**BUB BOUNTY**

A blue and orange logo

Description automatically generated

**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**

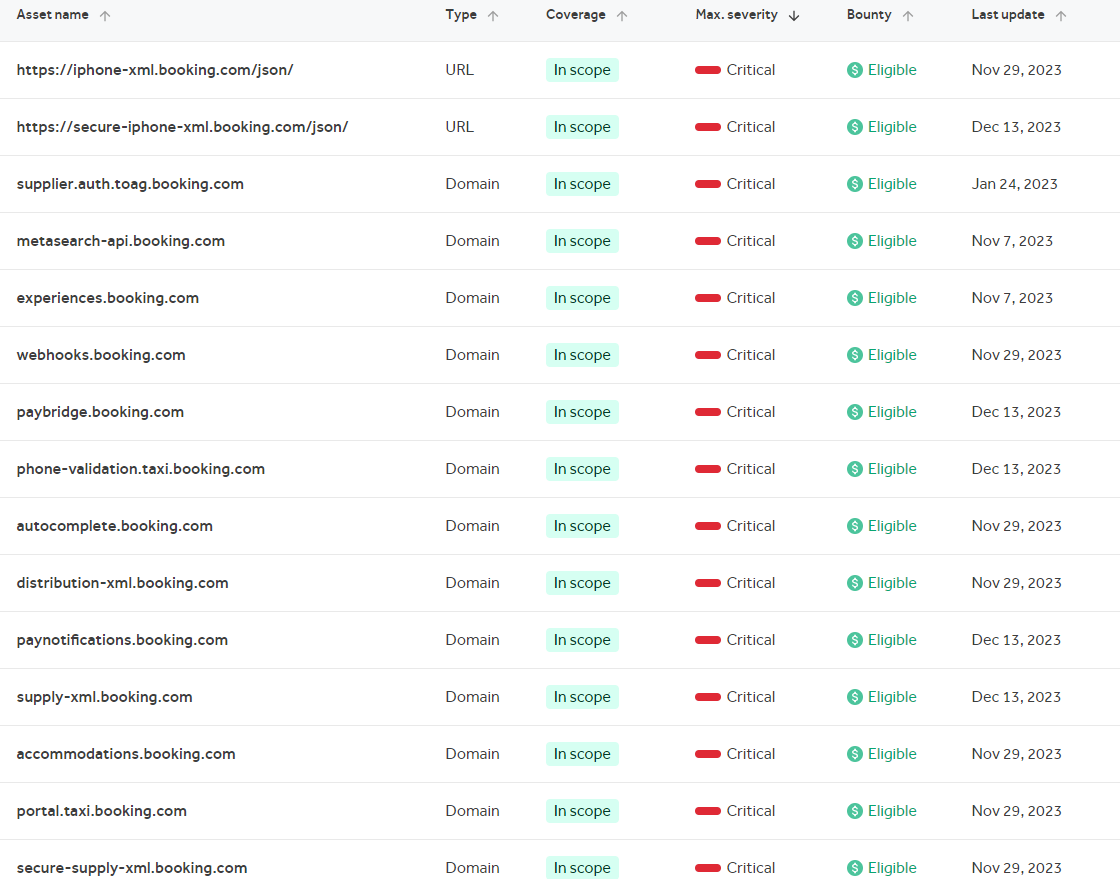
**Accepted, in-scope vulnerabilities include, but are not limited to:**

