**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**

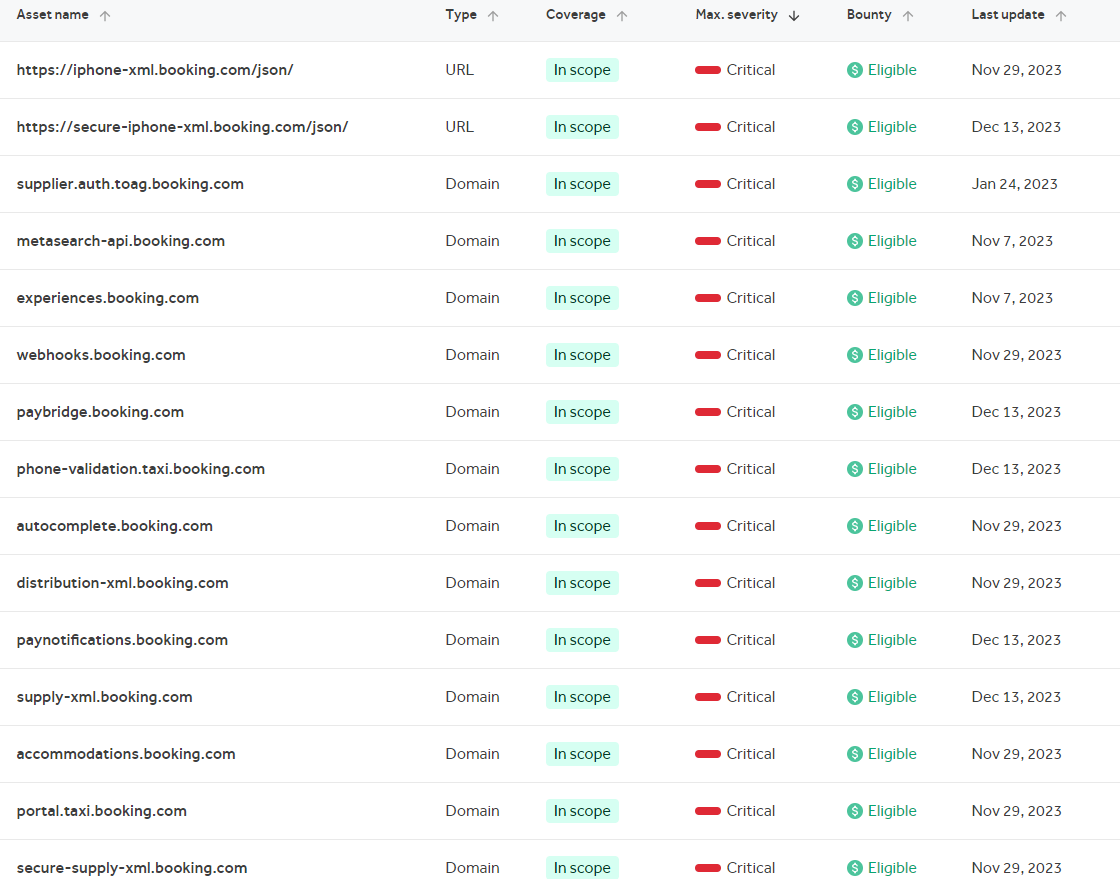
**Accepted, in-scope vulnerabilities include, but are not limited to:**

* 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.

