Sri Lanka Institute of Information Technology

Bug Bounty - Report 08

Absence of Anti-CSRF Tokens

Supabase.com

Student Name – Wanasinghe N.K

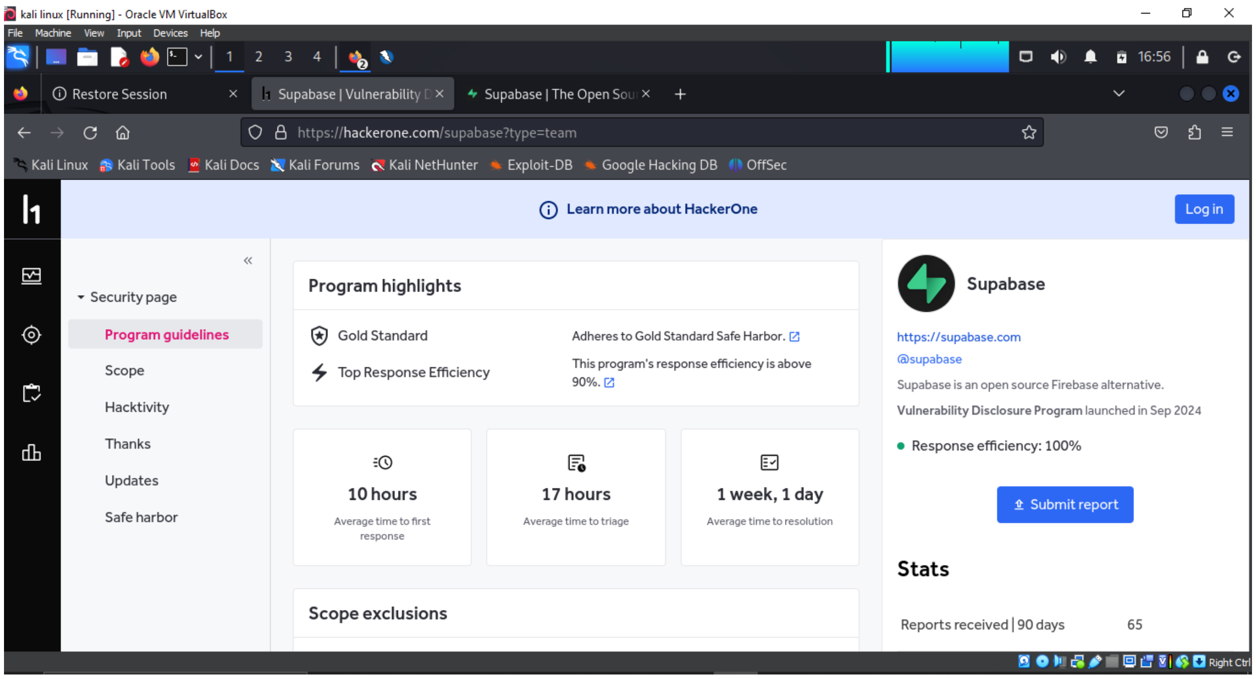
Student ID – IT23221000

IE2062 - Web Security

B.Sc. (Hons) in information Technology Specializing in Cyber Security

Report 08 – supabase.com (Hackerone)

Main domain – https://supabase.com/


kali linux [Running] - Oracle VM VirtualBox
Restore Session
Supabase | Vulnerability D
Supabase | The Open Sou
https://hackerone.com/supabase?type=team
Learn more about HackerOne
Log in
Security page
Program guidelines
Scope
Hacktivity
Thanks
Updates
Safe harbor
Submit report

Program highlights

Gold Standard	Adheres to Gold Standard Safe Harbor.
Top Response Efficiency	This program's response efficiency is above 90%.

10 hours	17 hours	1 week, 1 day
Average time to first response	Average time to triage	Average time to resolution