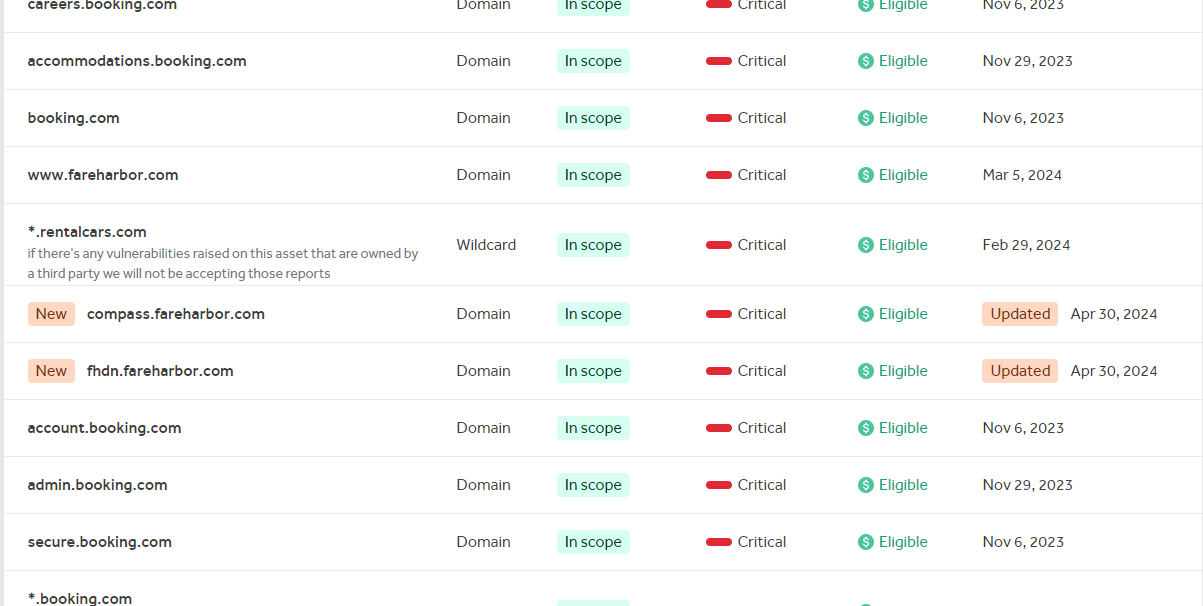
* Disclosure of sensitive or personally identifiable information
* Cross-Site Scripting (XSS) - Please note, for XSS if the same issue is reported for the different subdomains but with the same root cause, it will be considered duplicate
* Cross-Site Request Forgery (CSRF) for sensitive functions in 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

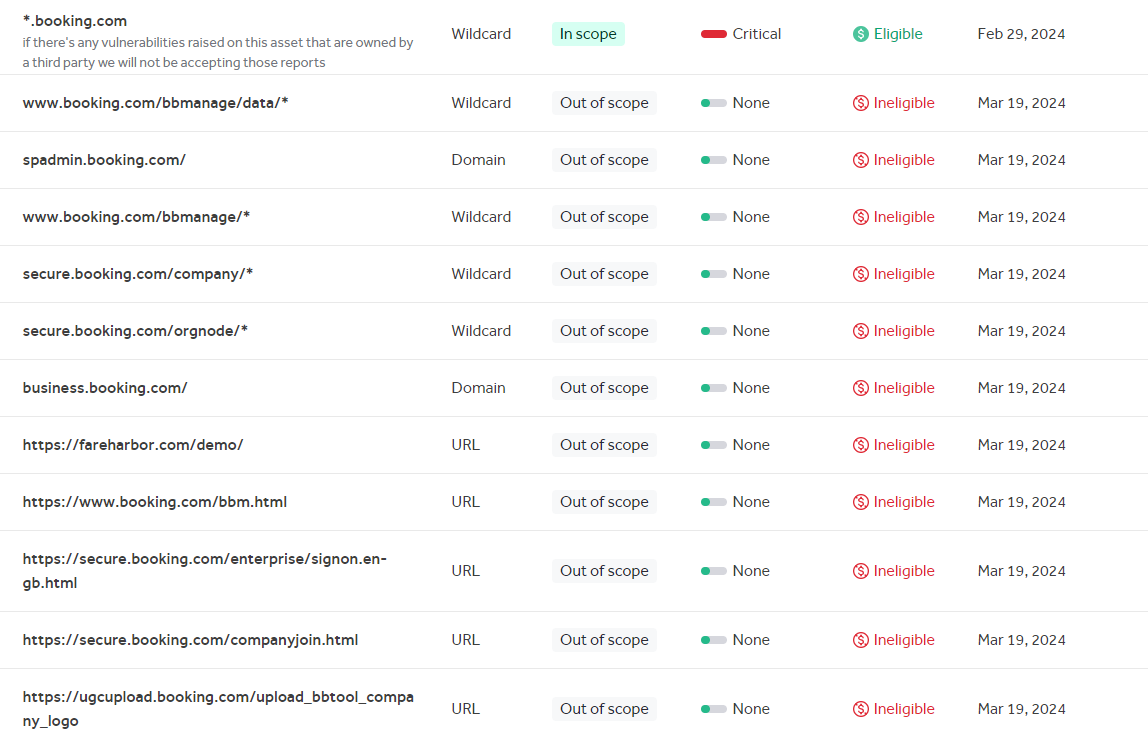
**Out-Of-Scope Vulnerabilities** Depending on their impact, not all reported issues may qualify for a monetary reward. However, all reports are reviewed on a case-by-case basis and any report that results in a change being made will at a minimum receive recognition. Please note that our **program terms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

