BSc (Hons) in Information Technology – Year 2

IE2062 – Web Security Semester 2, 2024

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

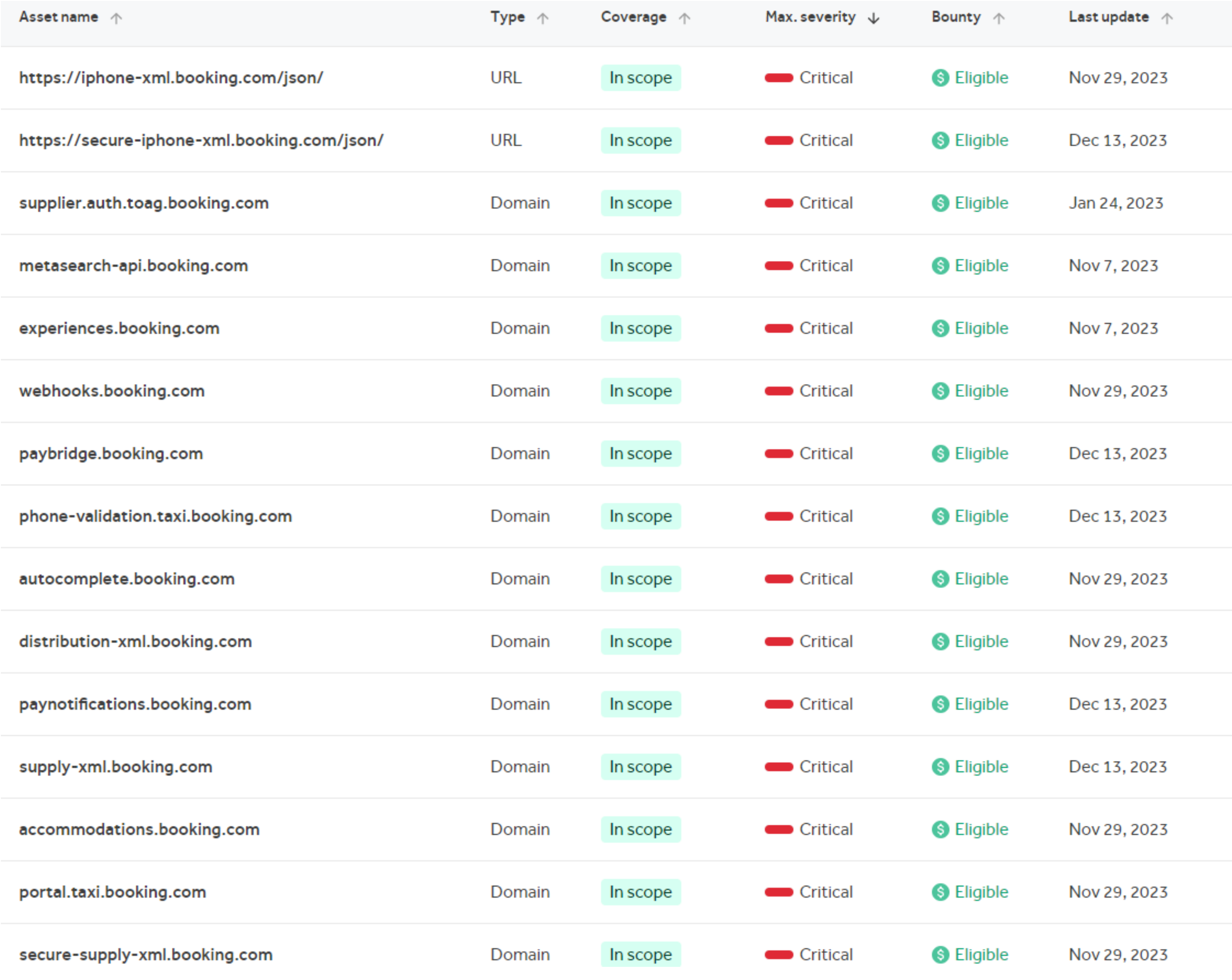
- SSRF
- XXE
- Subdomain takeover in *.booking.com domains

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.

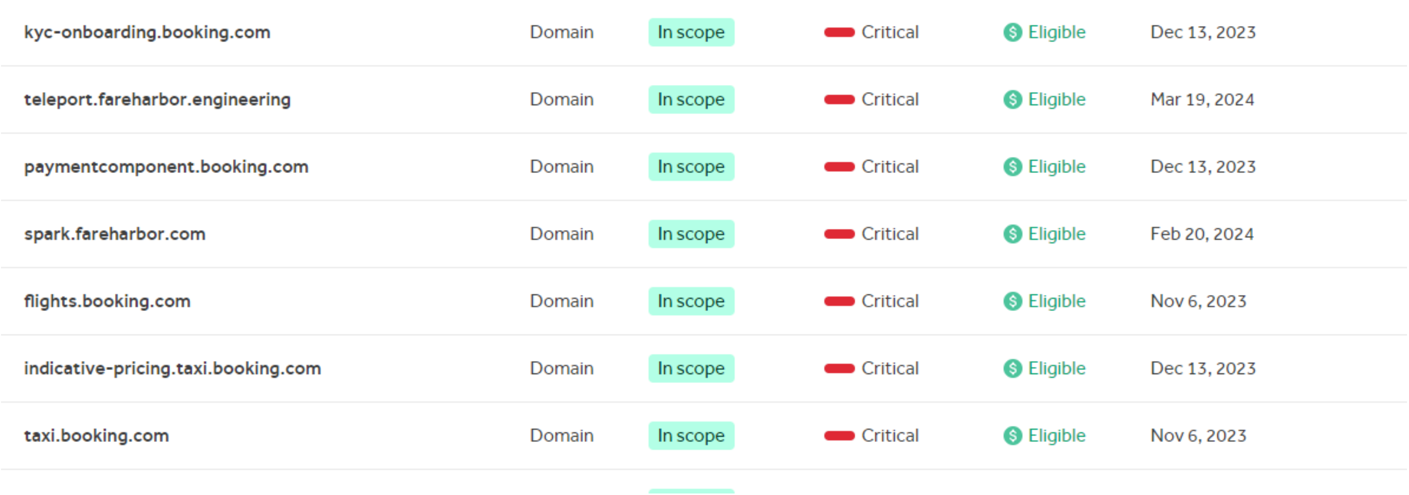
The following issues are outside the scope of our vulnerability rewards program:

- 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

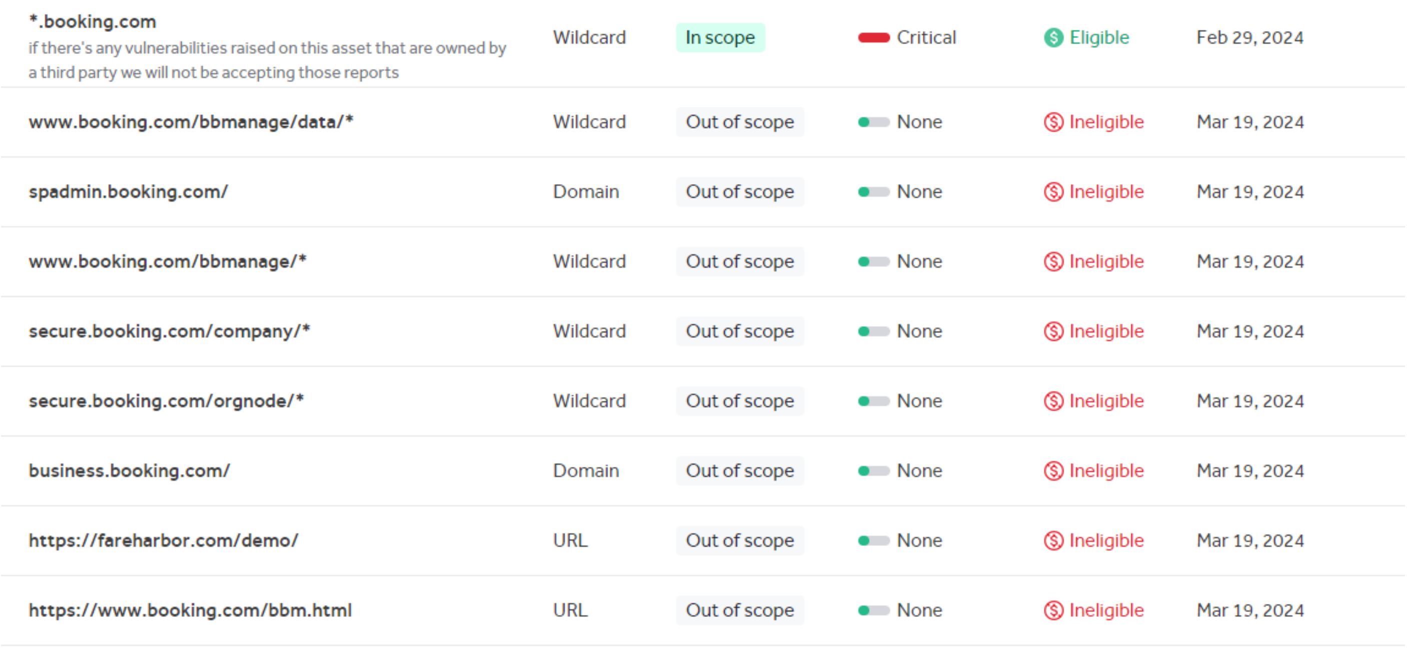
- Any vulnerabilities requiring significant and unlikely interaction by the victim, such as disabling browser controls
- Content spoofing without embedded HTML or JavaScript
- Hypothetical issues that do not have any practical impact
- Infrastructure vulnerabilities, including:
- Issues related to SSL 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



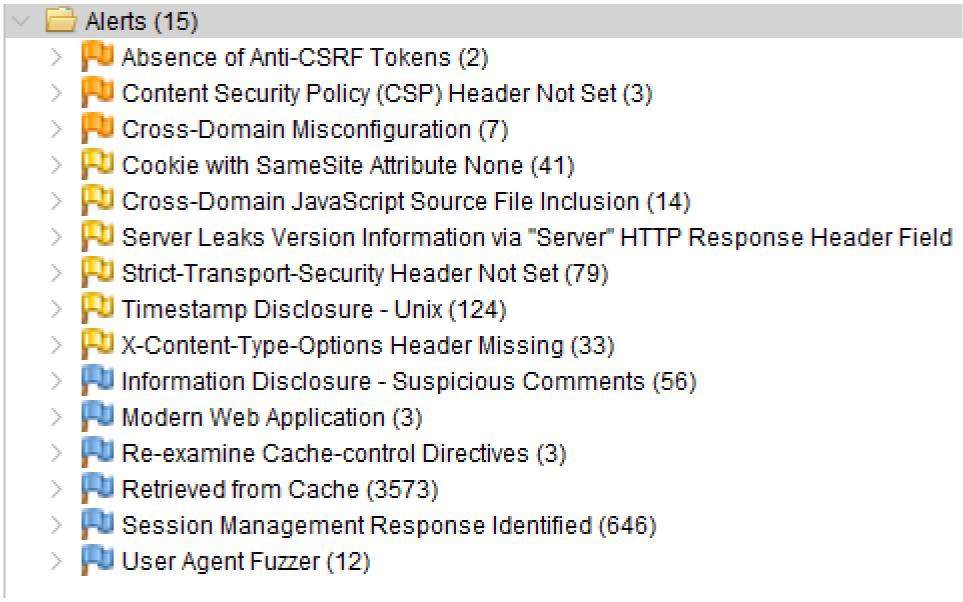
kyc-onboarding.booking.com	Domain	In scope	Critical	Eligible	Dec 13, 2023
teleport.fareharbor.engineering	Domain	In scope	Critical	Eligible	Mar 19, 2024
paymentcomponent.booking.com	Domain	In scope	Critical	Eligible	Dec 13, 2023
spark.fareharbor.com	Domain	In scope	Critical	Eligible	Feb 20, 2024
flights.booking.com	Domain	In scope	Critical	Eligible	Nov 6, 2023
indicative-pricing.taxi.booking.com	Domain	In scope	Critical	Eligible	Dec 13, 2023
taxi.booking.com	Domain	In scope	Critical	Eligible	Nov 6, 2023



