
79  var url = (window && window.location) ? window.location.href : '';
80  var hostname = url.indexOf('/') > -1 ? url.split('/')[2] : url.split('/')[0];
81  return (hostname.split(':')[0]).split('?')[0];
111 document.cookie = "OptanonConsent=" + consent + cookiesDomain + ";path=/;expires=" + exp
irationDateString + ";samesite=lax;";
113 document.cookie = "OptanonAlertBoxClosed=" + alertBoxClosed + cookiesDomain + ";path=/;e
xpires=" + expirationDateString + ";samesite=lax;";
137 httpRequest.send(JSON.stringify(Object.assign({ client_type: 'web', client_id: 'vO1Kblk7
xX9tUn2cpZLS' }, consent)));
143 document.cookie = "OptanonConsent=" + decodedConsent;
167 sendConsentToAP(Object.assign({
225 document.cookie = "OptanonConsent=" + consentStr + cookiesDomain + ";path=/;expires=" +
expirationDateString + ";samesite=lax;";
323 Object.assign(optanonObject, result.value);
451 window.PCM = Object.assign({
500 var e = document.cookie.split(";");
512 domain = domain || '.' + hostname;
513 document.cookie = name + '=;path=' + path + ';domain=' + domain + ';expires=Thu, 01 Jan
1970 00:00:01 GMT';
```



```
113 document.cookie = "OptanonAlertBoxClosed=" + alertBoxClosed + cookiesDomain + ";path=/;e
xpires=" + expirationDateString + ";samesite=lax;";
137 httpRequest.send(JSON.stringify(Object.assign({ client_type: 'web', client_id: 'vO1Kblk7
xX9tUn2cpZLS' }, consent)));
143 document.cookie = "OptanonConsent=" + decodedConsent;
167 sendConsentToAP(Object.assign({
225 document.cookie = "OptanonConsent=" + consentStr + cookiesDomain + ";path=/;expires=" +
expirationDateString + ";samesite=lax;";
323 Object.assign(optanonObject, result.value);
451 window.PCM = Object.assign({
500 var e = document.cookie.split(";");
512 domain = domain || '.' + hostname;
513 document.cookie = name + '=;path=' + path + ';domain=' + domain + ';expires=Thu, 01 Jan
1970 00:00:01 GMT';
527 if (!shareConsentWithin || hostname === shareConsentWithin) {
538 var iframe = document.createElement('iframe');
539 iframe.src = 'https://' + shareConsentWithin + '/cookiebanner.html';
540 iframe.id = "OTcrossDomain";
541 iframe['frameborder'] = "0";
542 iframe.height = "0";
543 iframe.width = "0";
545 iframe.setAttribute('style', style);
546 document.body.appendChild(iframe);
------------------------------------------------------------------------
[+] Potentially vulnerable objects found at http://www.rentalcars.com/
------------------------------------------------------------------------
6   setTimeout(showBlockPage, 10000);
7   window.location.reload(true);
------------------------------------------------------------------------
 !] Progress: 3/3
```