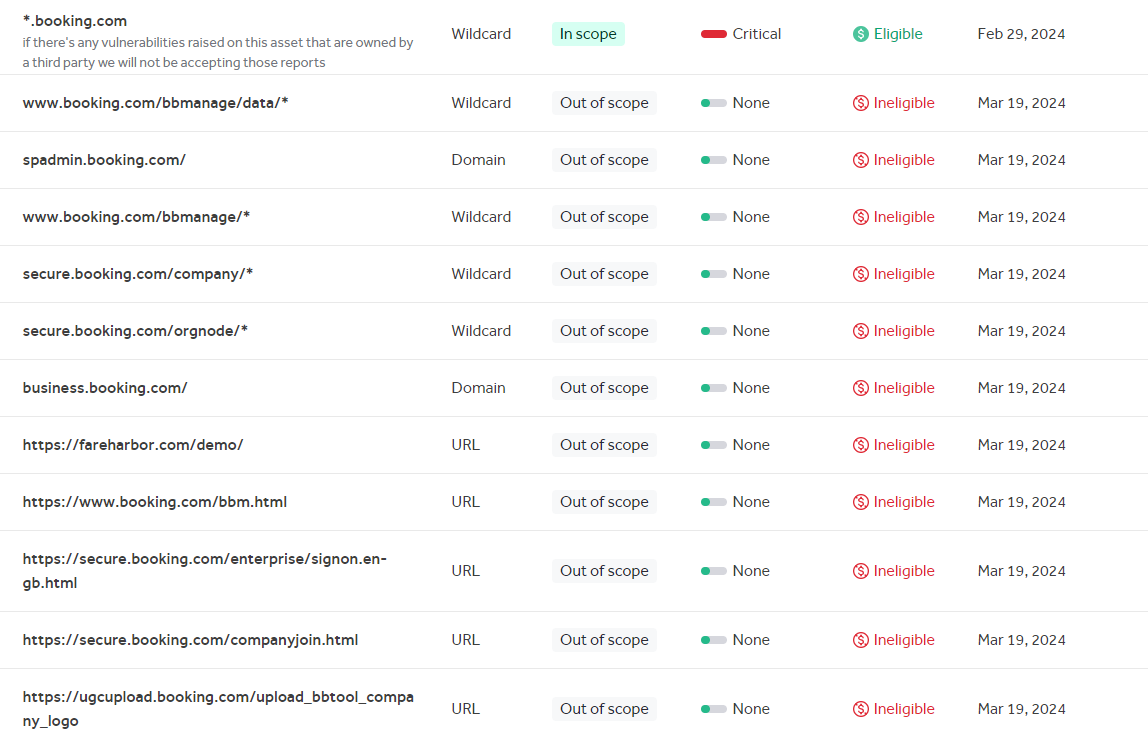
**The following issues are outside the scope of our vulnerability rewards program:**

* 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.)