Scope exclusions

Supabase

https://supabase.com

@supabase

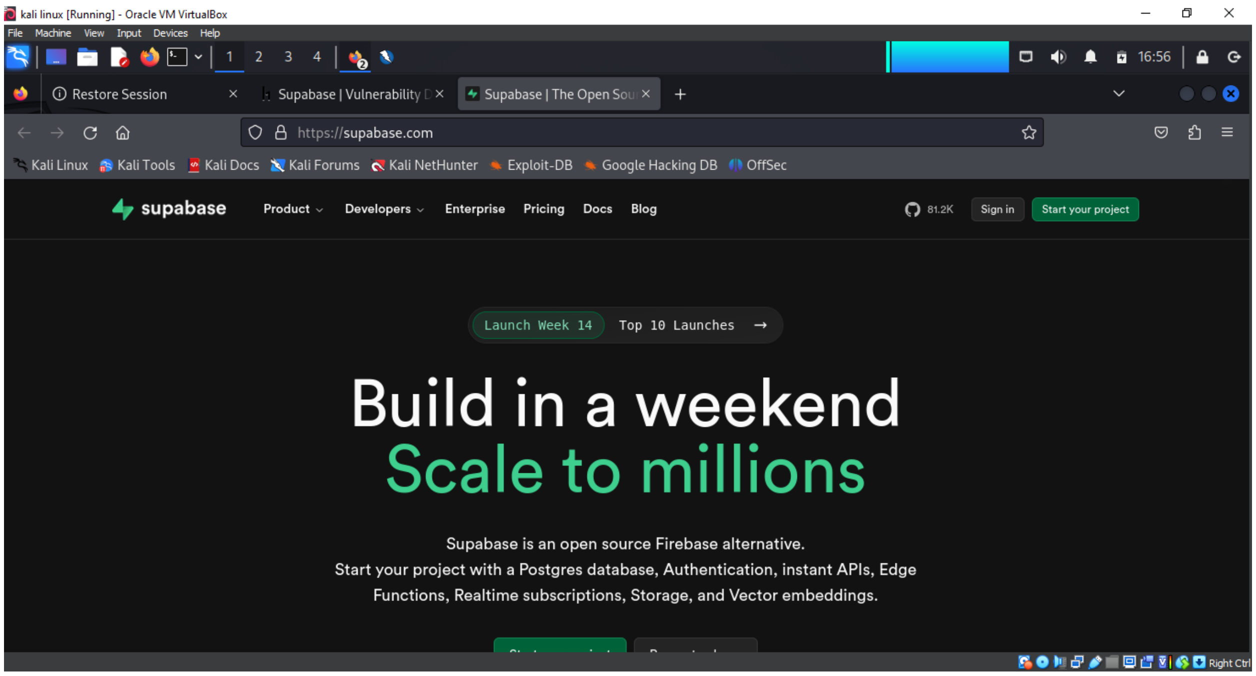
Supabase is an open source Firebase alternative.