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

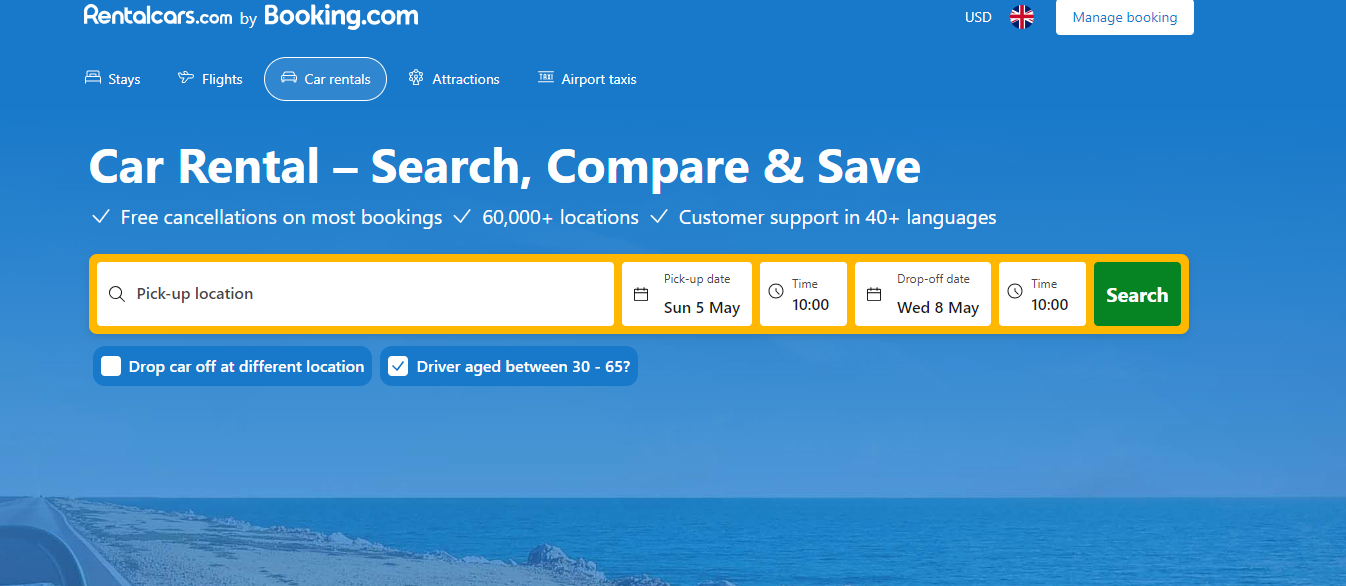
* Any vulnerability which requires access to a compromised email account or Booking.com account for successful exploitation
* Vulnerabilities on Third Party Products
* Attacks requiring physical access to a user's device or network.
* Forms missing CSRF tokens (we require evidence of actual CSRF vulnerability)
* 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.)