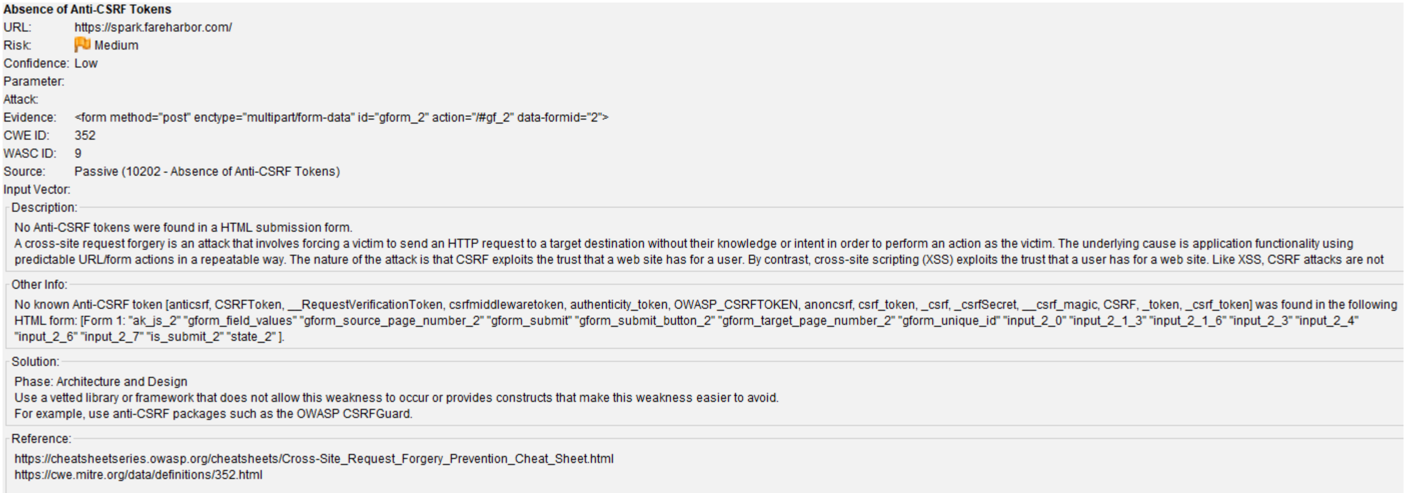
*.booking.com if there's any vulnerabilities raised on this asset that are owned by a third party we will not be accepting those reports	Wildcard	In scope	Critical	Eligible	Feb 29, 2024
www.booking.com/bbmanage/data/*	Wildcard	Out of scope	None	Ineligible	Mar 19, 2024
spadmin.booking.com/	Domain	Out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

https://spark.fareharbor.com/

The results that were obtained from the OWSAP ZAP automated scan



Alerts (15)
- Absence of Anti-CSRF Tokens (2)
- Content Security Policy (CSP) Header Not Set (3)
- Cross-Domain Misconfiguration (7)
- Cookie with SameSite Attribute None (41)
- Cross-Domain JavaScript Source File Inclusion (14)
- Server Leaks Version Information via "Server" HTTP Response Header Field
- Strict-Transport-Security Header Not Set (79)
- Timestamp Disclosure - Unix (124)
- X-Content-Type-Options Header Missing (33)
- Information Disclosure - Suspicious Comments (56)
- Modern Web Application (3)
- Re-examine Cache-control Directives (3)
- Retrieved from Cache (3573)
- Session Management Response Identified (646)
- User Agent Fuzzer (12)



Absence of Anti-CSRF Tokens

URL: https://spark.fareharbor.com/
Risk: Medium
Confidence: Low
Parameter:
Attack:
Evidence: <form method="post" enctype="multipart/form-data" id="gform_2" action="/#gf_2" data-formid="2">
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 not

Other Info:

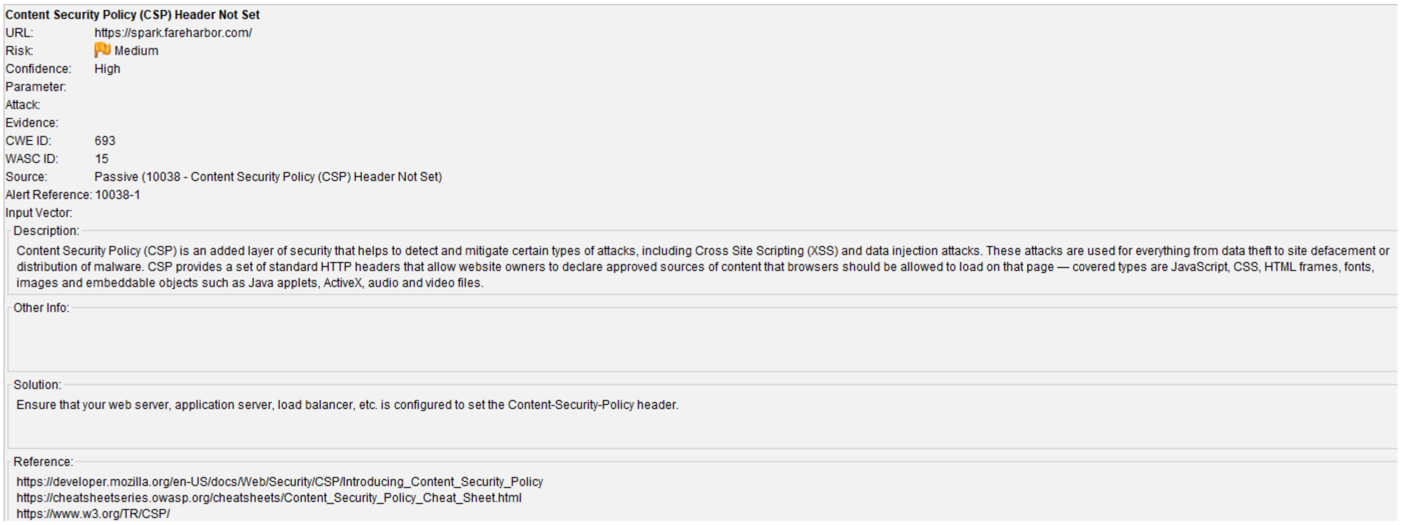
No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "ak_js_2" "gform_field_values" "gform_source_page_number_2" "gform_submit" "gform_submit_button_2" "gform_target_page_number_2" "gform_unique_id" "input_2_0" "input_2_1_3" "input_2_1_6" "input_2_3" "input_2_4" "input_2_6" "input_2_7" "is_submit_2" "state_2"].

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.

Reference:

https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
https://cwe.mitre.org/data/definitions/352.html



Content Security Policy (CSP) Header Not Set
URL: https://spark.fareharbor.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.

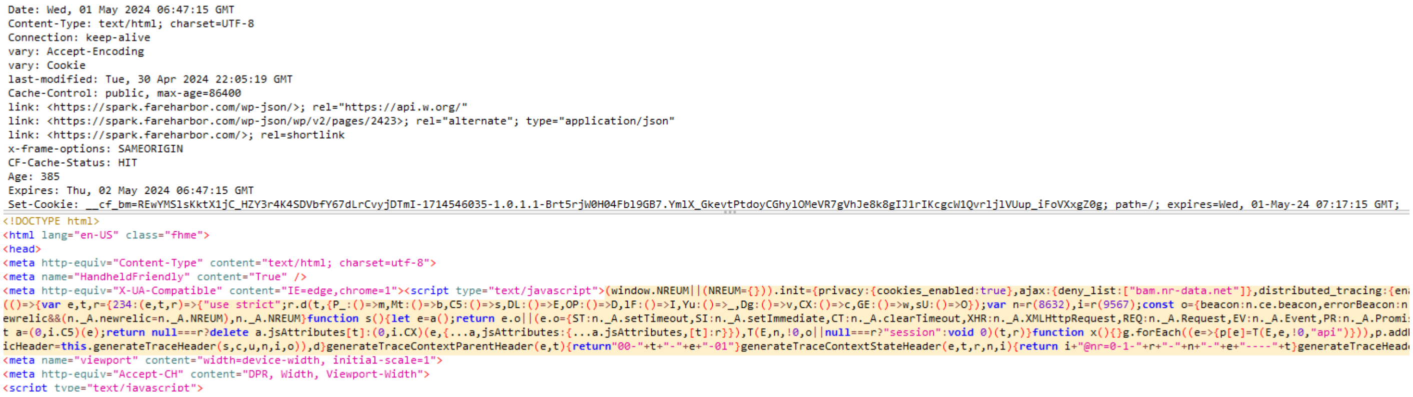
Other Info:

Solution:

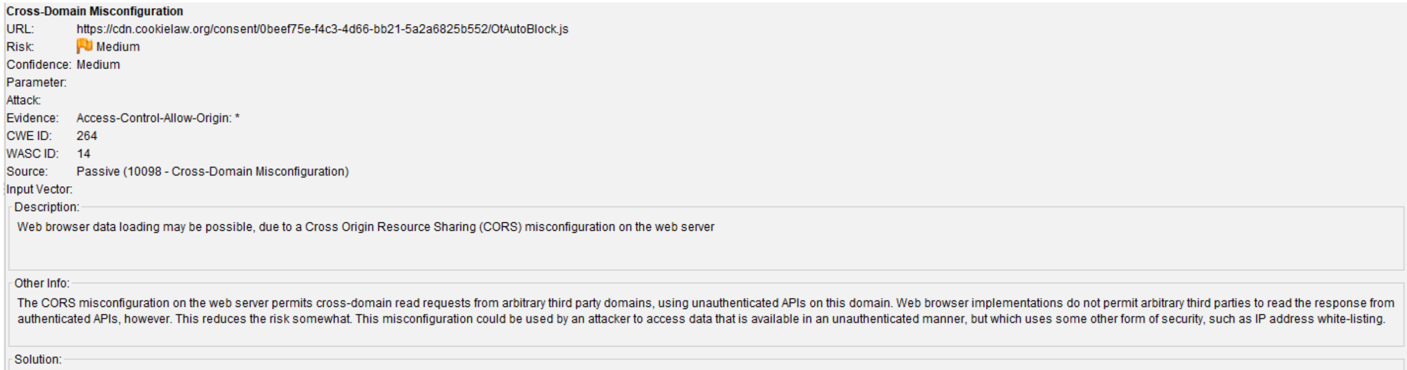
Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Reference:

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/



```
Date: Wed, 01 May 2024 06:47:15 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
vary: Accept-Encoding
vary: Cookie
last-modified: Tue, 30 Apr 2024 22:05:19 GMT
Cache-Control: public, max-age=86400
link: <https://spark.fareharbor.com/wp-json/>; rel="https://api.w.org/"
link: <https://spark.fareharbor.com/wp-json/wp/v2/pages/2423>; rel="alternate"; type="application/json"
link: <https://spark.fareharbor.com/>; rel=shortlink
x-frame-options: SAMEORIGIN
CF-Cache-Status: HIT
Age: 385
Expires: Thu, 02 May 2024 06:47:15 GMT
Set-Cookie: __cf_bm=REwYMSlsKktX1jC_HZY3r4K4SDVbfY67dLrCvyjDTmI-1714546035-1.0.1.1-Brt5rjW0H04Fbl9GB7.YmlX_GkevtPtdoyCGhylOMeVR7gVhJe8k8gIJ1rIKcgcW1QvrljlVUup_iFoVXxgZ0g; path=/; expires=Wed, 01-May-24 07:17:15 GMT;
<!DOCTYPE html>
<html lang="en-US" class="fhme">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="HandheldFriendly" content="True" />
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"><script type="text/javascript">(window.NREUM||(NREUM={})).init={privacy:{cookies_enabled:true},ajax:{deny_list:["bam.nr-data.net"]},distributed_tracing:{ena
(()=>{var e,t,r={234:(e,t,r)=>{"use strict";r.d(t,{P_:()=>m,Mt:()=>b,C5:()=>s,DL:()=>E,OP:()=>D,lF:()=>I,Yu:()=>_,Dg:()=>v,CX:()=>c,GE:()=>w,sU:()=>O});var n=r(8632),i=r(9567);const o={beacon:n.ce.beacon,errorBeacon:n
ewrelic&&(n._A.newrelic=n._A.NREUM),n._A.NREUM}function s(){let e=a();return e.o||(e.o={ST:n._A.setTimeout,SI:n._A.setImmediate,CT:n._A.clearTimeout,XHR:n._A.XMLHttpRequest,REQ:n._A.Request,EV:n._A.Event,PR:n._A.Promi
t a=(0,i.C5)(e);return null===r?delete a.jsAttributes[t]:(0,i.CX)(e,{...a,jsAttributes:{...a.jsAttributes,[t]:r}}),T(E,n,!0,o||null===r?"session":void 0)(t,r)}function x(){}g.forEach((e=>{p[e]=T(E,e,!0,"api")})),p.addl
icHeader=this.generateTraceHeader(s,c,u,n,i,o)),d}generateTraceContextParentHeader(e,t){return"00-"+t+"-"+e+"-01"}generateTraceContextStateHeader(e,t,r,n,i){return i+"@nr=0-1-"+r+"-"+n+"-"+e+"----"+t}generateTraceHead
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Accept-CH" content="DPR, Width, Viewport-Width">
<script type="text/javascript">
```



Cross-Domain Misconfiguration
URL: https://cdn.cookielaw.org/consent/0beef75e-f4c3-4d66-bb21-5a2a6825b552/OtAutoBlock.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:

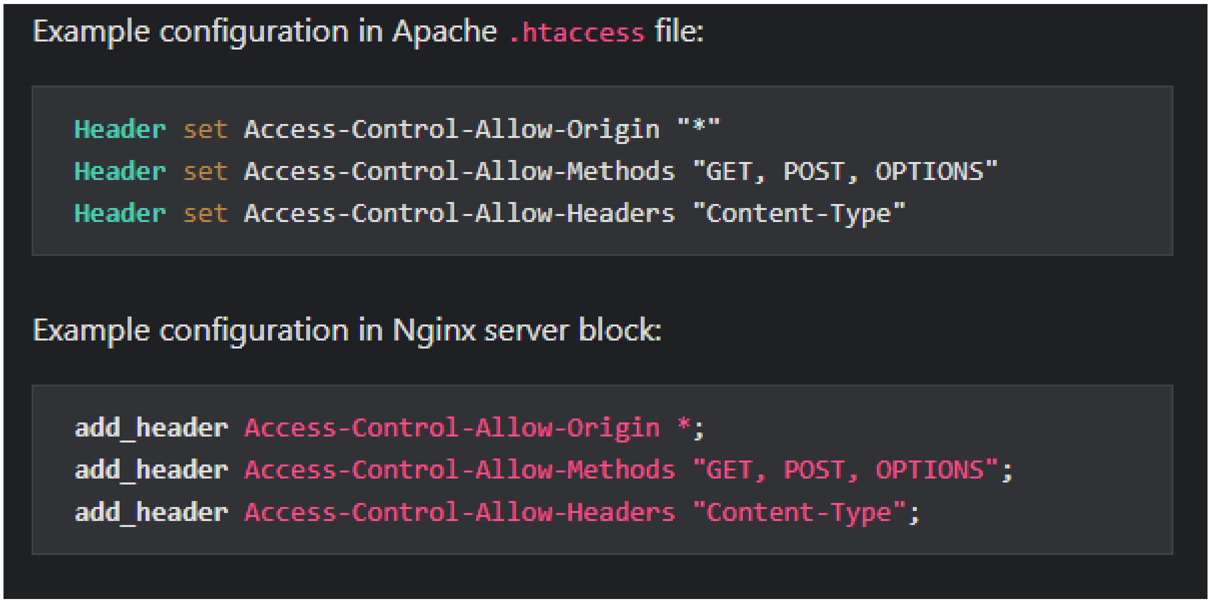
The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk 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:

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.