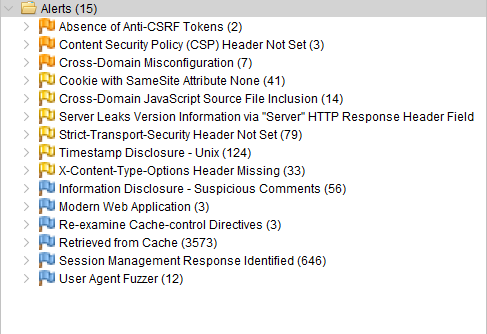
A screenshot of a computer

Description automatically generated

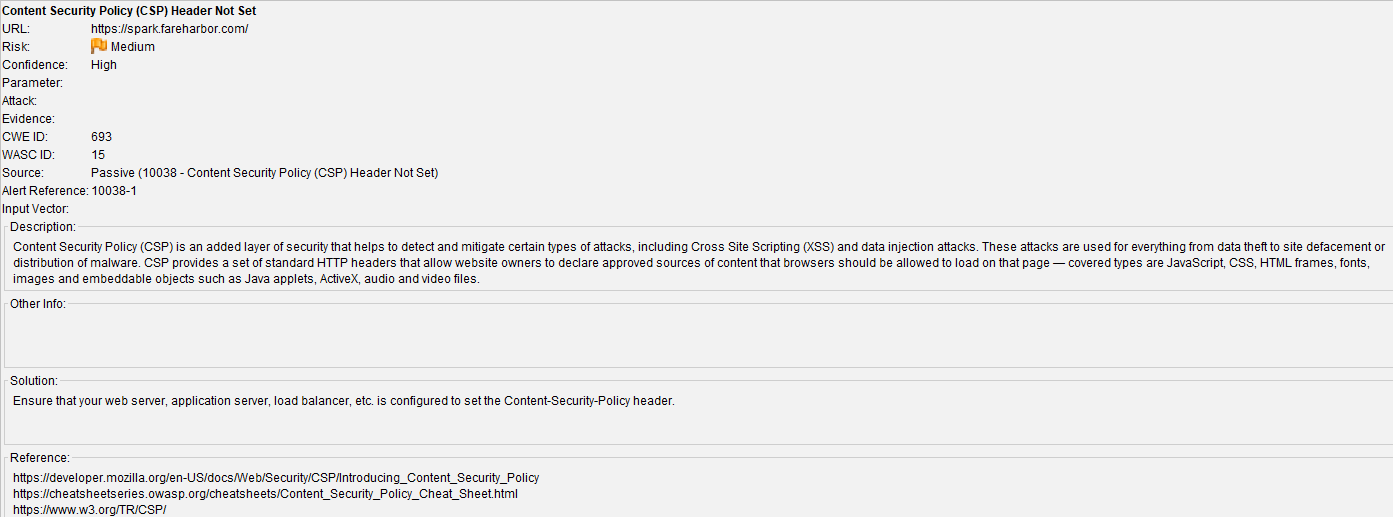


<https://spark.fareharbor.com/>

The results that were obtained from the OWSAP ZAP automated scan







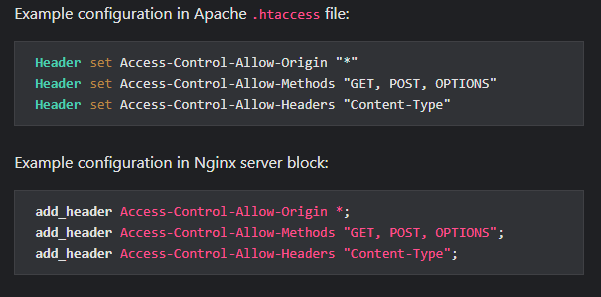




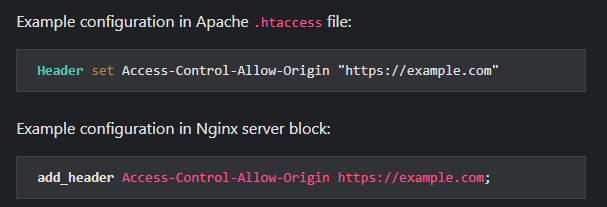
**Remediation**

To remediate the vulnerability of Cross-Domain Misconfiguration, the following steps can be taken:

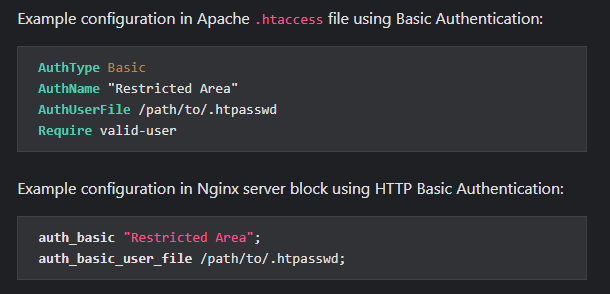
1. **Configure CORS properly:** Ensure that the Cross-Origin Resource Sharing (CORS) policy is correctly configured on the web server. This involves specifying the allowed origins, methods, headers, and credentials for cross-domain requests.



**Limit CORS to necessary domains:** Restrict the allowed origins to only the domains that require access. This helps prevent unauthorized cross-domain requests.

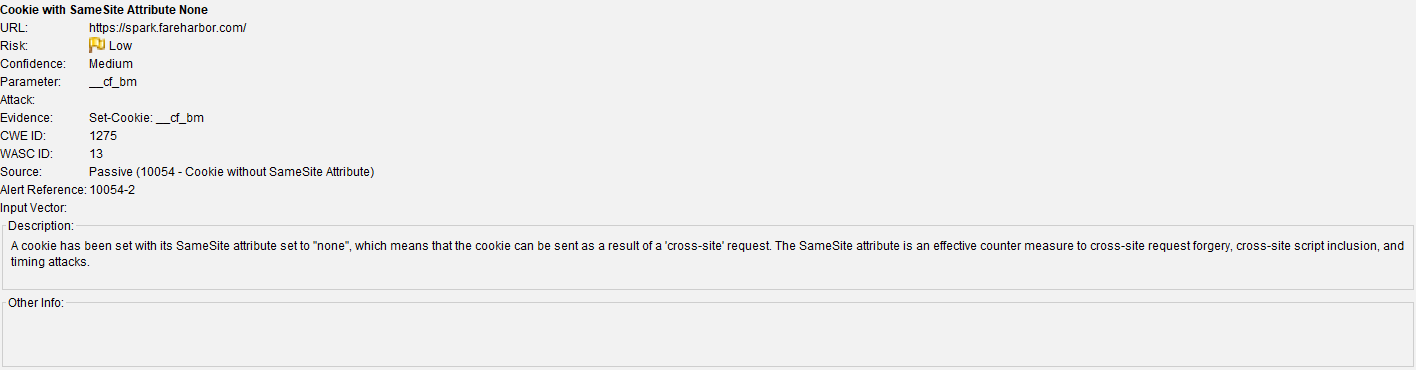


**Implement authentication and authorization:** Require authentication and authorization for sensitive resources to further control access to cross-domain requests.