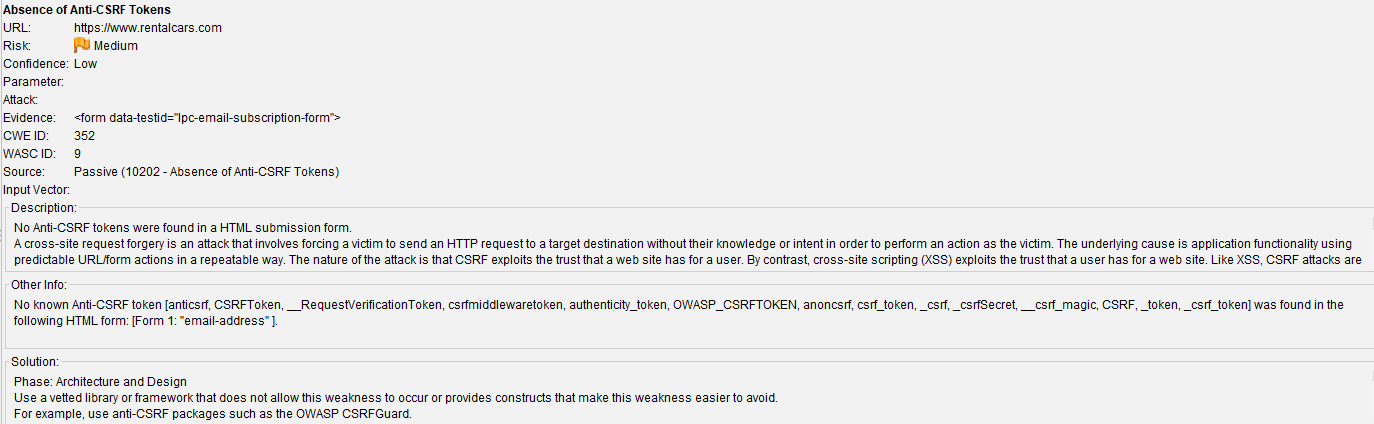




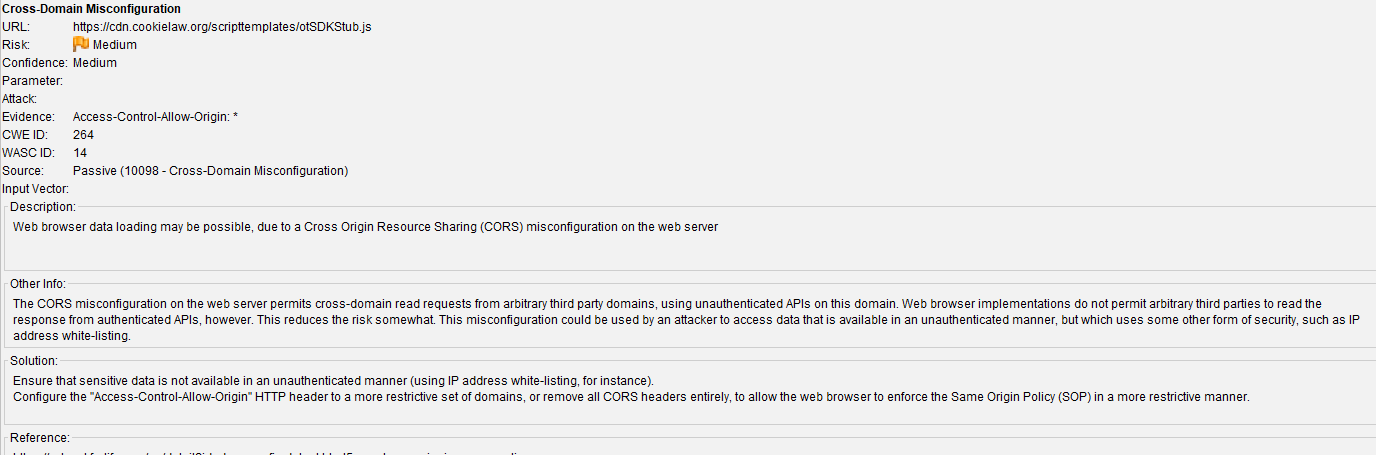


[www.rentalcars.com](http://www.rentalcars.com) (\*rentalcars.com )