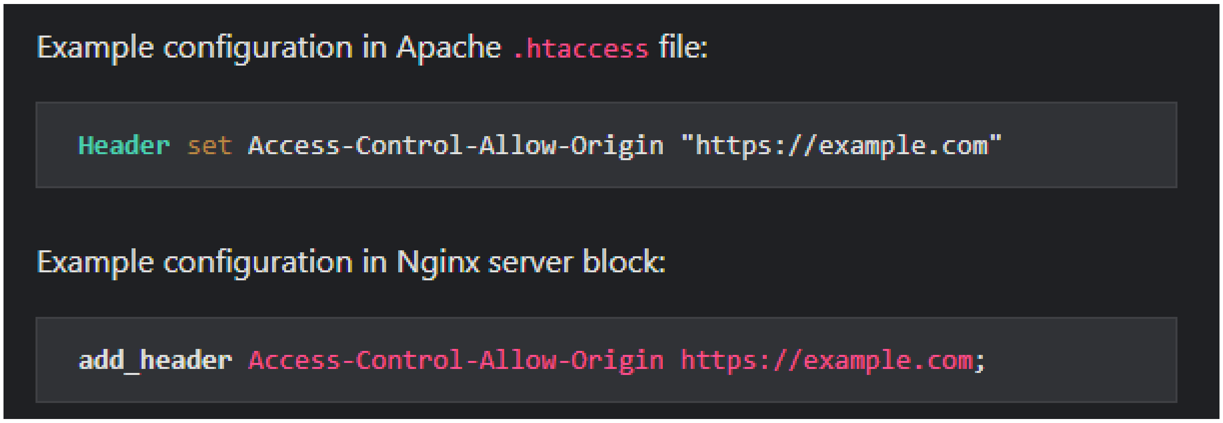
Example configuration in Apache `.htaccess` file:

```
Header set Access-Control-Allow-Origin "*"
Header set Access-Control-Allow-Methods "GET, POST, OPTIONS"
Header set Access-Control-Allow-Headers "Content-Type"
```

Example configuration in Nginx server block:

```
add_header Access-Control-Allow-Origin *;
add_header Access-Control-Allow-Methods "GET, POST, OPTIONS";
add_header Access-Control-Allow-Headers "Content-Type";
```

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



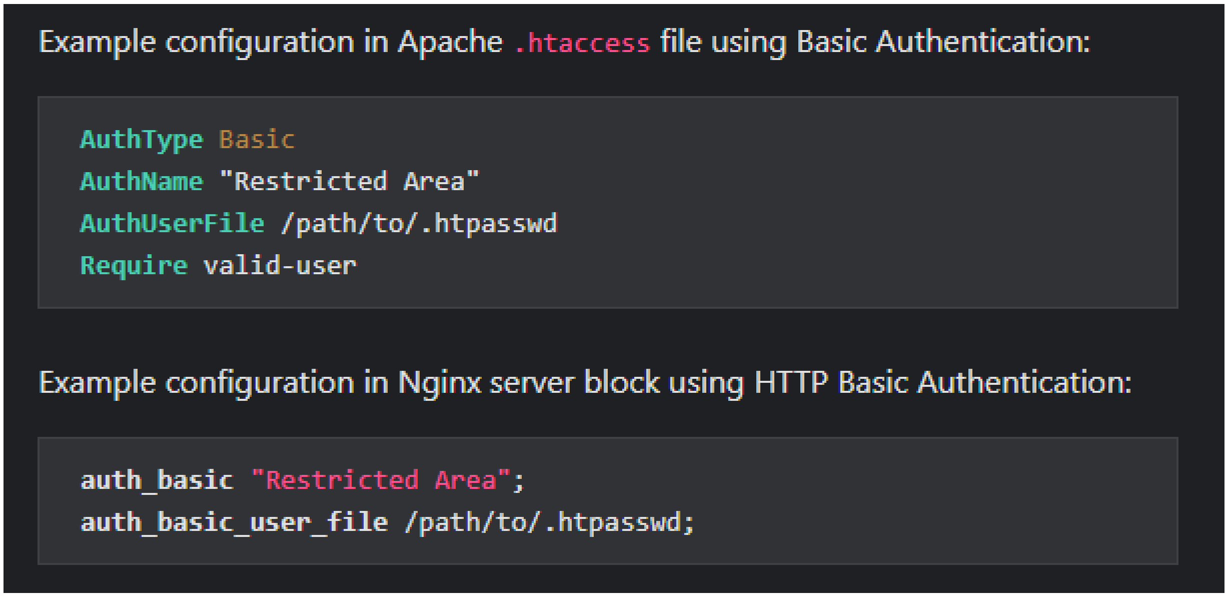
Example configuration in Apache `.htaccess` file:

```
Header set Access-Control-Allow-Origin "https://example.com"
```

Example configuration in Nginx server block:

```
add_header Access-Control-Allow-Origin https://example.com;
```

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



Example configuration in Apache .htaccess file using Basic Authentication:

```
AuthType Basic
AuthName "Restricted Area"
AuthUserFile /path/to/.htpasswd
Require valid-user
```

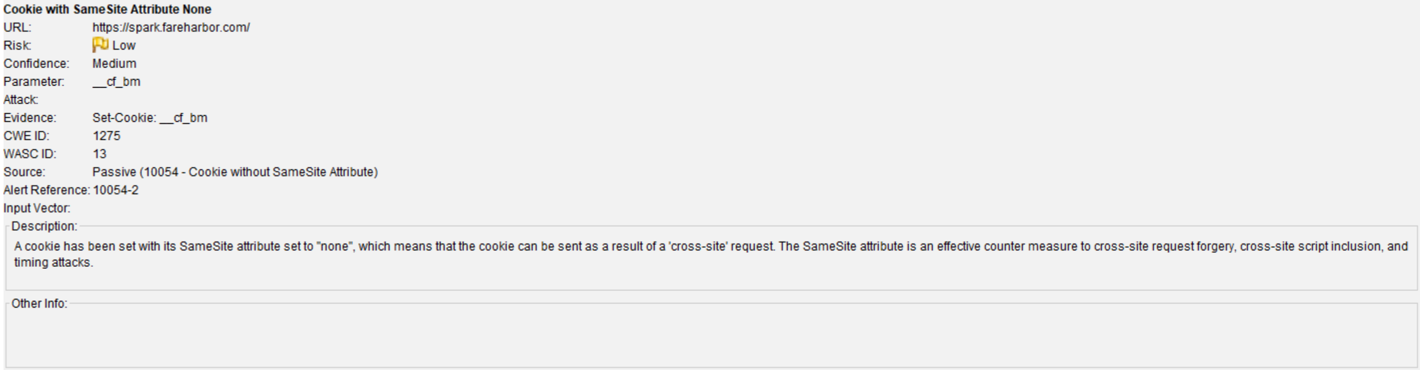
Example configuration in Nginx server block using HTTP Basic Authentication:

```
auth_basic "Restricted Area";
auth_basic_user_file /path/to/.htpasswd;
```

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.



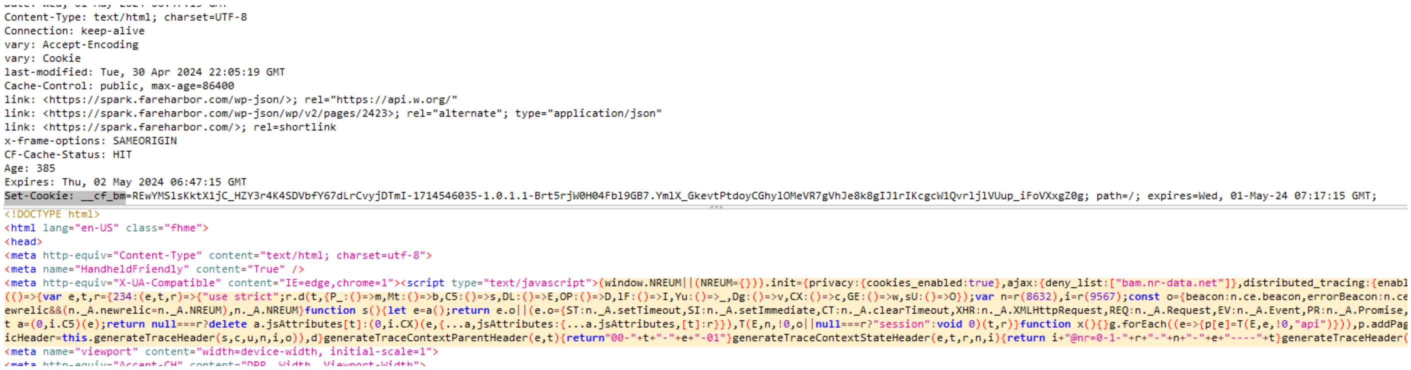
Cookie with SameSite Attribute None

URL: https://spark.fareharbor.com/
Risk: Low
Confidence: Medium
Parameter: __cf_bm
Attack:
Evidence: Set-Cookie: __cf_bm
CWE ID: 1275
WASC ID: 13
Source: Passive (10054 - Cookie without SameSite Attribute)
Alert Reference: 10054-2
Input Vector:

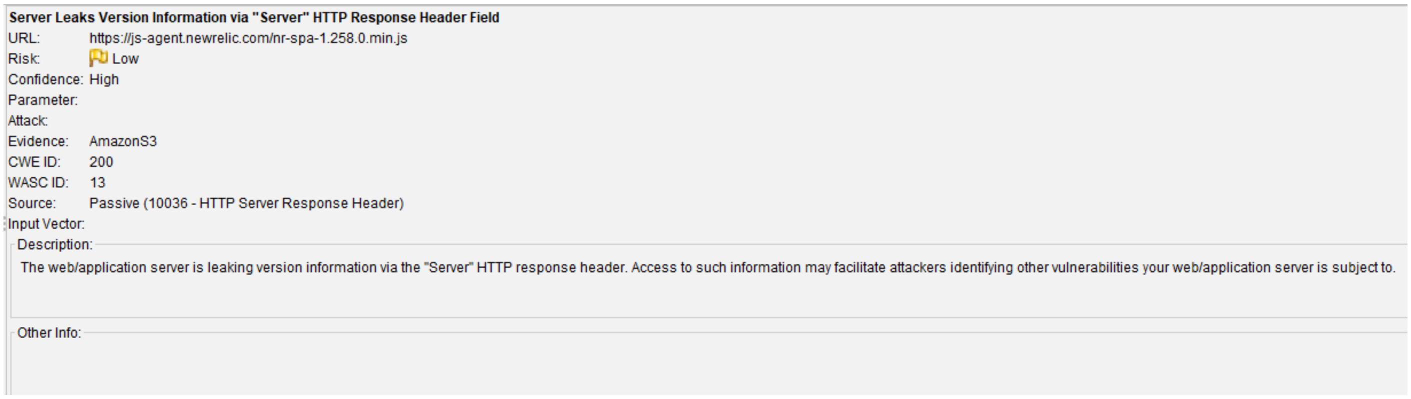
Description:

A cookie has been set with its SameSite attribute set to "none", 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:



```
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
vary: Accept-Encoding
vary: Cookie
last-modified: Tue, 30 Apr 2024 22:05:19 GMT
Cache-Control: public, max-age=86400
link: <https://spark.fareharbor.com/wp-json/>; rel="https://api.w.org/"
link: <https://spark.fareharbor.com/wp-json/wp/v2/pages/2423>; rel="alternate"; type="application/json"
link: <https://spark.fareharbor.com/>; rel=shortlink
x-frame-options: SAMEORIGIN
CF-Cache-Status: HIT
Age: 385
Expires: Thu, 02 May 2024 06:47:15 GMT
Set-Cookie: __cf_bm=REwYMSlsKktX1jC_HZY3r4K4SDVbfY67dLrCvyjDTmI-1714546035-1.0.1.1-Brt5rjW0H04Fbl9GB7.YmlX_GkevtPtdoyCGhylOMeVR7gVhJe8k8gIJ1rIKcgcW1QvrljlVUup_iFoVXxgZ0g; path=/; expires=Wed, 01-May-24 07:17:15 GMT;
<!DOCTYPE html>
<html lang="en-US" class="fhme">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="HandheldFriendly" content="True" />
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"><script type="text/javascript">(window.NREUM||(NREUM={})).init={privacy:{cookies_enabled:true},ajax:{deny_list:["bam.nr-data.net"]},distributed_tracing:{enabl
(()=>{var e,t,r={234:(e,t,r)=>{"use strict";r.d(t,{P_:()=>m,Mt:()=>b,C5:()=>s,DL:()=>E,OP:()=>D,lF:()=>I,Yu:()=>_,Dg:()=>v,CX:()=>c,GE:()=>w,sU:()=>O});var n=r(8632),i=r(9567);const o={beacon:n.ce.beacon,errorBeacon:n.ce
ewrelic&&(n._A.newrelic=n._A.NREUM),n._A.NREUM}function s(){let e=a();return e.o||(e.o={ST:n._A.setTimeout,SI:n._A.setImmediate,CT:n._A.clearTimeout,XHR:n._A.XMLHttpRequest,REQ:n._A.Request,EV:n._A.Event,PR:n._A.Promise,
t a=(0,i.C5)(e);return null===r?delete a.jsAttributes[t]:(0,i.CX)(e,{...a,jsAttributes:{...a.jsAttributes,[t]:r}}),T(E,n,!0,o||null===r?"session":void 0)(t,r)}function x(){}g.forEach((e=>{p[e]=T(E,e,!0,"api")})),p.addPag
icHeader=this.generateTraceHeader(s,c,u,n,i,o)),d}generateTraceContextParentHeader(e,t){return"00-"+t+"-"+e+"-01"}generateTraceContextStateHeader(e,t,r,n,i){return i+"@nr=0-1-"+r+"-"+n+"-"+e+"----"+t}generateTraceHeader(
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Accept-CH" content="DPR, Width, Viewport-Width">
```



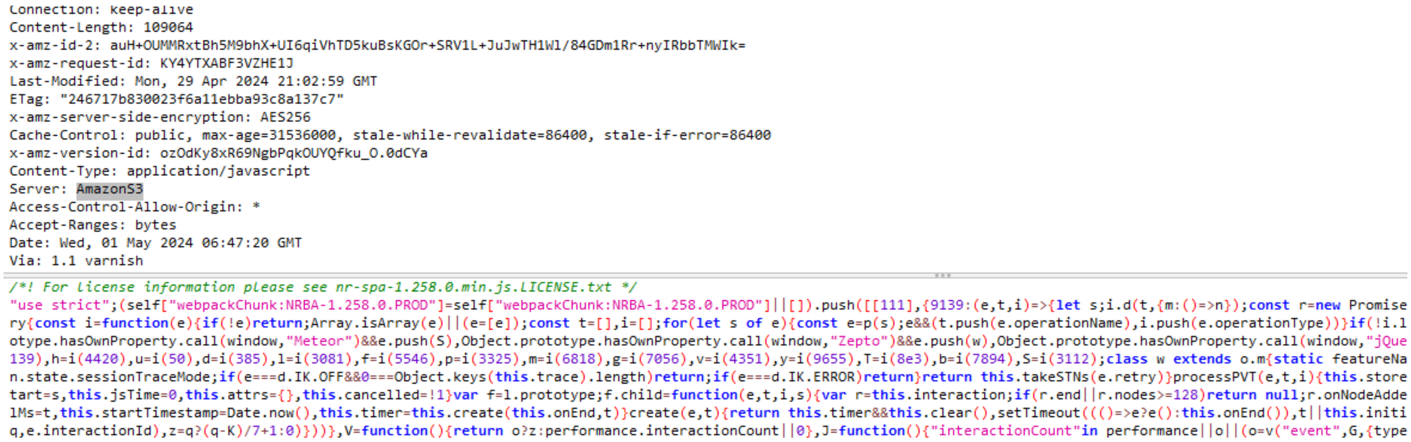
Server Leaks Version Information via "Server" HTTP Response Header Field

URL: https://js-agent.newrelic.com/nr-spa-1.258.0.min.js
Risk: Low
Confidence: High
Parameter:
Attack:
Evidence: AmazonS3
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:



```
Connection: keep-alive
Content-Length: 109064
x-amz-id-2: auH+OUMMRxtBh5M9bhX+UI6qiVhTD5kuBsKGOr+SRV1L+JuJwTH1Wl/84GDm1Rr+nyIRbbTMWIk=
x-amz-request-id: KY4YTXABF3VZHE1J
Last-Modified: Mon, 29 Apr 2024 21:02:59 GMT
ETag: "246717b830023f6a11ebba93c8a137c7"
x-amz-server-side-encryption: AES256
Cache-Control: public, max-age=31536000, stale-while-revalidate=86400, stale-if-error=86400
x-amz-version-id: ozOdKy8xR69NgbPqkOUYQfku_O.0dCYa
Content-Type: application/javascript
Server: AmazonS3
Access-Control-Allow-Origin: *
Accept-Ranges: bytes
Date: Wed, 01 May 2024 06:47:20 GMT
Via: 1.1 varnish
```

```
/*! For license information please see nr-spa-1.258.0.min.js.LICENSE.txt */
"use strict";(self["webpackChunk:NRBA-1.258.0.PROD"]=self["webpackChunk:NRBA-1.258.0.PROD"]||[]).push([[111],{9139:(e,t,i)=>{let s;i.d(t,{m:()=>n});const r=new Promise
ry{const i=function(e){if(!e)return;Array.isArray(e)||(e=[e]);const t=[],i=[];for(let s of e){const e=p(s);e&&(t.push(e.operationName),i.push(e.operationType))}if(!i.l
otype.hasOwnProperty.call(window,"Meteor")&&e.push(S),Object.prototype.hasOwnProperty.call(window,"Zepto")&&e.push(w),Object.prototype.hasOwnProperty.call(window,"jQue
139),h=i(4420),u=i(50),d=i(385),l=i(3081),f=i(5546),p=i(3325),m=i(6818),g=i(7056),v=i(4351),y=i(9655),T=i(8e3),b=i(7894),S=i(3112);class w extends o.m{static featureNa
n.state.sessionTraceMode;if(e===d.IK.OFF&&0===Object.keys(this.trace).length)return;if(e===d.IK.ERROR)return}return this.takeSTNs(e.retry)}processPVT(e,t,i){this.store
tart=s,this.jsTime=0,this.attrs={},this.cancelled=!1}var f=l.prototype;f.child=function(e,t,i,s){var r=this.interaction;if(r.end||r.nodes>=128)return null;r.onNodeAdde
lMs=t,this.startTimestamp=Date.now(),this.timer=this.create(this.onEnd,t)}create(e,t){return this.timer&&this.clear(),setTimeout((()=>e?e():this.onEnd()),t||this.initi
q,e.interactionId),z=q?(q-K)/7+1:0)}))},V=function(){return o?z:performance.interactionCount||0},J=function(){"interactionCount"in performance||o||(o=v("event",G,{type
```

Description

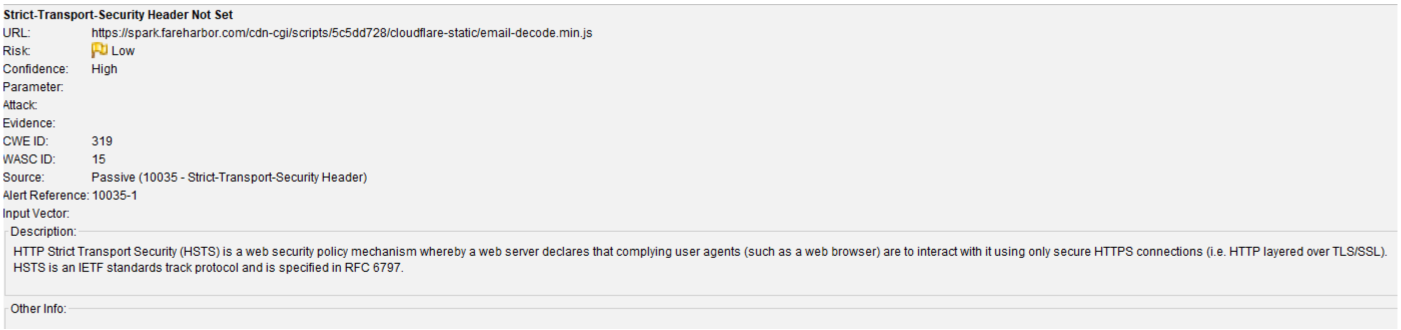
Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. Files within S3 are organized into "buckets", which are named logical containers accessible at a predictable URL. Access controls can be applied to both 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/



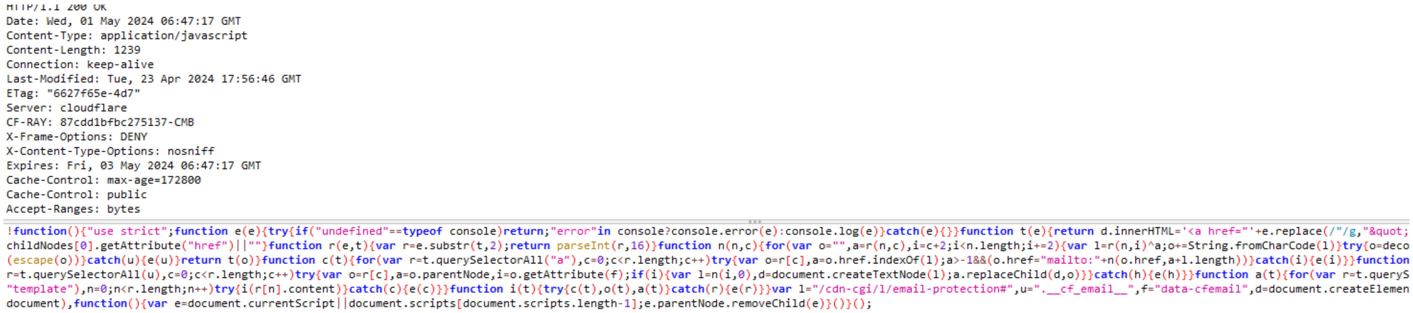
Strict-Transport-Security Header Not Set

URL: https://spark.fareharbor.com/cdn-cgi/scripts/5c5dd728/cloudflare-static/email-decode.min.js
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.

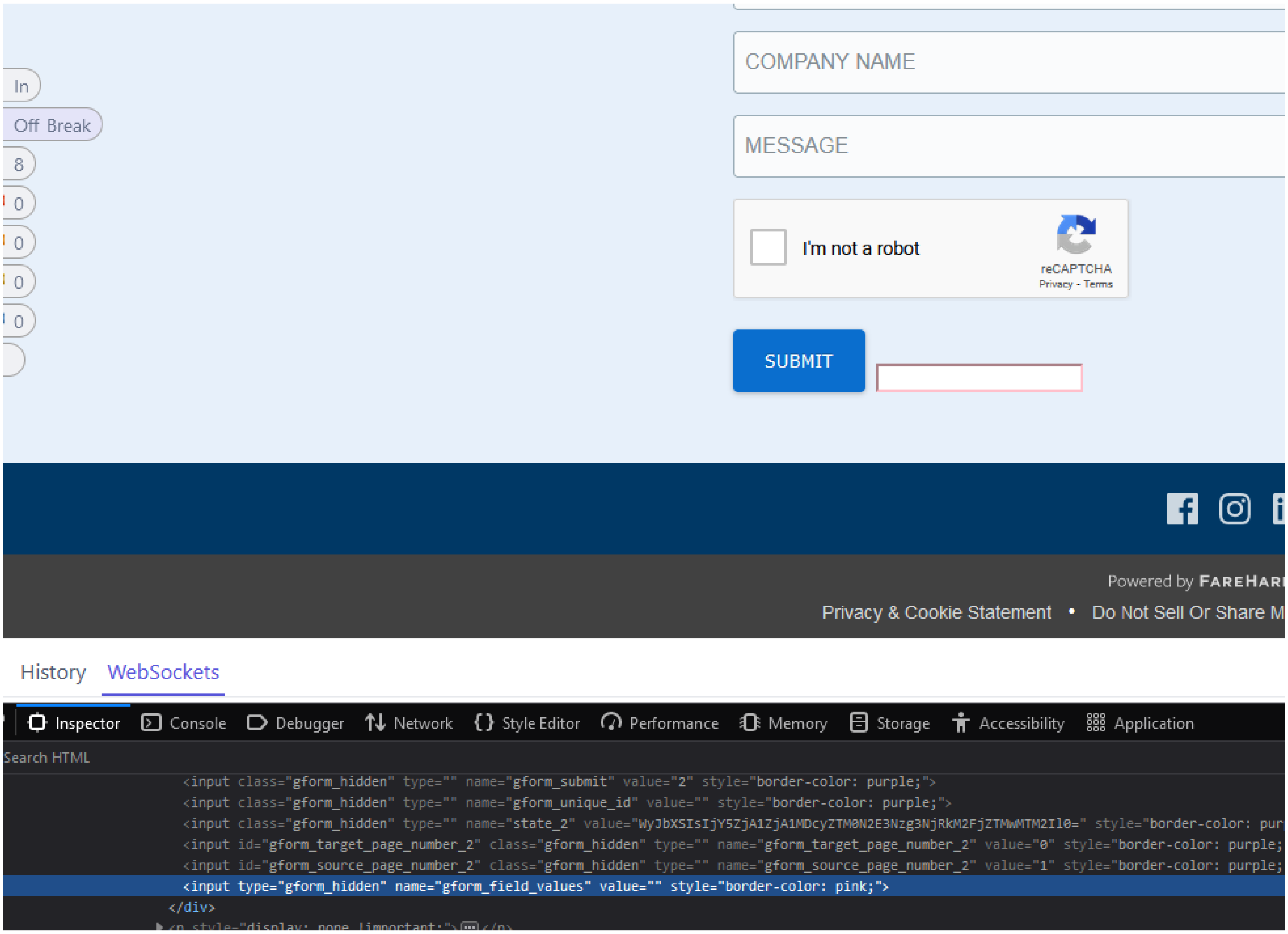
Other Info:



```
HTTP/1.1 200 OK
Date: Wed, 01 May 2024 06:47:17 GMT
Content-Type: application/javascript
Content-Length: 1239
Connection: keep-alive
Last-Modified: Tue, 23 Apr 2024 17:56:46 GMT
ETag: "6627f65e-4d7"
Server: cloudflare
CF-RAY: 87cdd1bfbc275137-CMB
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Expires: Fri, 03 May 2024 06:47:17 GMT
Cache-Control: max-age=172800
Cache-Control: public
Accept-Ranges: bytes
```