Vulnerability Disclosure Program launched in Sep 2024

● Response efficiency: 100%

Stats

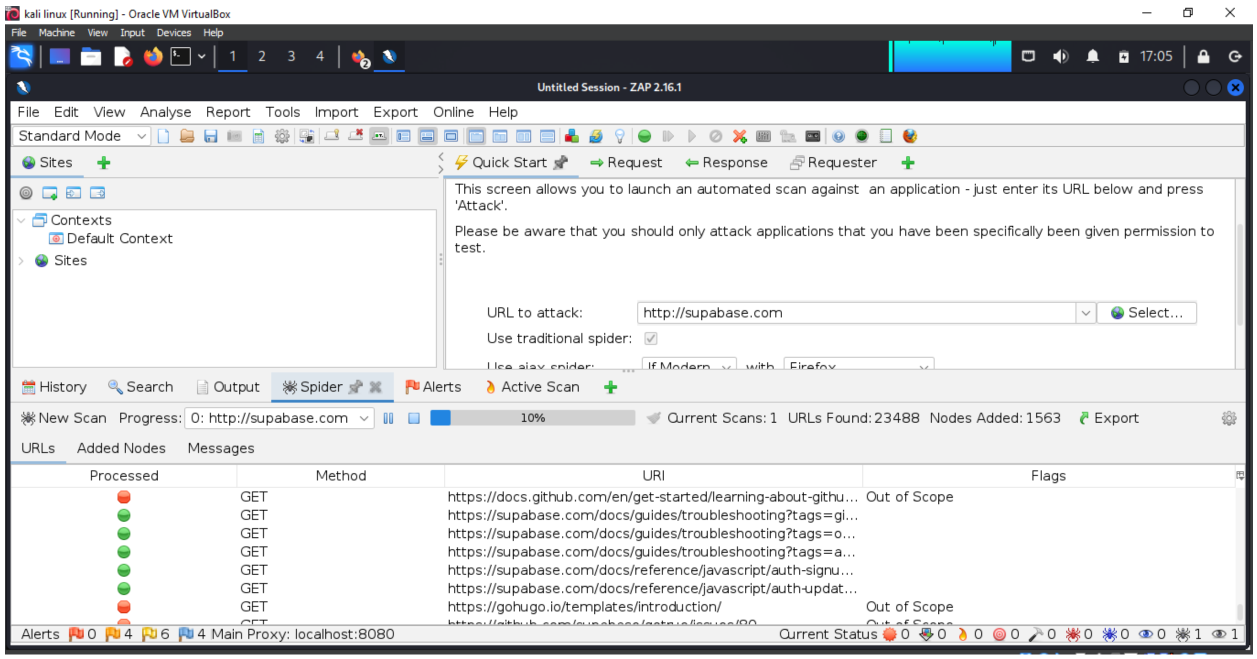
Reports received | 90 days 65



Build in a weekend
Scale to millions

Supabase is an open source Firebase alternative.
Start your project with a Postgres database, Authentication, instant APIs, Edge Functions, Realtime subscriptions, Storage, and Vector embeddings.

I used OWASP ZAP tool to scan the website and found all the in scope and out of scope domains.


kali linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Untitled Session - ZAP 2.16.1
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Quick Start
Request
Response
Requester
Contexts
Default Context
Sites
History Search Output Spider Alerts Active Scan
URLs Added Nodes Messages

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: http://supabase.com

Use traditional spider: ✓

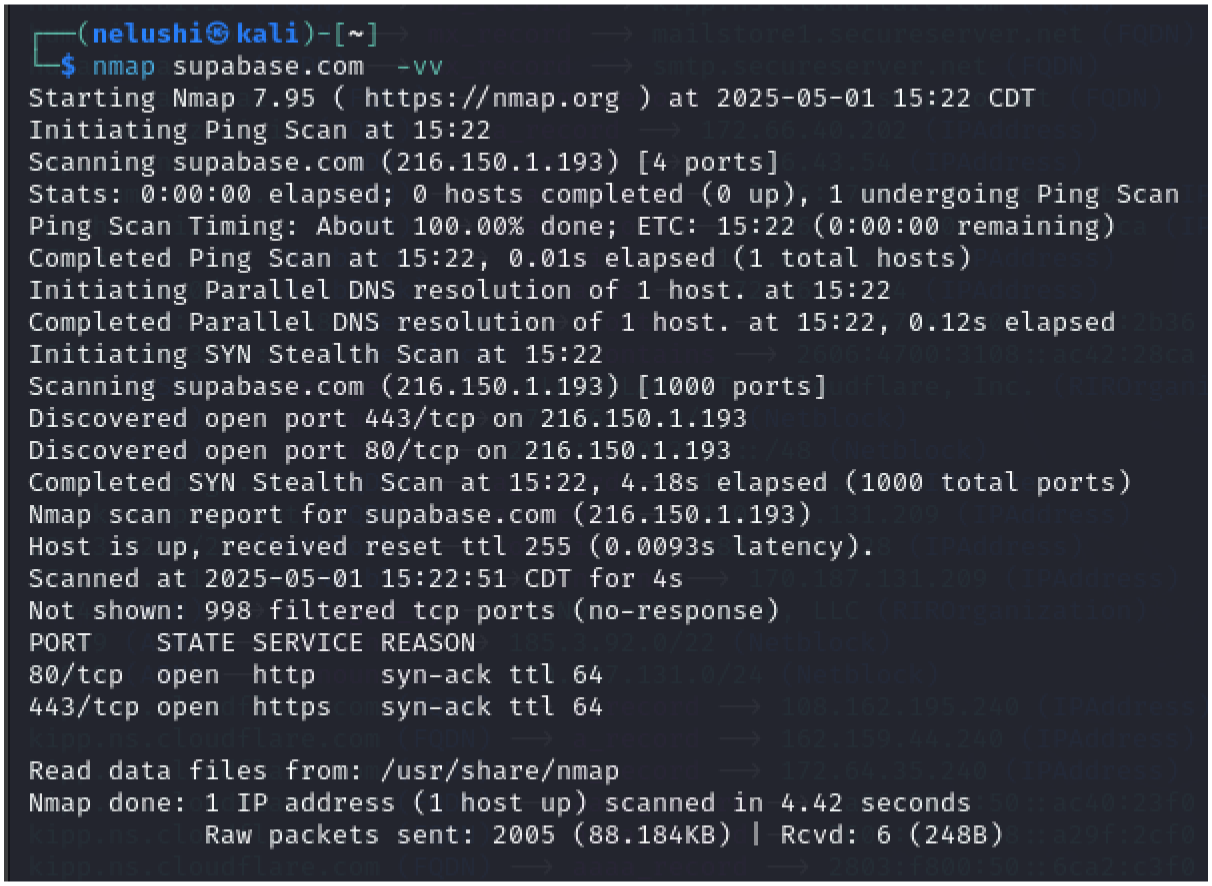
New Scan Progress: 0: http://supabase.com 10% Current Scans: 1 URLs Found: 23488 Nodes Added: 1563 Export