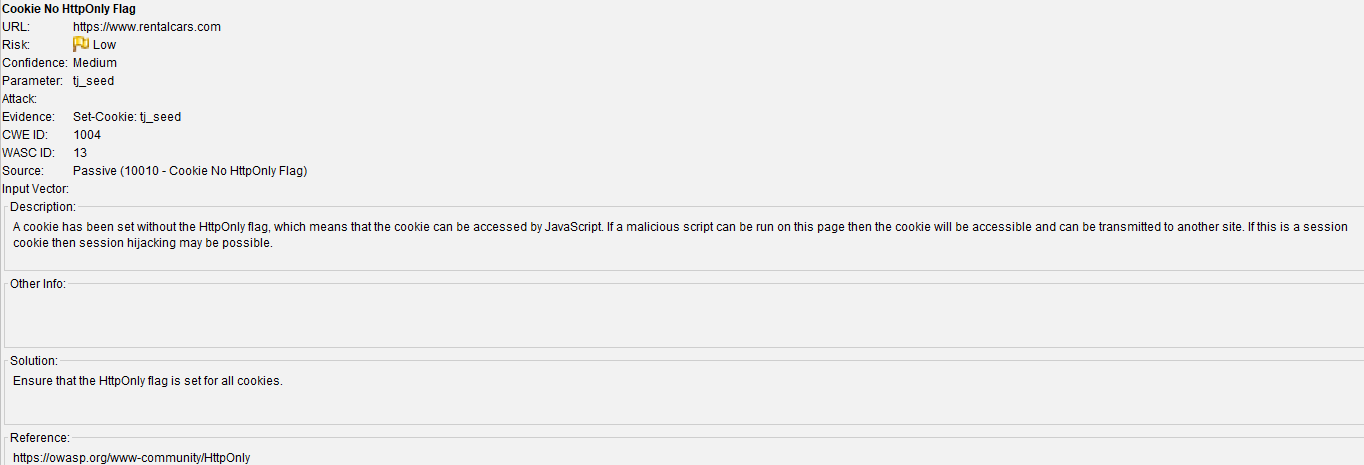


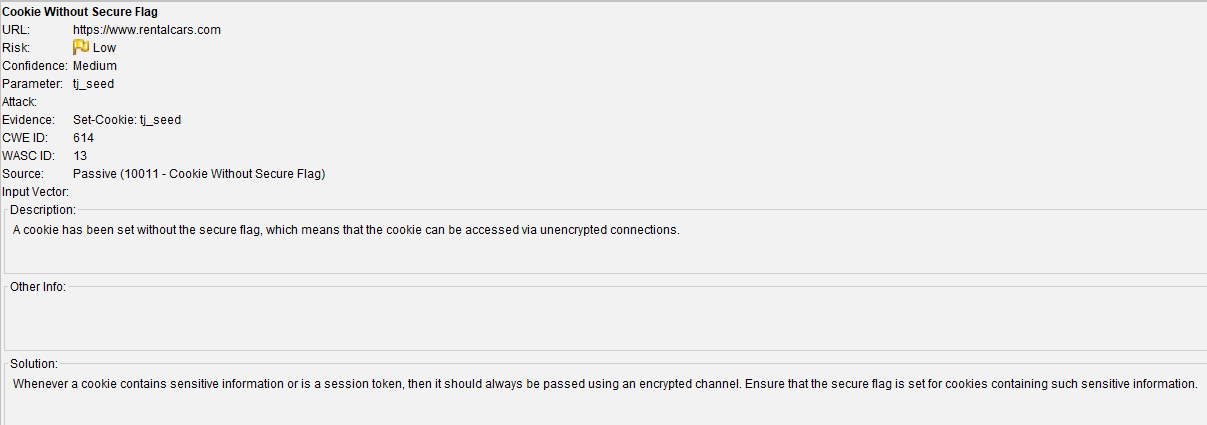
The ZAP automated scan gives the following results