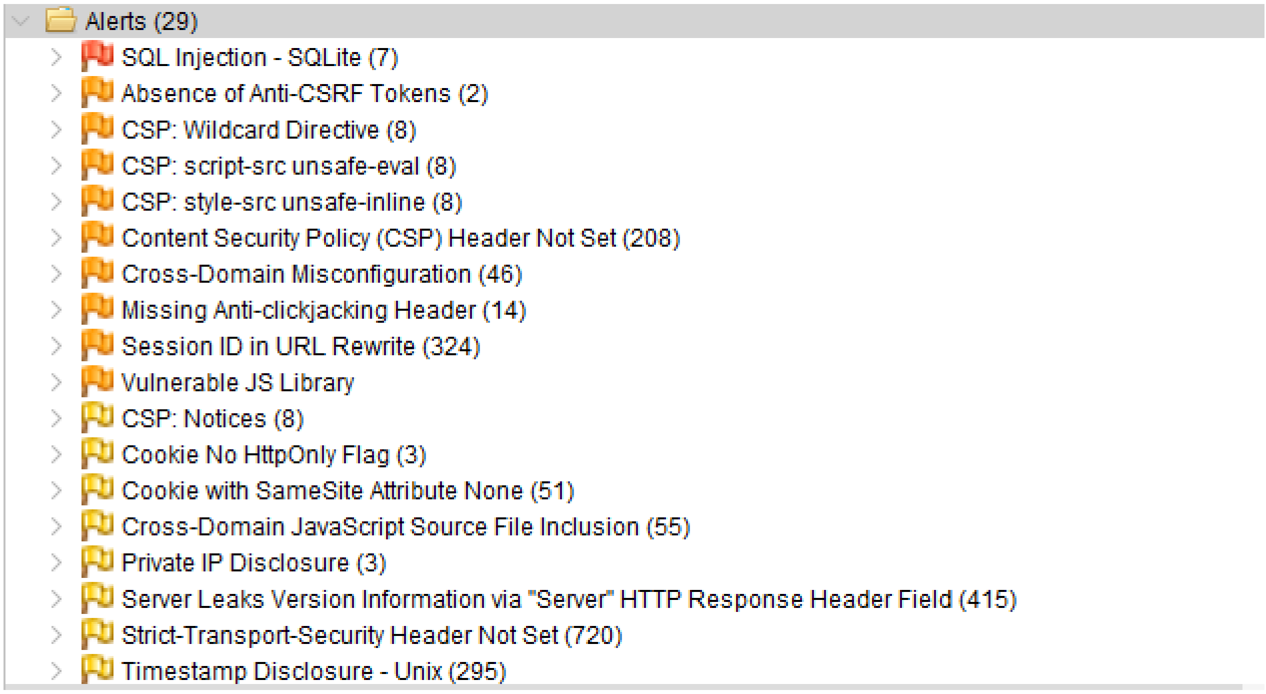
```
!function(){"use strict";function e(e){try{if("undefined"==typeof console)return;"error"in console?console.error(e):console.log(e)}catch(e){}}function t(e){return d.innerHTML='<a href="'+e.replace(/"/g,"&quot;
childNodes[0].getAttribute("href")||""}function r(e,t){var r=e.substr(t,2);return parseInt(r,16)}function n(n,c){for(var o="",a=r(n,c),i=c+2;i<n.length;i+=2){var l=r(n,i)^a;o+=String.fromCharCode(l)}try{o=deco
(escape(o))}catch(u){e(u)}return t(o)}function c(t){for(var r=t.querySelectorAll("a"),c=0;c<r.length;c++)try{var o=r[c],a=o.href.indexOf(l);a>-1&&(o.href="mailto:"+n(o.href,a+l.length))}catch(i){e(i)}}function
r=t.querySelectorAll(u),c=0;c<r.length;c++)try{var o=r[c],a=o.parentNode,i=o.getAttribute(f);if(i){var l=n(i,0),d=document.createTextNode(l);a.replaceChild(d,o)}}catch(h){e(h)}}function a(t){for(var r=t.queryS
"template"),n=0;n<r.length;n++)try{i(r[n].content)}catch(c){e(c)}}function i(t){try{c(t),o(t),a(t)}catch(r){e(r)}}var l="/cdn-cgi/l/email-protection#",u=".__cf_email__",f="data-cfemail",d=document.createElemen
document),function(){var e=document.currentScript||document.scripts[document.scripts.length-1];e.parentNode.removeChild(e)}()}();
```

In this site there are 8 hidden fields

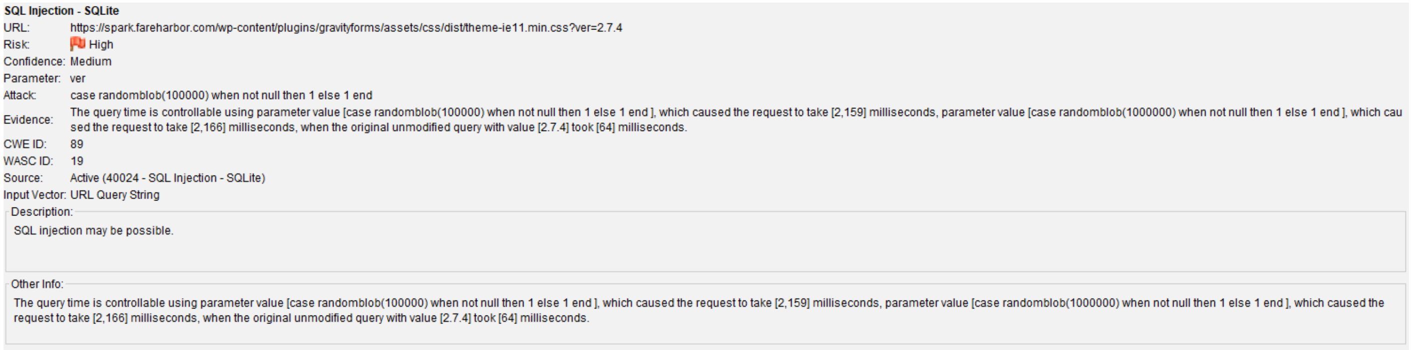


```
            <input class="gform_hidden" type="" name="gform_submit" value="2" style="border-color: purple;">
            <input class="gform_hidden" type="" name="gform_unique_id" value="" style="border-color: purple;">
            <input class="gform_hidden" type="" name="state_2" value="WyJbXSIsIjY5ZjA1ZjA1MDcyZTM0N2E3Nzg3NjRkM2FjZTMwMTM2Il0=" style="border-color: pur
            <input id="gform_target_page_number_2" class="gform_hidden" type="" name="gform_target_page_number_2" value="0" style="border-color: purple;
            <input id="gform_source_page_number_2" class="gform_hidden" type="" name="gform_source_page_number_2" value="1" style="border-color: purple;
            <input type="gform_hidden" name="gform_field_values" value="" style="border-color: pink;">
          </div>
        <p style="display: none !important;">…</p>
```

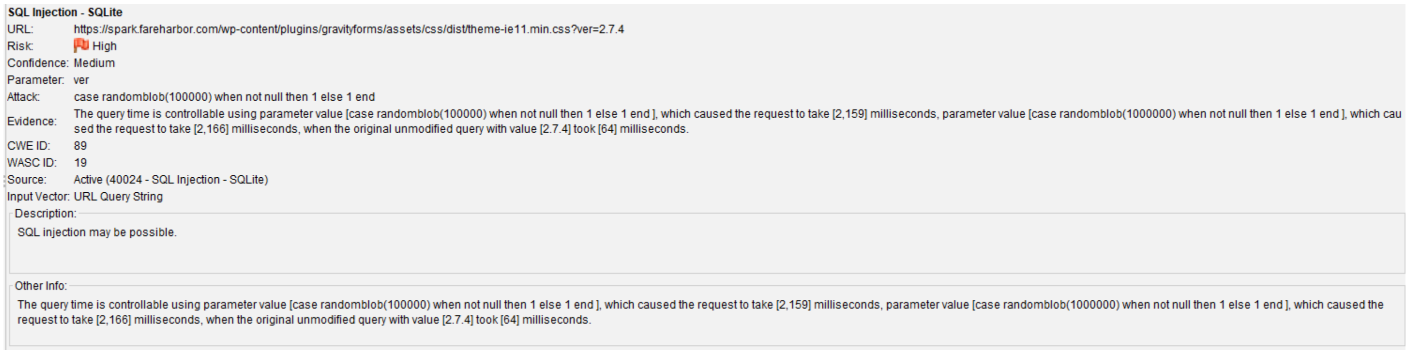
Results from the manual scan



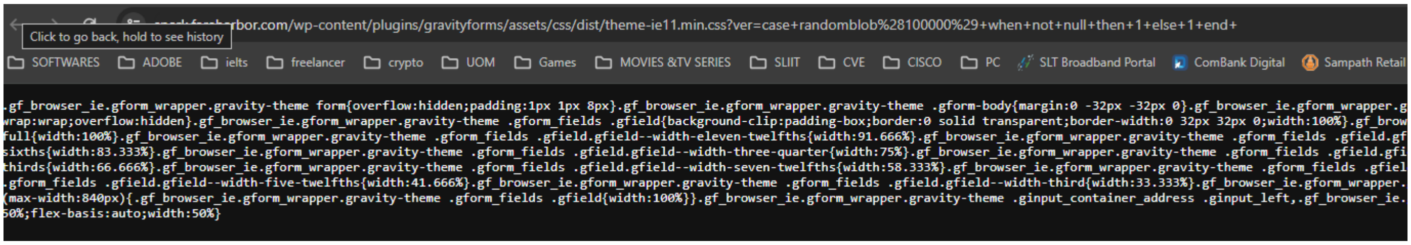
```
Alerts (29)
  SQL Injection - SQLite (7)
  Absence of Anti-CSRF Tokens (2)
  CSP: Wildcard Directive (8)
  CSP: script-src unsafe-eval (8)
  CSP: style-src unsafe-inline (8)
  Content Security Policy (CSP) Header Not Set (208)
  Cross-Domain Misconfiguration (46)
  Missing Anti-clickjacking Header (14)
  Session ID in URL Rewrite (324)
  Vulnerable JS Library
  CSP: Notices (8)
  Cookie No HttpOnly Flag (3)
  Cookie with SameSite Attribute None (51)
  Cross-Domain JavaScript Source File Inclusion (55)
  Private IP Disclosure (3)
  Server Leaks Version Information via "Server" HTTP Response Header Field (415)
  Strict-Transport-Security Header Not Set (720)
  Timestamp Disclosure - Unix (295)
```



```
SQL Injection - SQLite
URL:          https://spark.fareharbor.com/wp-content/plugins/gravityforms/assets/css/dist/theme-ie11.min.css?ver=2.7.4
Risk:         High
Confidence:   Medium
Parameter:    ver
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 [2,159] milliseconds, parameter value [case randomblob(1000000) when not null then 1 else 1 end ], which cau
              sed the request to take [2,166] milliseconds, when the original unmodified query with value [2.7.4] took [64] milliseconds.
CWE ID:       89
WASC ID:      19
Source:       Active (40024 - SQL Injection - SQLite)
Input Vector: URL Query String
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 [2,159] milliseconds, parameter value [case randomblob(1000000) when not null then 1 else 1 end ], which caused the
 request to take [2,166] milliseconds, when the original unmodified query with value [2.7.4] took [64] milliseconds.
```