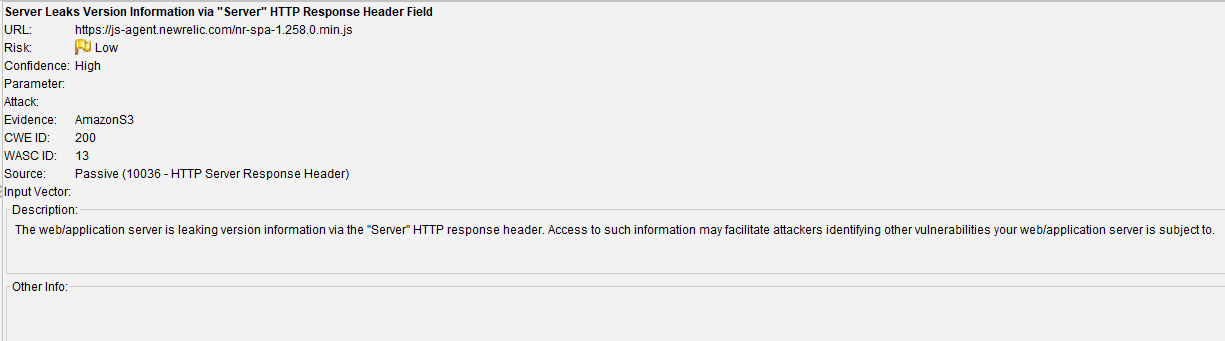


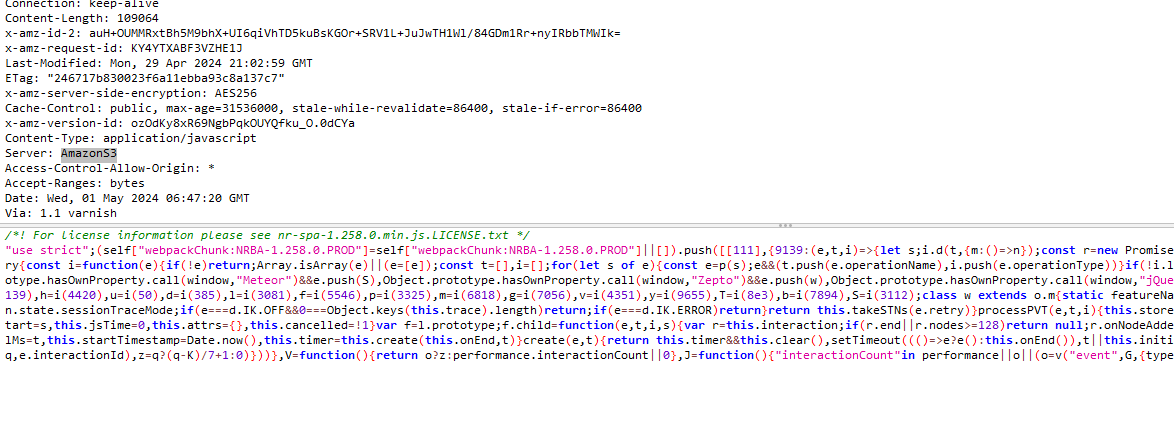
The risks associated with Cross-Domain Misconfiguration include:

* **Data leakage:** Attackers can exploit the misconfiguration to access sensitive data from other domains, potentially leading to data breaches and privacy violations.
* **Cross-Site Request Forgery (CSRF):** Misconfigured CORS can enable CSRF attacks, where an attacker tricks a user into performing unintended actions on a trusted website by leveraging the victim’s authenticated session.
* **Unauthorized access:** By bypassing the same-origin policy, attackers can perform actions on behalf of the user, leading to unauthorized access to resources and potential account compromise.
* **Malicious code execution:** If an attacker can inject malicious code into a vulnerable website, they can execute arbitrary scripts in the victim’s browser, leading to further exploitation and compromise.