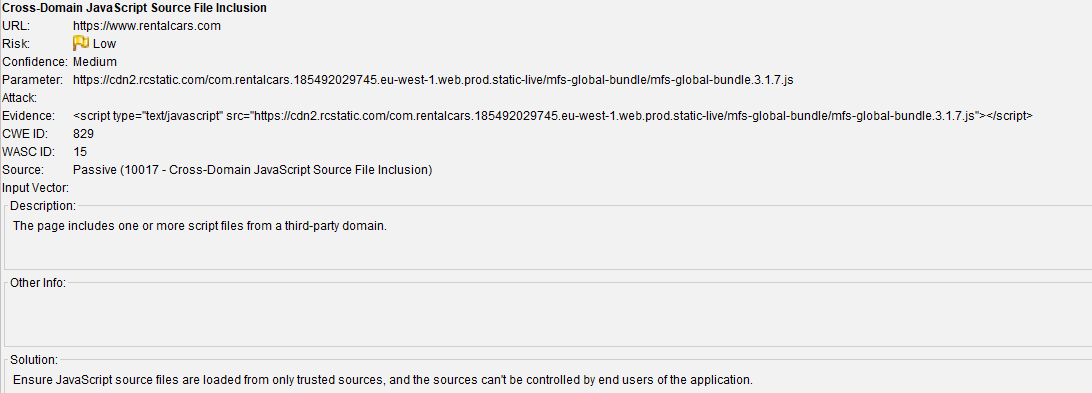


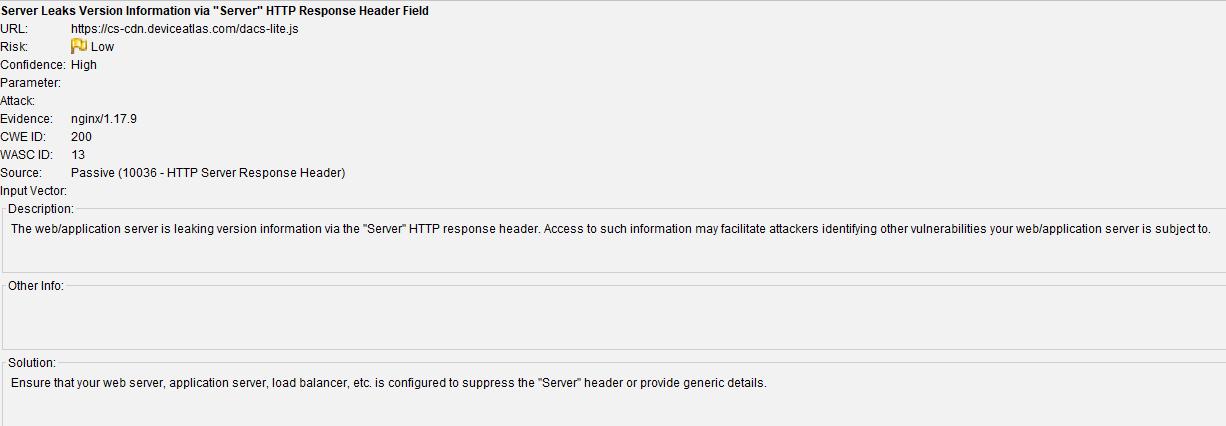


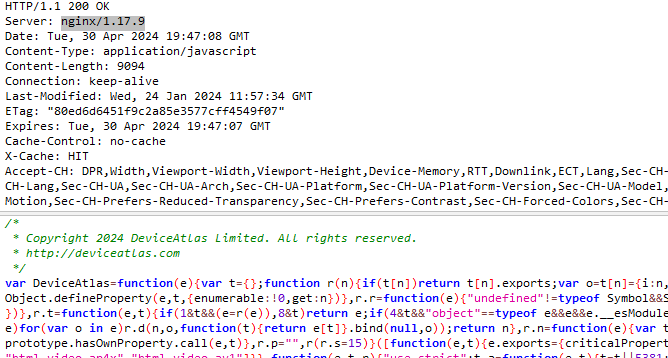












According to the HTTP response header the server is nginx/1.17.9

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>

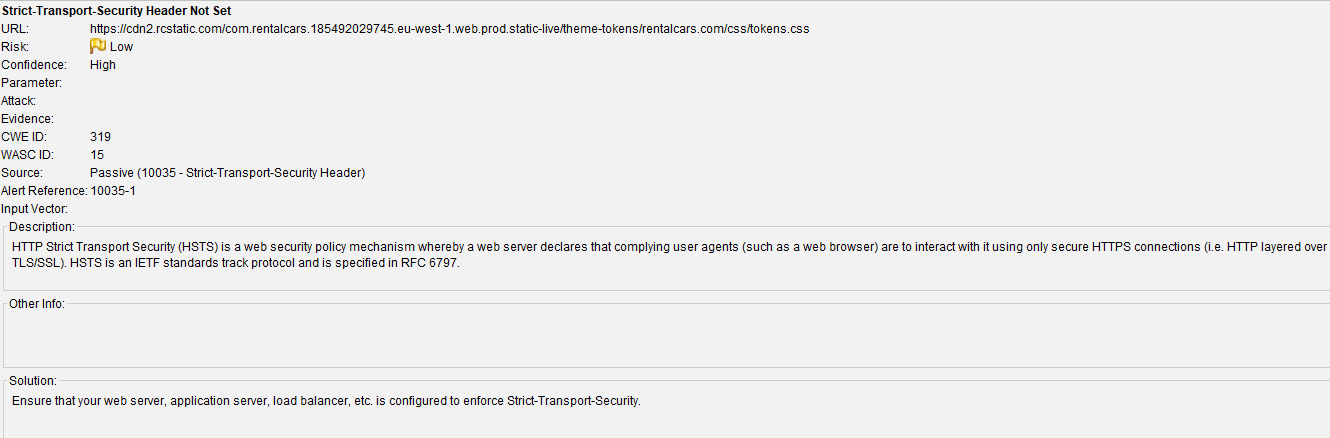
<https://www.cybersecurity-help.cz/vdb/SB2022101941>

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>



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](https://kinsta.com/blog/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](https://kinsta.com/knowledgebase/redirect-http-to-https/). 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](https://www.encryptionconsulting.com/detailed-guide-to-preventing-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)](https://kinsta.com/knowledgebase/hsts-strict-transport-security/). 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**