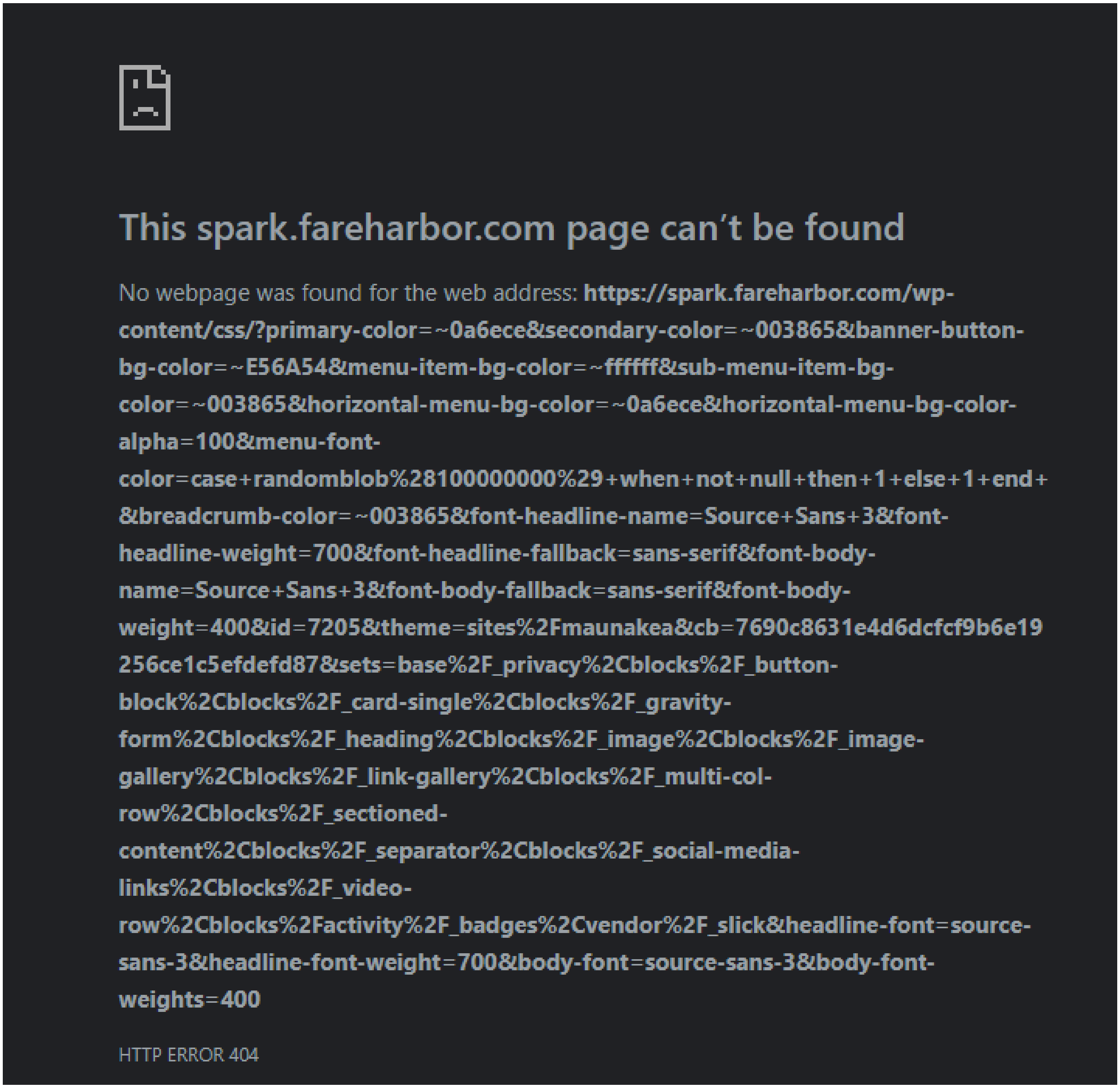


```
SQL Injection - SQLite
URL:          https://spark.fareharbor.com/wp-content/plugins/gravityforms/assets/css/dist/theme-ie11.min.css?ver=2.7.4
Risk:         High
Confidence:   Medium
Parameter:    ver
Attack:       case randomblob(100000) when not null then 1 else 1 end
Evidence:     The query time is controllable using parameter value [case randomblob(100000) when not null then 1 else 1 end ], which caused the request to take [2,159] milliseconds, parameter value [case randomblob(1000000) when not null then 1 else 1 end ], which cau
              sed the request to take [2,166] milliseconds, when the original unmodified query with value [2.7.4] took [64] milliseconds.
CWE ID:       89
WASC ID:      19
Source:       Active (40024 - SQL Injection - SQLite)
Input Vector: URL Query String
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 [2,159] milliseconds, parameter value [case randomblob(1000000) when not null then 1 else 1 end ], which caused the
 request to take [2,166] milliseconds, when the original unmodified query with value [2.7.4] took [64] milliseconds.
```



```
rbor.com/wp-content/plugins/gravityforms/assets/css/dist/theme-ie11.min.css?ver=case+randomblob%28100000%29+when+not+null+then+1+else+1+end+
```

```
.gf_browser_ie.gform_wrapper.gravity-theme form{overflow:hidden;padding:1px 1px 8px}.gf_browser_ie.gform_wrapper.gravity-theme .gform-body{margin:0 -32px -32px 0}.gf_browser_ie.gform_wrapper.g
wrap:wrap;overflow:hidden}.gf_browser_ie.gform_wrapper.gravity-theme .gform_fields .gfield{background-clip:padding-box;border:0 solid transparent;border-width:0 32px 32px 0;width:100%}.gf_brow
full{width:100%}.gf_browser_ie.gform_wrapper.gravity-theme .gform_fields .gfield.gfield--width-eleven-twelfths{width:91.666%}.gf_browser_ie.gform_wrapper.gravity-theme .gform_fields .gfield.gf
sixths{width:83.333%}.gf_browser_ie.gform_wrapper.gravity-theme .gform_fields .gfield.gfield--width-three-quarter{width:75%}.gf_browser_ie.gform_wrapper.gravity-theme .gform_fields .gfield.gfi
thirds{width:66.666%}.gf_browser_ie.gform_wrapper.gravity-theme .gform_fields .gfield.gfield--width-seven-twelfths{width:58.333%}.gf_browser_ie.gform_wrapper.gravity-theme .gform_fields .gfiel
.gform_fields .gfield.gfield--width-five-twelfths{width:41.666%}.gf_browser_ie.gform_wrapper.gravity-theme .gform_fields .gfield.gfield--width-third{width:33.333%}.gf_browser_ie.gform_wrapper.
(max-width:840px){.gf_browser_ie.gform_wrapper.gravity-theme .gform_fields .gfield{width:100%}}.gf_browser_ie.gform_wrapper.gravity-theme .ginput_container_address .ginput_left,.gf_browser_ie.
50%;flex-basis:auto;width:50%}
```



This spark.fareharbor.com page can’t be found

No webpage was found for the web address: **https://spark.fareharbor.com/wp-content/css/?primary-color=~0a6ece&secondary-color=~003865&banner-button-bg-color=~E56A54&menu-item-bg-color=~ffffff&sub-menu-item-bg-color=~003865&horizontal-menu-bg-color=~0a6ece&horizontal-menu-bg-color-alpha=100&menu-font-color=case+randomblob%28100000000%29+when+not+null+then+1+else+1+end+&breadcrumb-color=~003865&font-headline-name=Source+Sans+3&font-headline-weight=700&font-headline-fallback=sans-serif&font-body-name=Source+Sans+3&font-body-fallback=sans-serif&font-body-weight=400&id=7205&theme=sites%2Fmaunakea&cb=7690c8631e4d6dcfcf9b6e19256ce1c5efdefd87&sets=base%2F_privacy%2Cblocks%2F_button-block%2Cblocks%2F_card-single%2Cblocks%2F_gravity-form%2Cblocks%2F_heading%2Cblocks%2F_image%2Cblocks%2F_image-gallery%2Cblocks%2F_link-gallery%2Cblocks%2F_multi-col-row%2Cblocks%2F_sectioned-content%2Cblocks%2F_separator%2Cblocks%2F_social-media-links%2Cblocks%2F_video-row%2Cblocks%2Factivity%2F_badges%2Cvendor%2F_slick&headline-font=source-sans-3&headline-font-weight=700&body-font=source-sans-3&body-font-weights=400**

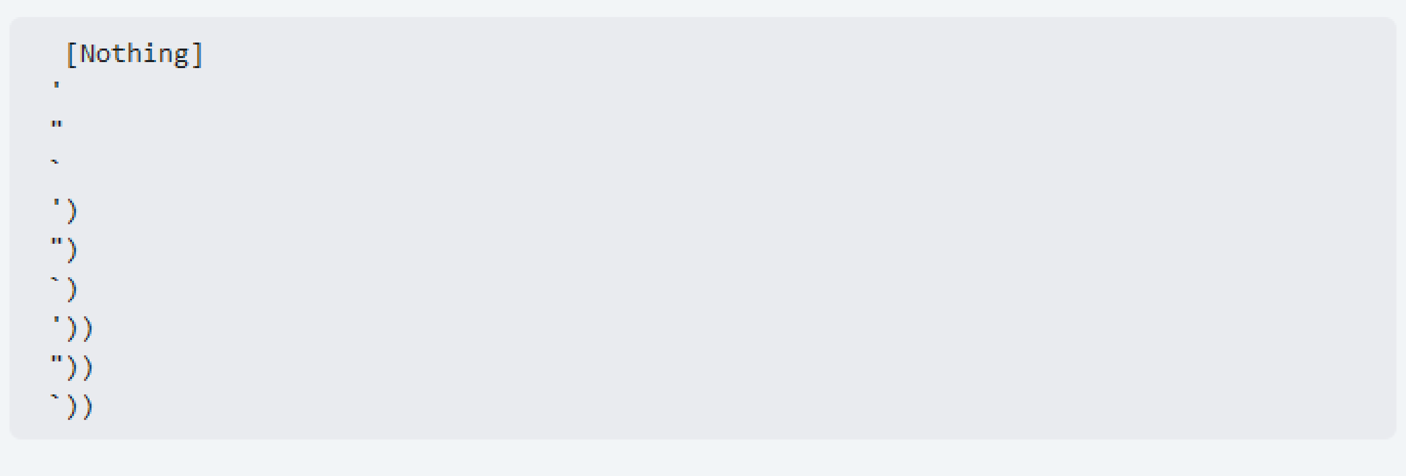
HTTP ERROR 404

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

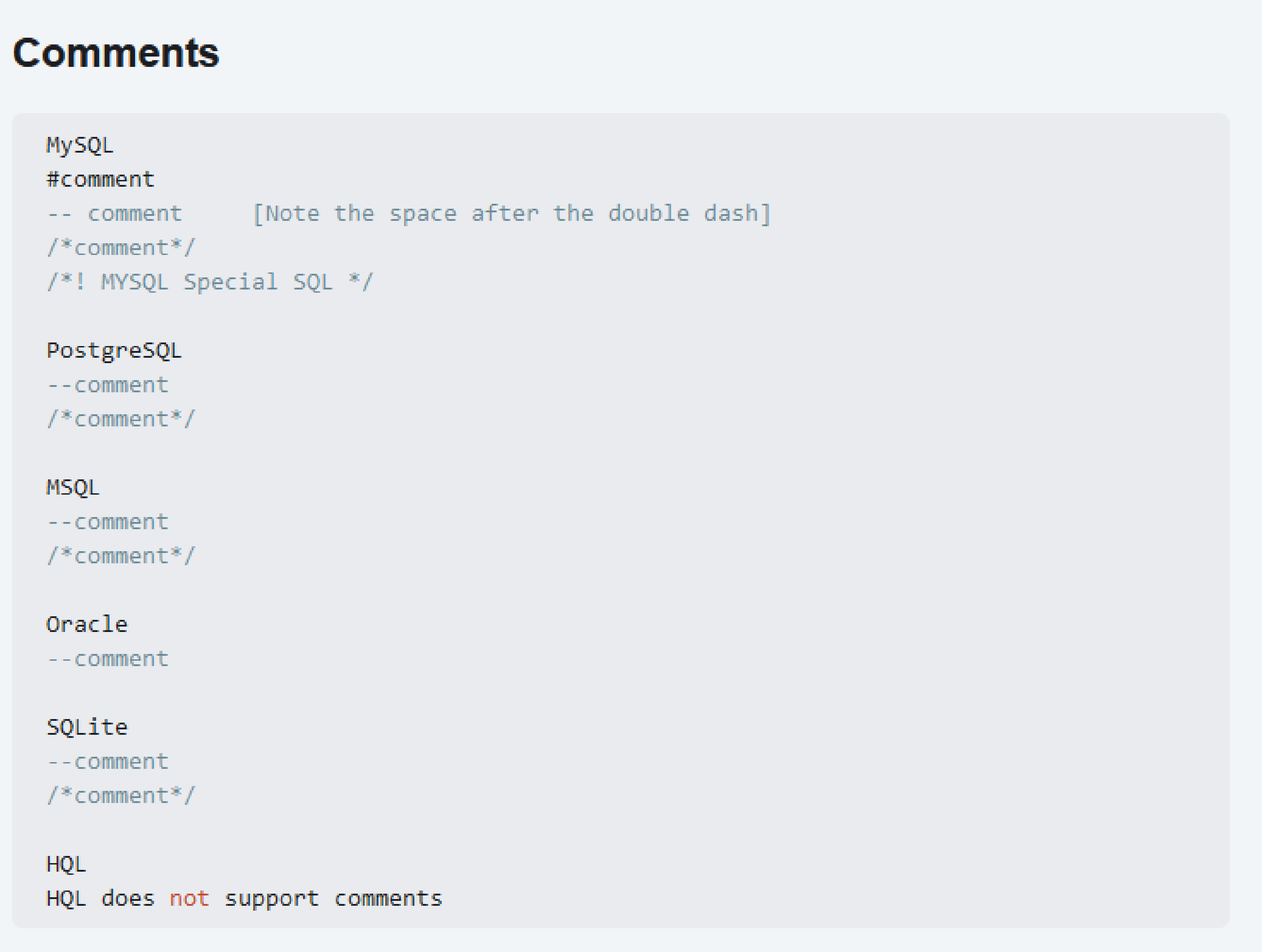
When a site appears to be **vulnerable to SQL injection (SQLi)** due to unusual server responses to SQLi-related inputs, the **first step** is to understand how to **inject data into the query without disrupting it**. This requires identifying the method to **escape from the current context** effectively.



```
 [Nothing]
'
"
`
')
")
`)
'))
"))
`))
```

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**.