Processed	Method	URI	Flags
🔴	GET	https://docs.github.com/en/get-started/learning-about-githu…	Out of Scope
🟢	GET	https://supabase.com/docs/guides/troubleshooting?tags=gi…	
🟢	GET	https://supabase.com/docs/guides/troubleshooting?tags=o…	
🟢	GET	https://supabase.com/docs/guides/troubleshooting?tags=a…	
🟢	GET	https://supabase.com/docs/reference/javascript/auth-signu…	
🟢	GET	https://supabase.com/docs/reference/javascript/auth-updat…	
🔴	GET	https://gohugo.io/templates/introduction/	Out of Scope

Alerts 0 4 6 4 Main Proxy: localhost:8080 Current Status 0 0 0 0 0 0 0 0 1 1

Nmap – Network scanning and enumeration

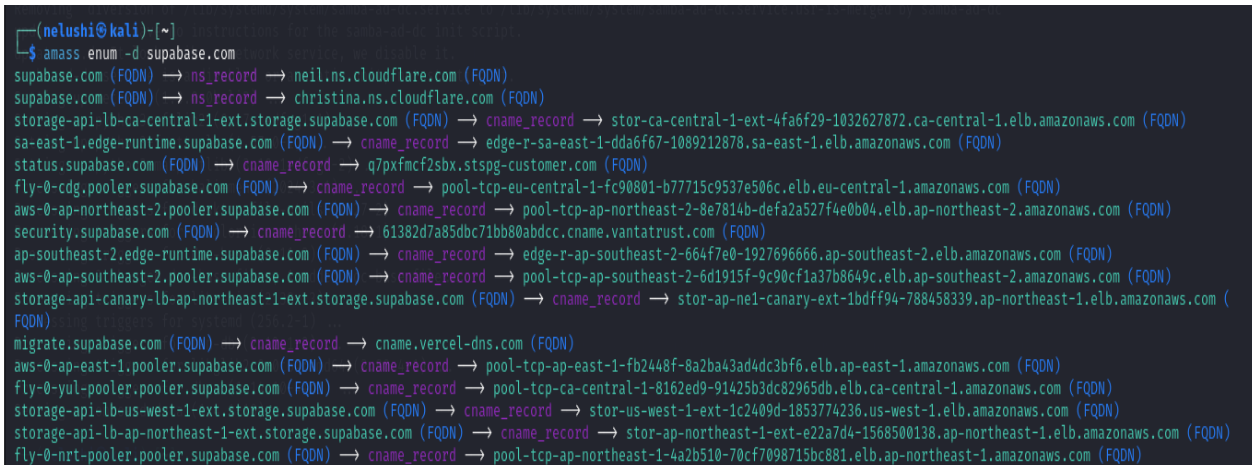
I found all the open ports and detected the running services on the target server using Nmap.



```
┌──(nelushi㉿kali)-[~]
└─$ nmap supabase.com  -vv
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-01 15:22 CDT
Initiating Ping Scan at 15:22
Scanning supabase.com (216.150.1.193) [4 ports]
Stats: 0:00:00 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Ping Scan Timing: About 100.00% done; ETC: 15:22 (0:00:00 remaining)
Completed Ping Scan at 15:22, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 15:22
Completed Parallel DNS resolution of 1 host. at 15:22, 0.12s elapsed
Initiating SYN Stealth Scan at 15:22
Scanning supabase.com (216.150.1.193) [1000 ports]
Discovered open port 443/tcp on 216.150.1.193
Discovered open port 80/tcp on 216.150.1.193
Completed SYN Stealth Scan at 15:22, 4.18s elapsed (1000 total ports)
Nmap scan report for supabase.com (216.150.1.193)
Host is up, received reset ttl 255 (0.0093s latency).
Scanned at 2025-05-01 15:22:51 CDT for 4s
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE REASON
80/tcp  open  http    syn-ack ttl 64
443/tcp open  https   syn-ack ttl 64

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 4.42 seconds
           Raw packets sent: 2005 (88.184KB) | Rcvd: 6 (248B)
```