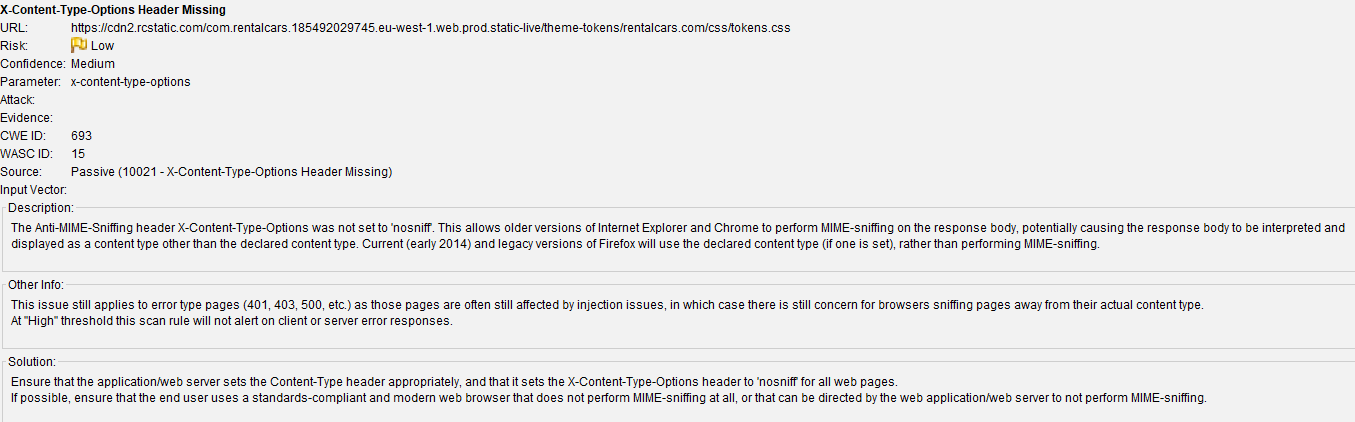
*max-age* is the time in seconds indicating how long the browser should remember that a site is accessible via HTTPS only. The time is refreshed (set again to max-age) after each request to the domain. In the example the time is equal to one year.

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

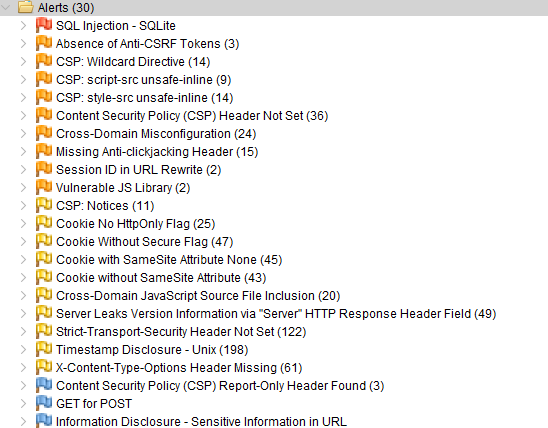
*preload* is optional (and not the part of the official specification) and allows you to add your website to a preload list maintained by Google. This means that your domain will be hardcoded in the list and browsers will never try to connect using an insecure connection. Note that it will have permanent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

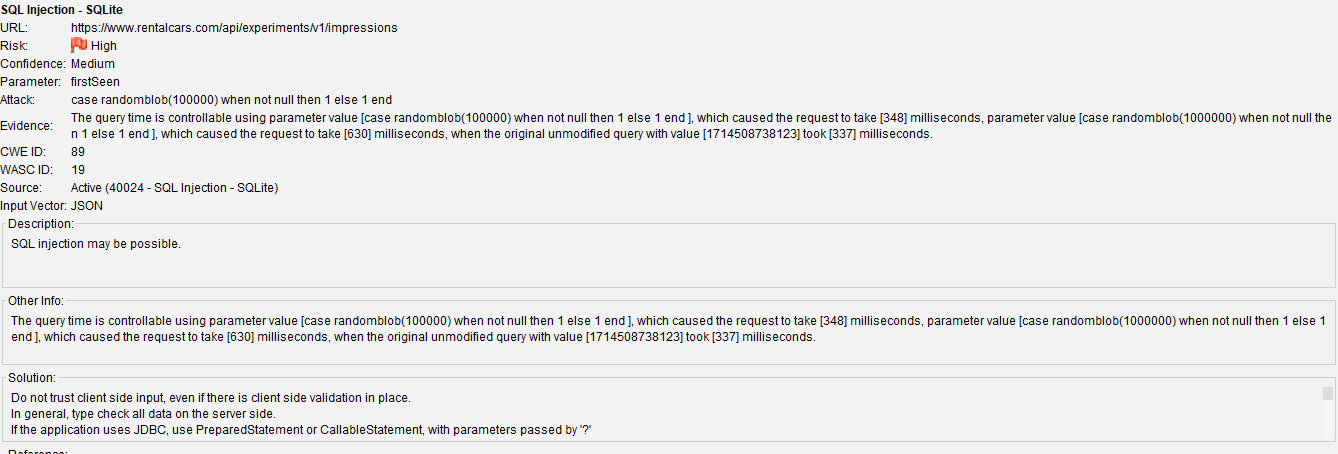
<https://scanrepeat.com/web-security-knowledge-base/strict-transport-security-header-not-set>