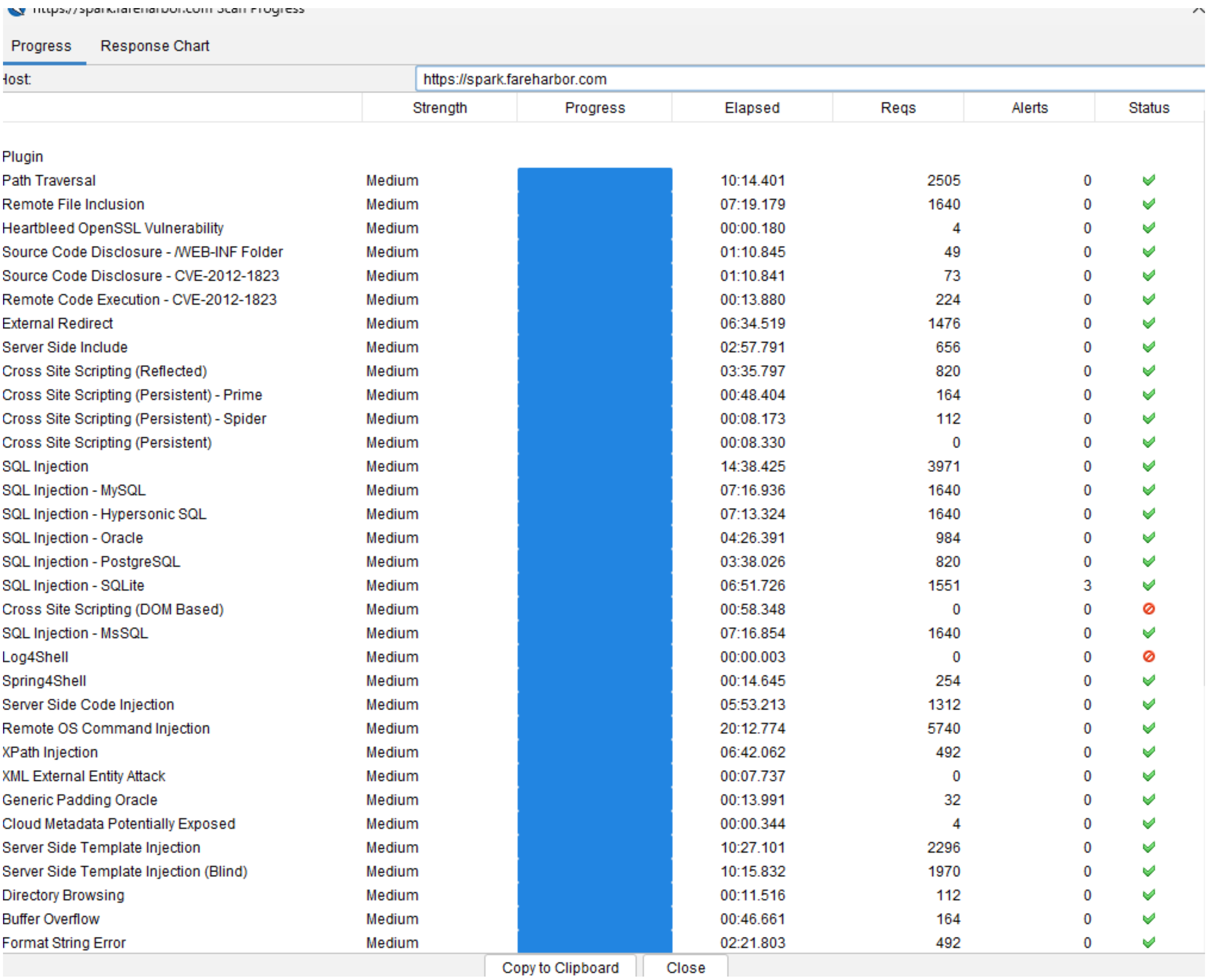
Note that if you can see error messages or you can spot differences when a query is working and when it's not this phase will be more easy.



Comments

```
MySQL
#comment
-- comment     [Note the space after the double dash]
/*comment*/
/*! MYSQL Special SQL */

PostgreSQL
--comment
/*comment*/

MSQL
--comment
/*comment*/

Oracle
--comment

SQLite
--comment
/*comment*/

HQL
HQL does not support comments
```

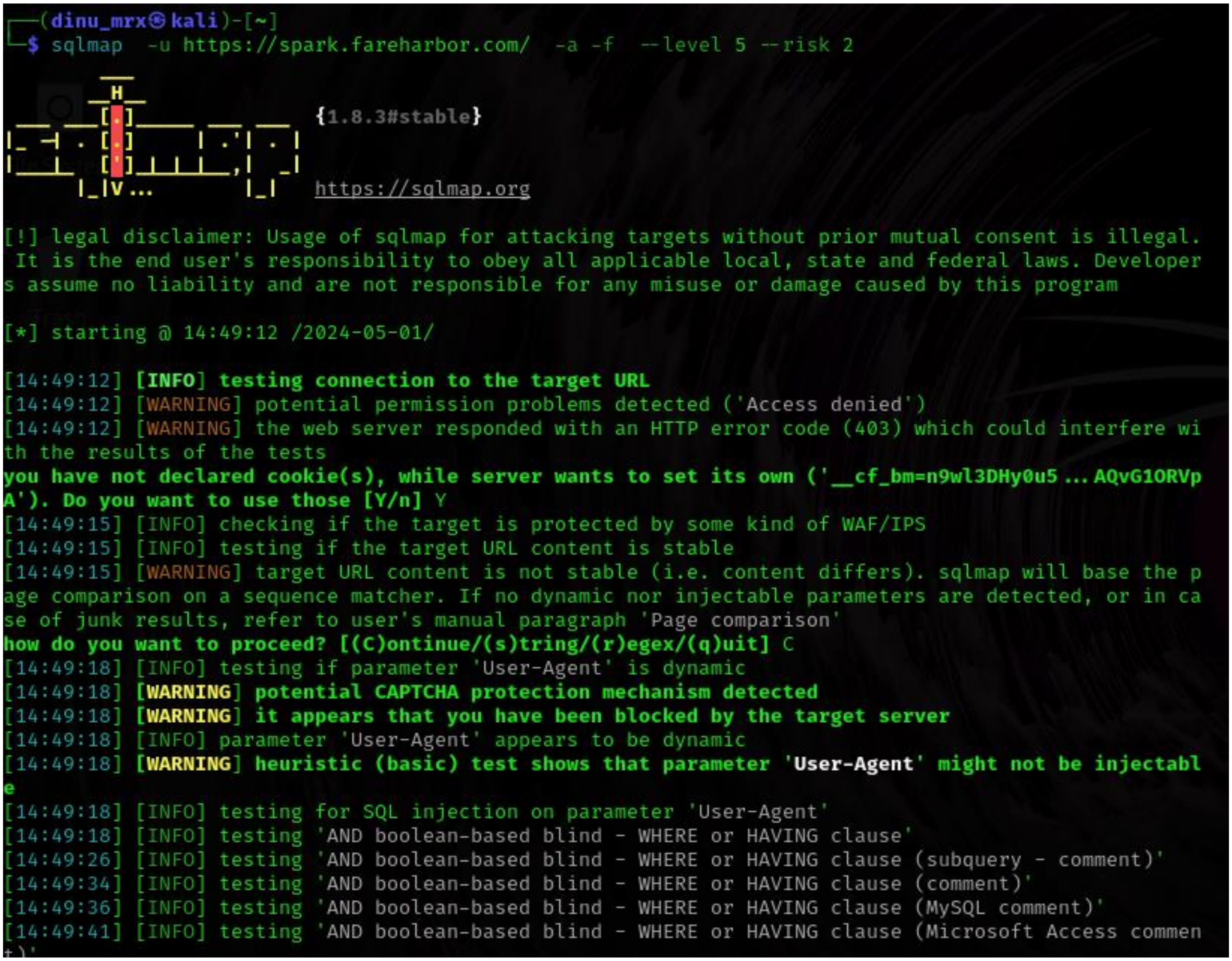
https://book.hacktricks.xyz/pentesting-web/sql-injection



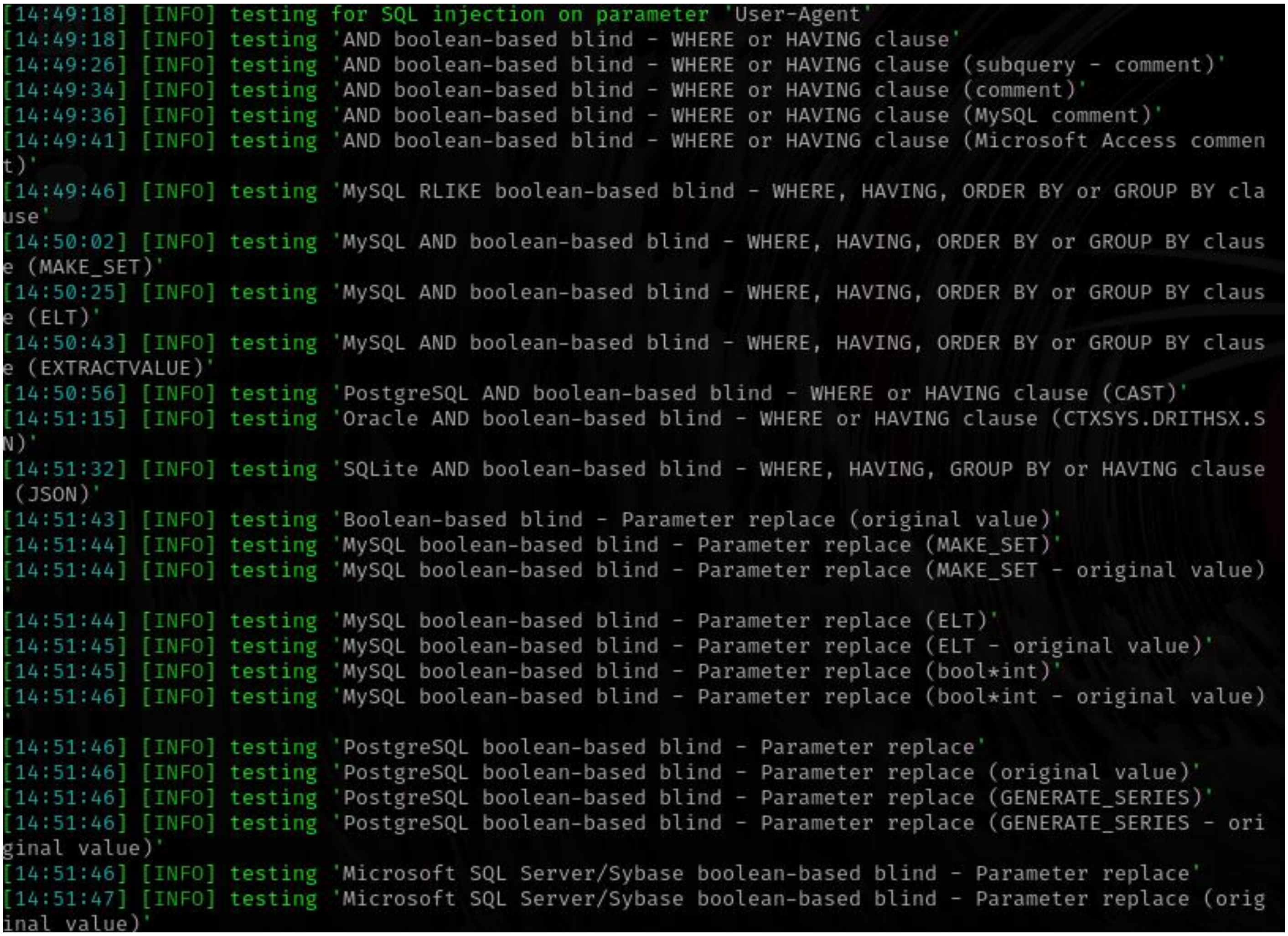
https://spark.fareharbor.com Scan Progress