Amass – Subdomain and DNS mapping

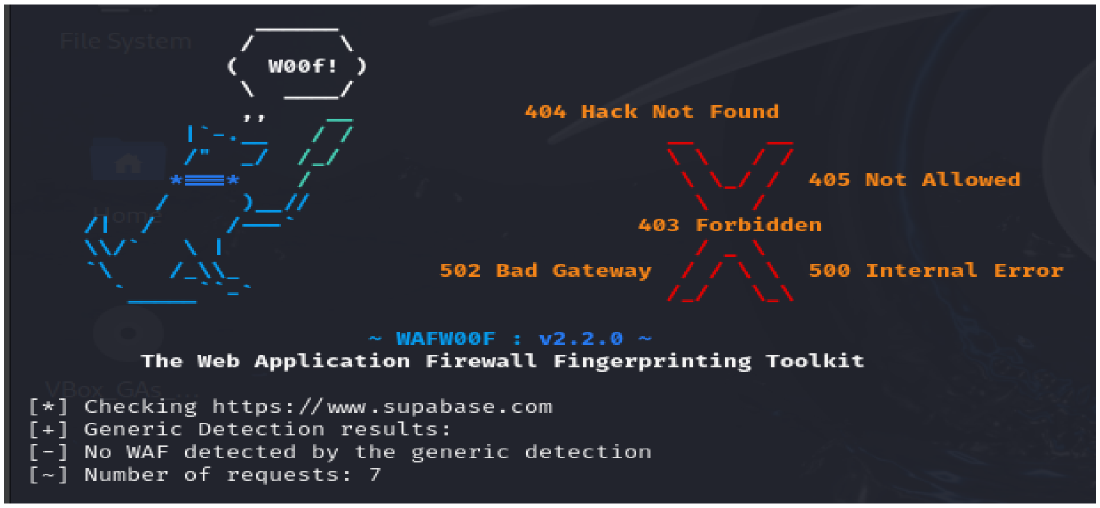
I found all the subdomains related to the target domain using Amass.



```
┌──(nelushi㉿kali)-[~]
└─$ amass enum -d supabase.com
supabase.com (FQDN) ⟶ ns_record ⟶ neil.ns.cloudflare.com (FQDN)
supabase.com (FQDN) ⟶ ns_record ⟶ christina.ns.cloudflare.com (FQDN)
storage-api-lb-ca-central-1-ext.storage.supabase.com (FQDN) ⟶ cname_record ⟶ stor-ca-central-1-ext-4fa6f29-1032627872.ca-central-1.elb.amazonaws.com (FQDN)
sa-east-1.edge-runtime.supabase.com (FQDN) ⟶ cname_record ⟶ edge-r-sa-east-1-dda6f67-1089212878.sa-east-1.elb.amazonaws.com (FQDN)
status.supabase.com (FQDN) ⟶ cname_record ⟶ q7pxfmcf2sbx.stspg-customer.com (FQDN)
fly-0-cdg.pooler.supabase.com (FQDN) ⟶ cname_record ⟶ pool-tcp-eu-central-1-fc90801-b77715c9537e506c.elb.eu-central-1.amazonaws.com (FQDN)
aws-0-ap-northeast-2.pooler.supabase.com (FQDN) ⟶ cname_record ⟶ pool-tcp-ap-northeast-2-8e7814b-defa2a527f4e0b04.elb.ap-northeast-2.amazonaws.com (FQDN)
security.supabase.com (FQDN) ⟶ cname_record ⟶ 61382d7a85dbc71bb80abdcc.cname.vantatrust.com (FQDN)
ap-southeast-2.edge-runtime.supabase.com (FQDN) ⟶ cname_record ⟶ edge-r-ap-southeast-2-664f7e0-1927696666.ap-southeast-2.elb.amazonaws.com (FQDN)
aws-0-ap-southeast-2.pooler.supabase.com (FQDN) ⟶ cname_record ⟶ pool-tcp-ap-southeast-2-6d1915f-9c90cf1a37b8649c.elb.ap-southeast-2.amazonaws.com (FQDN)
storage-api-canary-lb-ap-northeast-1-ext.storage.supabase.com (FQDN) ⟶ cname_record ⟶ stor-ap-ne1-canary-ext-1bdff94-788458339.ap-northeast-1.elb.amazonaws.com (
FQDN)
migrate.supabase.com (FQDN) ⟶ cname_record ⟶ cname.vercel-dns.com (FQDN)
aws-0-ap-east-1.pooler.supabase.com (FQDN) ⟶ cname_record ⟶ pool-tcp-ap-east-1-fb2448f-8a2ba43ad4dc3bf6.elb.ap-east-1.amazonaws.com (FQDN)
fly-0-yul-pooler.pooler.supabase.com (FQDN) ⟶ cname_record ⟶ pool-tcp-ca-central-1-8162ed9-91425b3dc82965db.elb.ca-central-1.amazonaws.com (FQDN)
storage-api-lb-us-west-1-ext.storage.supabase.com (FQDN) ⟶ cname_record ⟶ stor-us-west-1-ext-1c2409d-1853774236.us-west-1.elb.amazonaws.com (FQDN)
storage-api-lb-ap-northeast-1-ext.storage.supabase.com (FQDN) ⟶ cname_record ⟶ stor-ap-northeast-1-ext-e22a7d4-1568500138.ap-northeast-1.elb.amazonaws.com (FQDN)
fly-0-nrt-pooler.pooler.supabase.com (FQDN) ⟶ cname_record ⟶ pool-tcp-ap-northeast-1-4a2b510-70cf7098715bc881.elb.ap-northeast-1.amazonaws.com (FQDN)
```