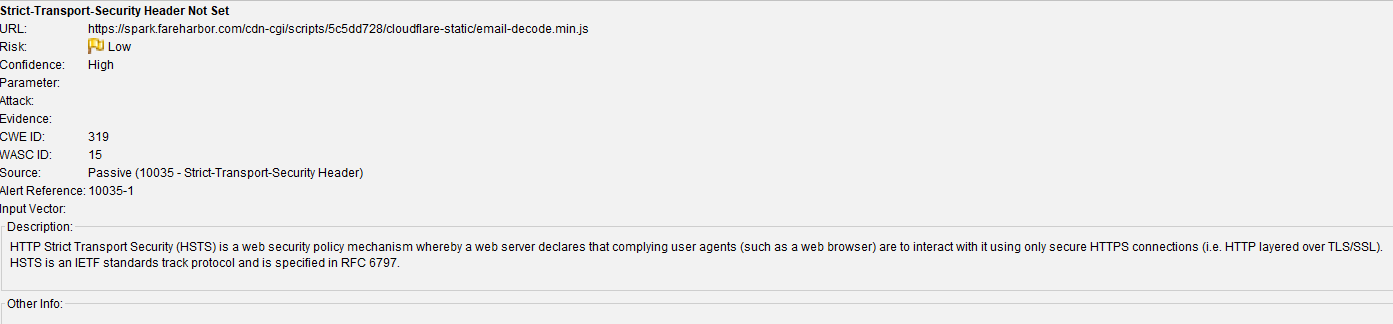
**Description**

Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. Files within S3 are organized into "buckets", which are named logical containers accessible at a predictable URL. Access controls can be applied to both the bucket itself and to individual objects (files and directories) stored within that bucket. A bucket is considered public if any user can list the contents of the bucket, and private if the bucket's contents can only be listed or written by certain S3 users.  
  
This web application is using a public Amazon S3 bucket. This is not recommended, as a public bucket will list all of its files and directories to an any user that asks.

**Remediation**

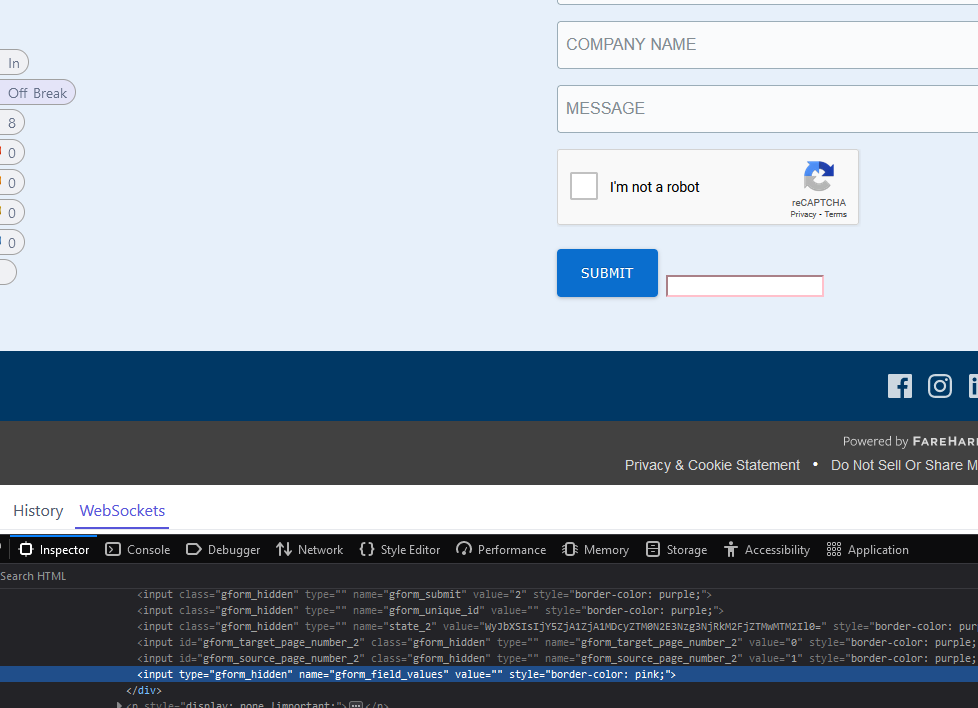
Make sure all the Amazon S3 buckets you are using are marked as private.