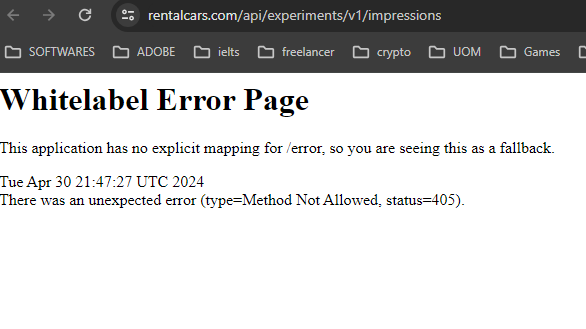


Once the automated scanner is finished then I moved into manual explore tab and check the site manually the got 30 alerts





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



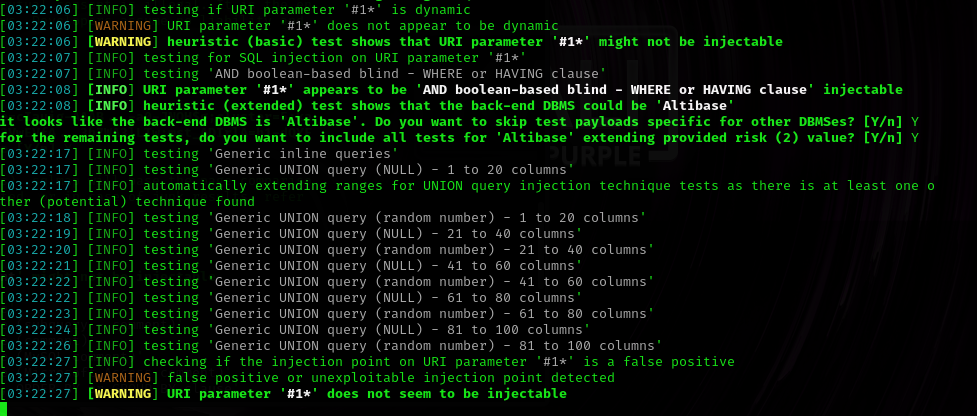


Other errors are quite similar to previous scan.

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.

A screenshot of a computer

Description automatically generated



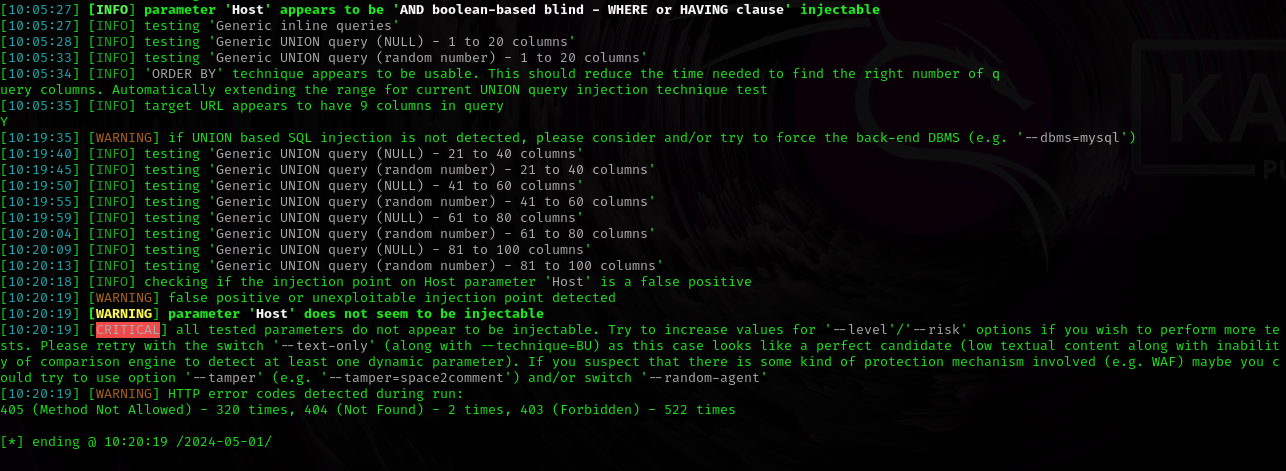
In the SQL map says that  **AND Boolean-based blind -WHERE or HAVING Clause’** is injectable

[www.rentalcars.com/api/experiments/v1/impressions](http://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/>



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

The results that were obtained from XSStrike