Wafw00f – Firewall Detection

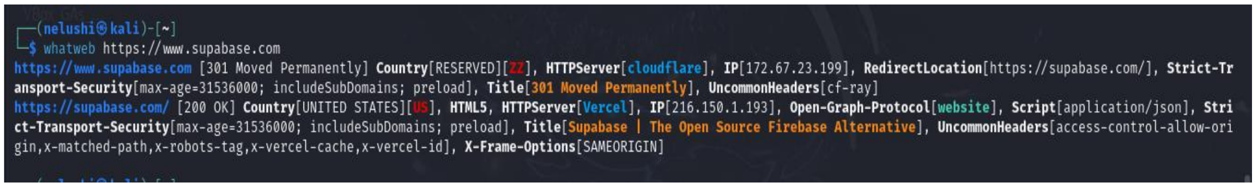
Command used – wafw00f https://www.supabase.com/



```
                ( W00f! )

                                     404 Hack Not Found

                                                  405 Not Allowed

                                        403 Forbidden

                              502 Bad Gateway          500 Internal Error

                       ~ WAFW00F : v2.2.0 ~
        The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://www.supabase.com
[+] Generic Detection results:
[-] No WAF detected by the generic detection
[~] Number of requests: 7
```

Whatweb – to identify the technologies used by the site.

Commans used – whatweb https://www.supabase.com/



```
┌──(nelushi㉿kali)-[~]
└─$ whatweb https://www.supabase.com
https://www.supabase.com [301 Moved Permanently] Country[RESERVED][ZZ], HTTPServer[cloudflare], IP[172.67.23.199], RedirectLocation[https://supabase.com/], Strict-Tr
ansport-Security[max-age=31536000; includeSubDomains; preload], Title[301 Moved Permanently], UncommonHeaders[cf-ray]
https://supabase.com/ [200 OK] Country[UNITED STATES][US], HTML5, HTTPServer[Vercel], IP[216.150.1.193], Open-Graph-Protocol[website], Script[application/json], Stri
ct-Transport-Security[max-age=31536000; includeSubDomains; preload], Title[Supabase | The Open Source Firebase Alternative], UncommonHeaders[access-control-allow-ori
gin,x-matched-path,x-robots-tag,x-vercel-cache,x-vercel-id], X-Frame-Options[SAMEORIGIN]
```