<https://www.acunetix.com/vulnerabilities/web/amazon-s3-public-bucket/>





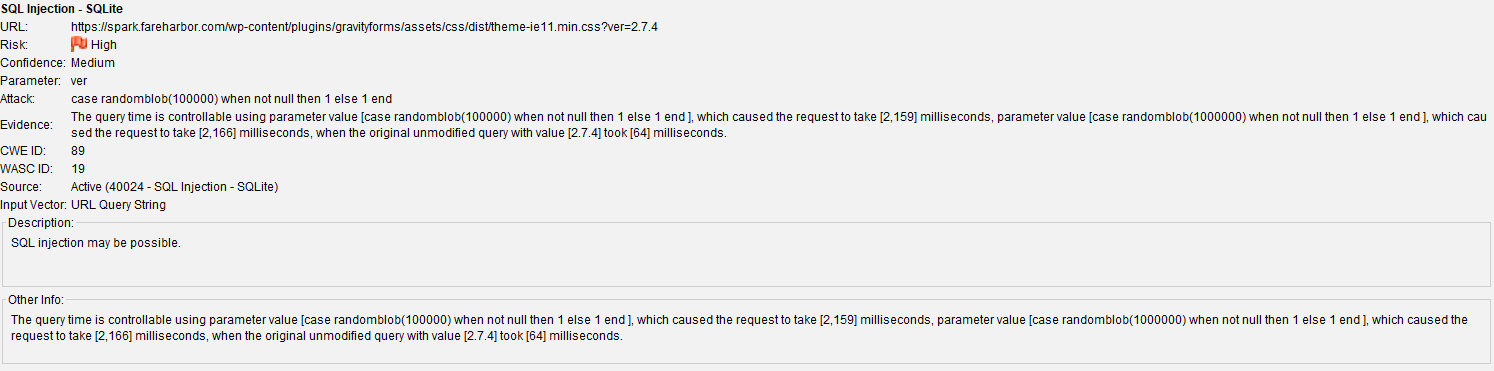
In this site there are 8 hidden fields

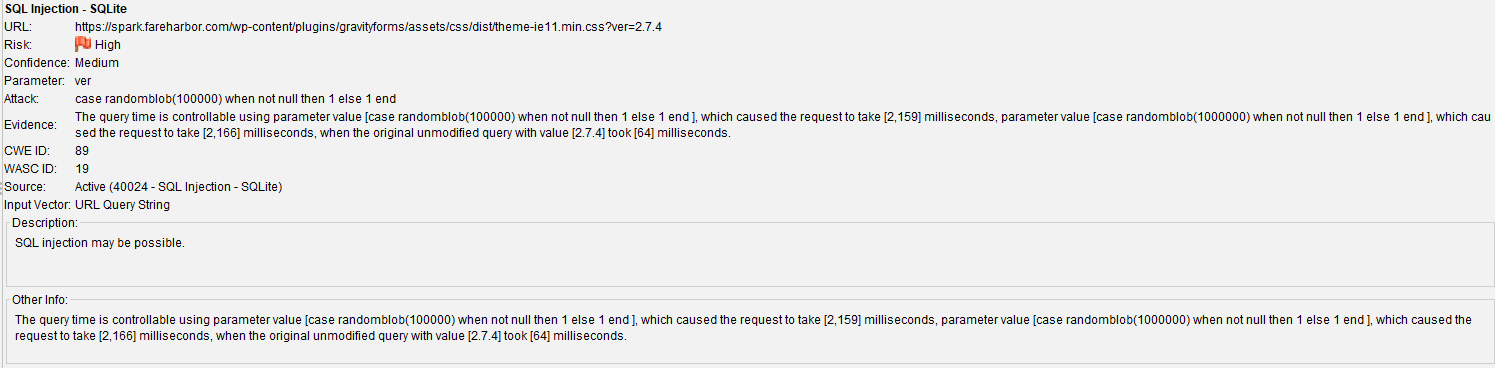


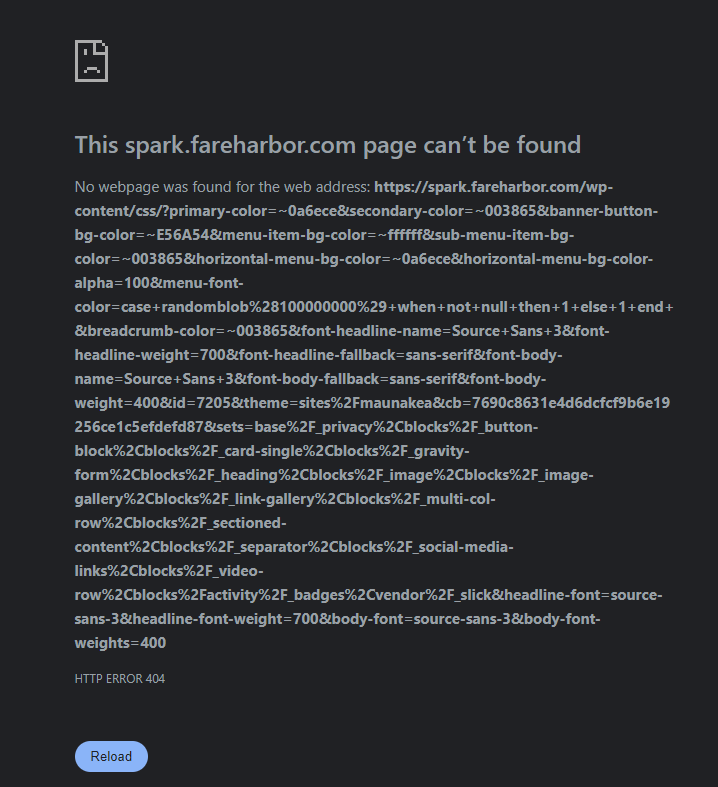
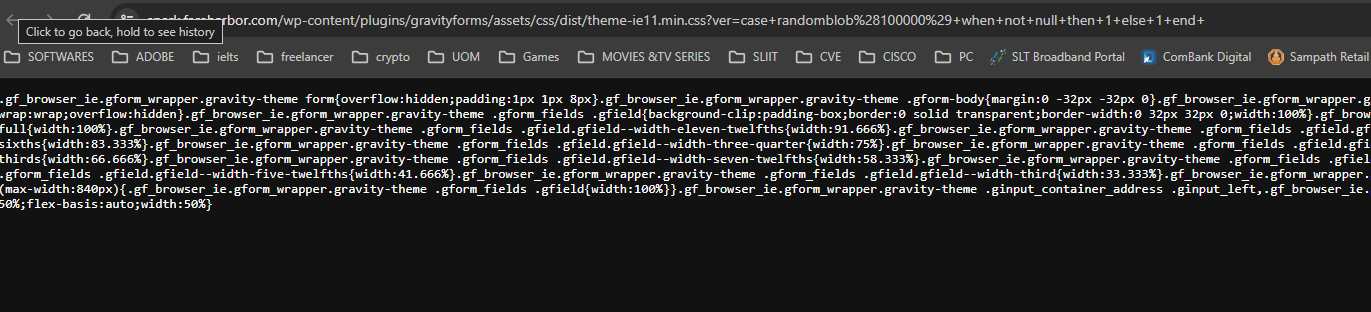
Results from the manual scan

A screenshot of a computer

Description automatically generated





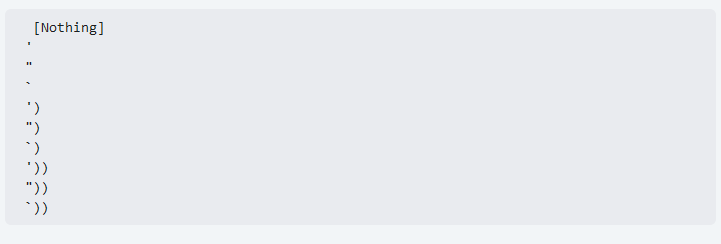


**What is SQL injection?**

An **SQL injection** is a security flaw that allows attackers to **interfere with database queries** of an application. This vulnerability can enable attackers to **view**, **modify**, or **delete** data they shouldn't access, including information of other users or any data the application can access. Such actions may result in permanent changes to the application's functionality or content or even compromision of the server or denial of service.

**Entry point detection**

When a site appears to be **vulnerable to SQL injection (SQLi)** due to unusual server responses to SQLi-related inputs, the **first step** is to understand how to **inject data into the query without disrupting it**. This requires identifying the method to **escape from the current context** effectively.



Then, you need to know how to **fix the query so there isn't errors**. In order to fix the query you can **input** data so the **previous query accept the new data**, or you can just **input** your data and **add a comment symbol add the end**.