Host: https://spark.fareharbor.com

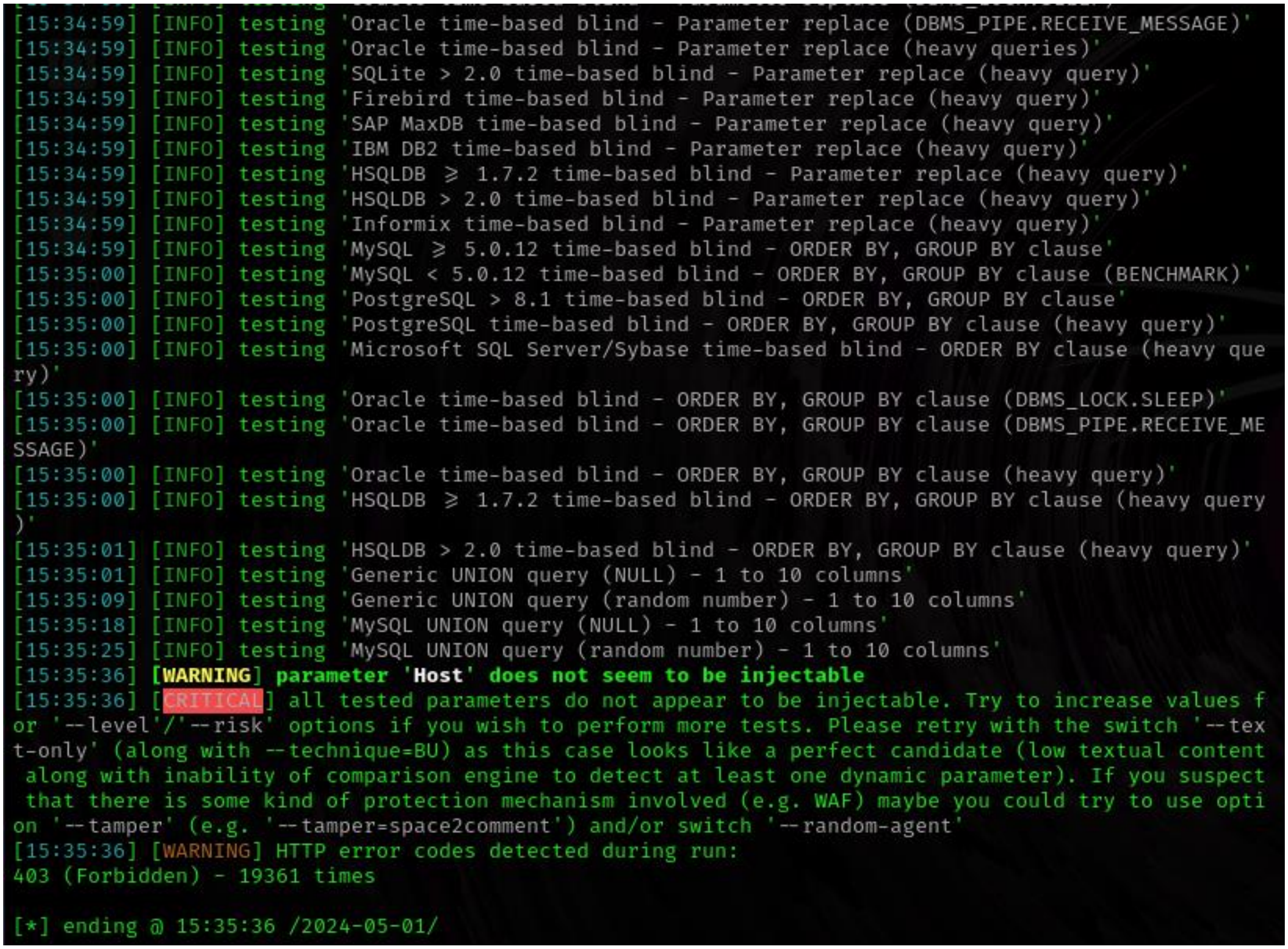
Plugin	Strength	Progress	Elapsed	Reqs	Alerts	Status
Path Traversal	Medium		10:14.401	2505	0	✓
Remote File Inclusion	Medium		07:19.179	1640	0	✓
Heartbleed OpenSSL Vulnerability	Medium		00:00.180	4	0	✓
Source Code Disclosure - /WEB-INF Folder	Medium		01:10.845	49	0	✓
Source Code Disclosure - CVE-2012-1823	Medium		01:10.841	73	0	✓
Remote Code Execution - CVE-2012-1823	Medium		00:13.880	224	0	✓
External Redirect	Medium		06:34.519	1476	0	✓
Server Side Include	Medium		02:57.791	656	0	✓
Cross Site Scripting (Reflected)	Medium		03:35.797	820	0	✓
Cross Site Scripting (Persistent) - Prime	Medium		00:48.404	164	0	✓
Cross Site Scripting (Persistent) - Spider	Medium		00:08.173	112	0	✓
Cross Site Scripting (Persistent)	Medium		00:08.330	0	0	✓
SQL Injection	Medium		14:38.425	3971	0	✓
SQL Injection - MySQL	Medium		07:16.936	1640	0	✓
SQL Injection - Hypersonic SQL	Medium		07:13.324	1640	0	✓
SQL Injection - Oracle	Medium		04:26.391	984	0	✓
SQL Injection - PostgreSQL	Medium		03:38.026	820	0	✓
SQL Injection - SQLite	Medium		06:51.726	1551	3	✓
Cross Site Scripting (DOM Based)	Medium		00:58.348	0	0	🚫
SQL Injection - MsSQL	Medium		07:16.854	1640	0	✓
Log4Shell	Medium		00:00.003	0	0	🚫
Spring4Shell	Medium		00:14.645	254	0	✓
Server Side Code Injection	Medium		05:53.213	1312	0	✓
Remote OS Command Injection	Medium		20:12.774	5740	0	✓
XPath Injection	Medium		06:42.062	492	0	✓
XML External Entity Attack	Medium		00:07.737	0	0	✓
Generic Padding Oracle	Medium		00:13.991	32	0	✓
Cloud Metadata Potentially Exposed	Medium		00:00.344	4	0	✓
Server Side Template Injection	Medium		10:27.101	2296	0	✓
Server Side Template Injection (Blind)	Medium		10:15.832	1970	0	✓
Directory Browsing	Medium		00:11.516	112	0	✓
Buffer Overflow	Medium		00:46.661	164	0	✓
Format String Error	Medium		02:21.803	492	0	✓



```
┌──(dinu_mrx㉿kali)-[~]
└─$ sqlmap  -u https://spark.fareharbor.com/  -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 user's responsibility to obey all applicable local, state and federal laws. Developer
s assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 14:49:12 /2024-05-01/

[14:49:12] [INFO] testing connection to the target URL
[14:49:12] [WARNING] potential permission problems detected ('Access denied')
[14:49:12] [WARNING] the web server responded with an HTTP error code (403) which could interfere wi
th the results of the tests
you have not declared cookie(s), while server wants to set its own ('__cf_bm=n9wl3DHy0u5...AQvG1ORVp
A'). Do you want to use those [Y/n] Y
[14:49:15] [INFO] checking if the target is protected by some kind of WAF/IPS
[14:49:15] [INFO] testing if the target URL content is stable
[14:49:15] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the p
age comparison on a sequence matcher. If no dynamic nor injectable parameters are detected, or in ca
se 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
[14:49:18] [INFO] testing if parameter 'User-Agent' is dynamic
[14:49:18] [WARNING] potential CAPTCHA protection mechanism detected
[14:49:18] [WARNING] it appears that you have been blocked by the target server
[14:49:18] [INFO] parameter 'User-Agent' appears to be dynamic
[14:49:18] [WARNING] heuristic (basic) test shows that parameter 'User-Agent' might not be injectabl
e
[14:49:18] [INFO] testing for SQL injection on parameter 'User-Agent'
[14:49:18] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[14:49:26] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (subquery - comment)'
[14:49:34] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (comment)'
[14:49:36] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[14:49:41] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Microsoft Access commen
t)'
```



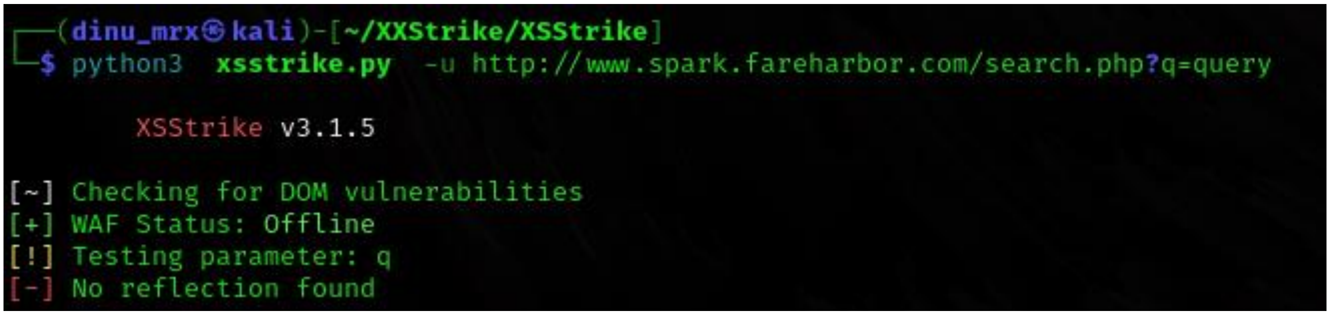
```
[14:49:18] [INFO] testing for SQL injection on parameter 'User-Agent'
[14:49:18] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[14:49:26] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (subquery - comment)'
[14:49:34] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (comment)'
[14:49:36] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[14:49:41] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Microsoft Access commen
t)'
[14:49:46] [INFO] testing 'MySQL RLIKE boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY cla
use'
[14:50:02] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY claus
e (MAKE_SET)'
[14:50:25] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY claus
e (ELT)'
[14:50:43] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY claus
e (EXTRACTVALUE)'
[14:50:56] [INFO] testing 'PostgreSQL AND boolean-based blind - WHERE or HAVING clause (CAST)'
[14:51:15] [INFO] testing 'Oracle AND boolean-based blind - WHERE or HAVING clause (CTXSYS.DRITHSX.S
N)'
[14:51:32] [INFO] testing 'SQLite AND boolean-based blind - WHERE, HAVING, GROUP BY or HAVING clause
 (JSON)'
[14:51:43] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[14:51:44] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET)'
[14:51:44] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET - original value)
'
[14:51:44] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT)'
[14:51:45] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT - original value)'
[14:51:45] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int)'
[14:51:46] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int - original value)
'
[14:51:46] [INFO] testing 'PostgreSQL boolean-based blind - Parameter replace'
[14:51:46] [INFO] testing 'PostgreSQL boolean-based blind - Parameter replace (original value)'
[14:51:46] [INFO] testing 'PostgreSQL boolean-based blind - Parameter replace (GENERATE_SERIES)'
[14:51:46] [INFO] testing 'PostgreSQL boolean-based blind - Parameter replace (GENERATE_SERIES - ori
ginal value)'
[14:51:46] [INFO] testing 'Microsoft SQL Server/Sybase boolean-based blind - Parameter replace'
[14:51:47] [INFO] testing 'Microsoft SQL Server/Sybase boolean-based blind - Parameter replace (orig
inal value)'
```



```
[15:34:59] [INFO] testing 'Oracle time-based blind - Parameter replace (DBMS_PIPE.RECEIVE_MESSAGE)'
[15:34:59] [INFO] testing 'Oracle time-based blind - Parameter replace (heavy queries)'
[15:34:59] [INFO] testing 'SQLite > 2.0 time-based blind - Parameter replace (heavy query)'
[15:34:59] [INFO] testing 'Firebird time-based blind - Parameter replace (heavy query)'
[15:34:59] [INFO] testing 'SAP MaxDB time-based blind - Parameter replace (heavy query)'
[15:34:59] [INFO] testing 'IBM DB2 time-based blind - Parameter replace (heavy query)'
[15:34:59] [INFO] testing 'HSQLDB ≥ 1.7.2 time-based blind - Parameter replace (heavy query)'
[15:34:59] [INFO] testing 'HSQLDB > 2.0 time-based blind - Parameter replace (heavy query)'
[15:34:59] [INFO] testing 'Informix time-based blind - Parameter replace (heavy query)'
[15:34:59] [INFO] testing 'MySQL ≥ 5.0.12 time-based blind - ORDER BY, GROUP BY clause'
[15:35:00] [INFO] testing 'MySQL < 5.0.12 time-based blind - ORDER BY, GROUP BY clause (BENCHMARK)'
[15:35:00] [INFO] testing 'PostgreSQL > 8.1 time-based blind - ORDER BY, GROUP BY clause'
[15:35:00] [INFO] testing 'PostgreSQL time-based blind - ORDER BY, GROUP BY clause (heavy query)'
[15:35:00] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind - ORDER BY clause (heavy que
ry)'
[15:35:00] [INFO] testing 'Oracle time-based blind - ORDER BY, GROUP BY clause (DBMS_LOCK.SLEEP)'
[15:35:00] [INFO] testing 'Oracle time-based blind - ORDER BY, GROUP BY clause (DBMS_PIPE.RECEIVE_ME
SSAGE)'
[15:35:00] [INFO] testing 'Oracle time-based blind - ORDER BY, GROUP BY clause (heavy query)'
[15:35:00] [INFO] testing 'HSQLDB ≥ 1.7.2 time-based blind - ORDER BY, GROUP BY clause (heavy query
)'
[15:35:01] [INFO] testing 'HSQLDB > 2.0 time-based blind - ORDER BY, GROUP BY clause (heavy query)'
[15:35:01] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[15:35:09] [INFO] testing 'Generic UNION query (random number) - 1 to 10 columns'
[15:35:18] [INFO] testing 'MySQL UNION query (NULL) - 1 to 10 columns'
[15:35:25] [INFO] testing 'MySQL UNION query (random number) - 1 to 10 columns'
[15:35:36] [WARNING] parameter 'Host' does not seem to be injectable
[15:35:36] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values f
or '--level'/'--risk' options if you wish to perform more tests. Please retry with the switch '--tex
t-only' (along with --technique=BU) as this case looks like a perfect candidate (low textual content
 along with inability 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 could try to use opti
on '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
[15:35:36] [WARNING] HTTP error codes detected during run:
403 (Forbidden) - 19361 times

[*] ending @ 15:35:36 /2024-05-01/
```

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



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

        XSStrike v3.1.5

[~] Checking for DOM vulnerabilities
[+] WAF Status: Offline
[!] Testing parameter: q
[-] No reflection found
```

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