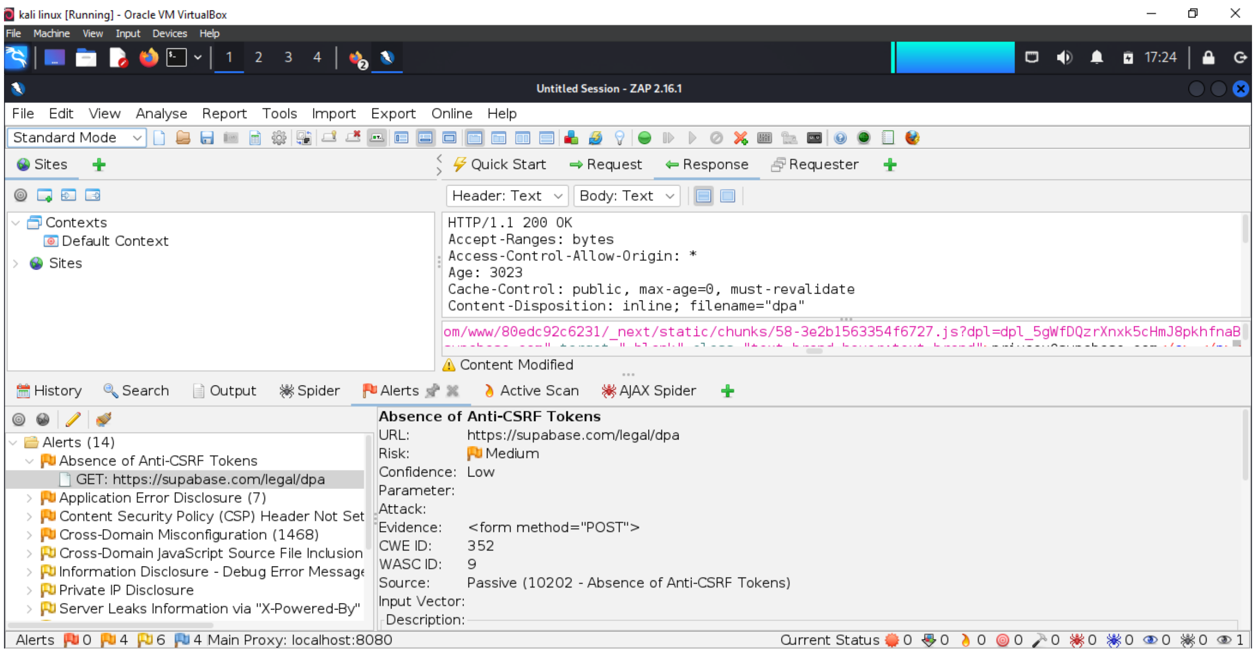
Vulnerability 01

Domain

Login page of supabase.com

Vulnerability title

Absence of Anti-CSRF Tokens


kali linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Untitled Session - ZAP 2.16.1
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Quick Start
Request
Response
Requester
Header: Text
Body: Text
Contexts
Default Context
Sites
History
Search
Output
Spider
Alerts
Active Scan
AJAX Spider

```
HTTP/1.1 200 OK
Accept-Ranges: bytes
Access-Control-Allow-Origin: *
Age: 3023
Cache-Control: public, max-age=0, must-revalidate
Content-Disposition: inline; filename="dpa"
```

```
om/www/80edc92c6231/_next/static/chunks/58-3e2b1563354f6727.js?dpl=dpl_5gWfDQzrXnxk5cHmJ8pkhfnaB
```

Content Modified

Alerts (14)
- Absence of Anti-CSRF Tokens
  - GET: https://supabase.com/legal/dpa
- Application Error Disclosure (7)
- Content Security Policy (CSP) Header Not Set
- Cross-Domain Misconfiguration (1468)
- Cross-Domain JavaScript Source File Inclusion
- Information Disclosure - Debug Error Message
- Private IP Disclosure
- Server Leaks Information via "X-Powered-By"

Absence of Anti-CSRF Tokens

URL:	https://supabase.com/legal/dpa
Risk:	Medium
Confidence:	Low
Parameter:	
Attack:	
Evidence:	<form method="POST">
CWE ID:	352
WASC ID:	9
Source:	Passive (10202 - Absence of Anti-CSRF Tokens)
Input Vector:	
Description:	

Alerts 0 4 6 4 Main Proxy: localhost:8080 Current Status 0 0 0 0 0 0 0 0 0 1

Vulnerability description

Missing Anti-CSRF tokens were found in a HTML submission form.

Cross-Site Request Forgery (CSRF) is a type of attack where the victim is misled into sending an HTTP request to a target website without the victim's intent. In here, the malicious request forces the victim’s browser to perform an action on a website where the victim is already authenticated.

The vulnerability is that an application allows predictable and repeatable URL or form actions to be executed that can lead to unauthorized action being taken to exploit what a website trusts the user to do.