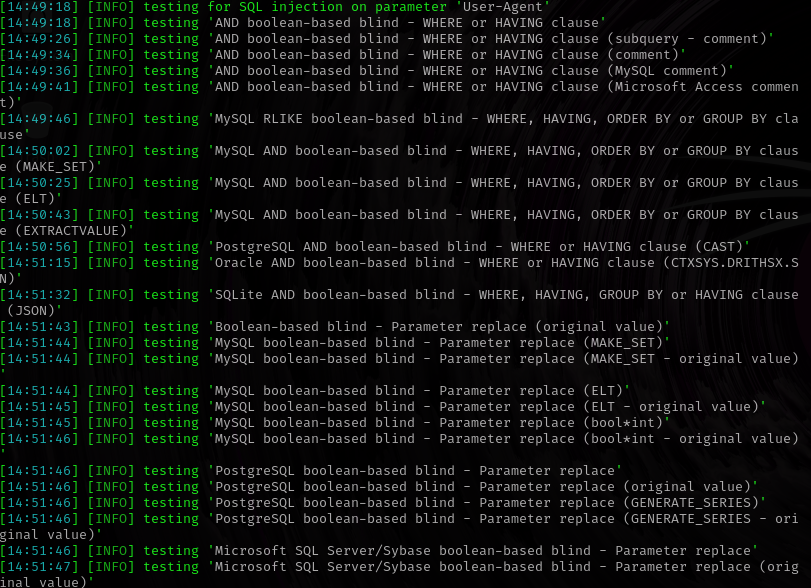
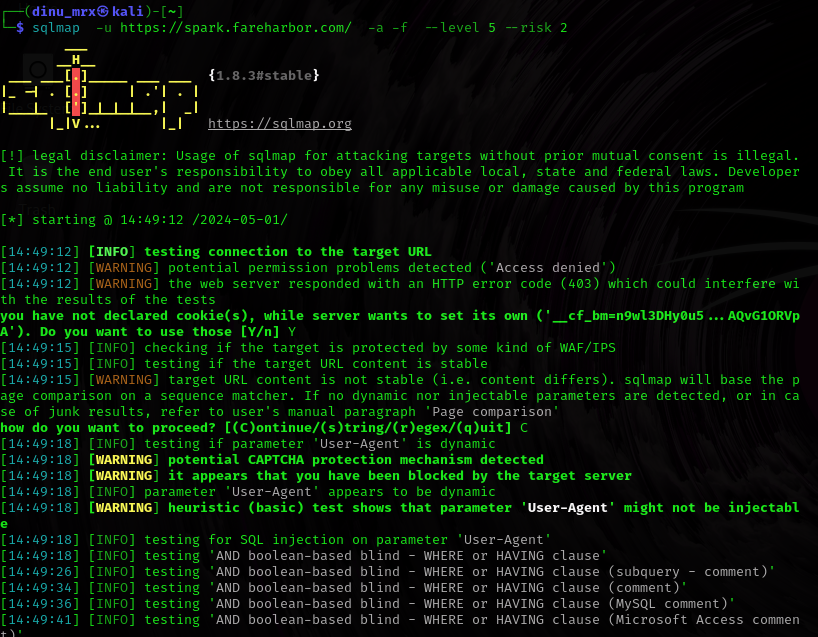
*Note that if you can see error messages or you can spot differences when a query is working and when it's not this phase will be more easy.*

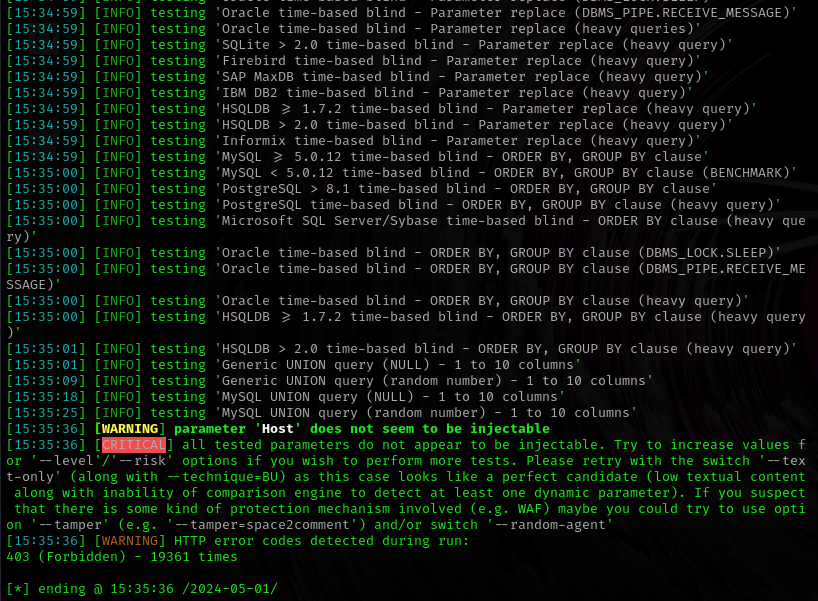


<https://book.hacktricks.xyz/pentesting-web/sql-injection>

A screenshot of a computer

Description automatically generated





According to sqlmap it says that there are no vulnerabilities I run scan with the risk level 2 and level 5 but still it doesnot found any vulnerabilities.

The results obtain from XSStrike



It seems that there are no reflection XSS found currently the web application firewall is disable or offline.