CSRF attacks are notable because they demonstrate a weakness in the trust a site has for the user rather than how Cross-Site Scripting (XSS) exploits trust a user has for a site.

When Are CSRF Attacks Effective?

- CSRF attacks are effective when the victim has an active session on the target website.
- The victim is authenticated via HTTP auth on the target site.
- The victim is on the same local network as the target site.

An Attacker can exploit the active session of the victim to perform unauthorized actions, such as transferring money, changing account settings, without the victim being aware**.**

What are the Affected components?

- Web Forms: Forms that can do state-changing actions without anti-CSRF tokens.

Reflection – I didn't find any input tag like <input type="hidden" name="csrf_token" value="..."> in the source code, which means the page may indicate a **missing CSRF token**, which is a vulnerability.

Sensitive actions can be submitted without origin validation.

Impact assessment

Severity – Medium

- On the Victim's Behalf , an attacker may,

  Alter account settings, such as email or password

  Make unauthorized purchases

  Transfer money into online banking

  Delete or change sensitive information

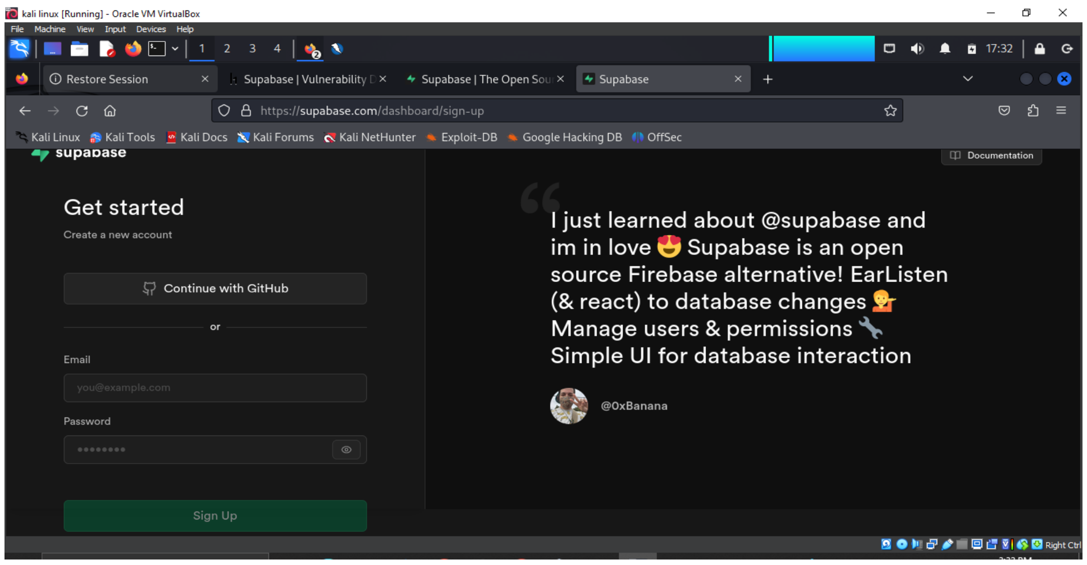
  Send emails/messages from their account.

- Exploit User Trust - Take advantage of the site's trust in the user's browser to perform malicious actions.
- Take Control of the User's Account - Change account settings to lock the legitimate user out.
- Perform Malicious Actions in the Background - Submit forms or change configuration without the victim's awareness.

Steps to reproduce with Proof of Concept (poc)

1. First, I Inspected the Target Website

- I Opened the target website in my browser. (the login page of supabase.com)
- I Focused on the login form and I logged in and now my session is active.


kali linux [Running] - Oracle VM VirtualBox

https://supabase.com/dashboard/sign-up

Get started

Create a new account

“I just learned about @supabase and im in love 😍 Supabase is an open source Firebase alternative! EarListen (& react) to database changes 💁 Manage users & permissions 🔧 Simple UI for database interaction

@0xBanana

2. I captured the request and intercepted it via **burp suite** to check if this site uses CSRF tokens.

If a sire uses CSRF tokens, it might show like this.

Most common one is looking for tokens **in the Request Body**

If it’s a form submission (POST request), we might see something like,

```
POST /profile/update HTTP/1.1
Host: spabase.com
Content-Type: application/x-www-form-urlencoded
name=JohnDoe&email=john@example.com&csrf_token=9f3a5b2adf9e4a...
```

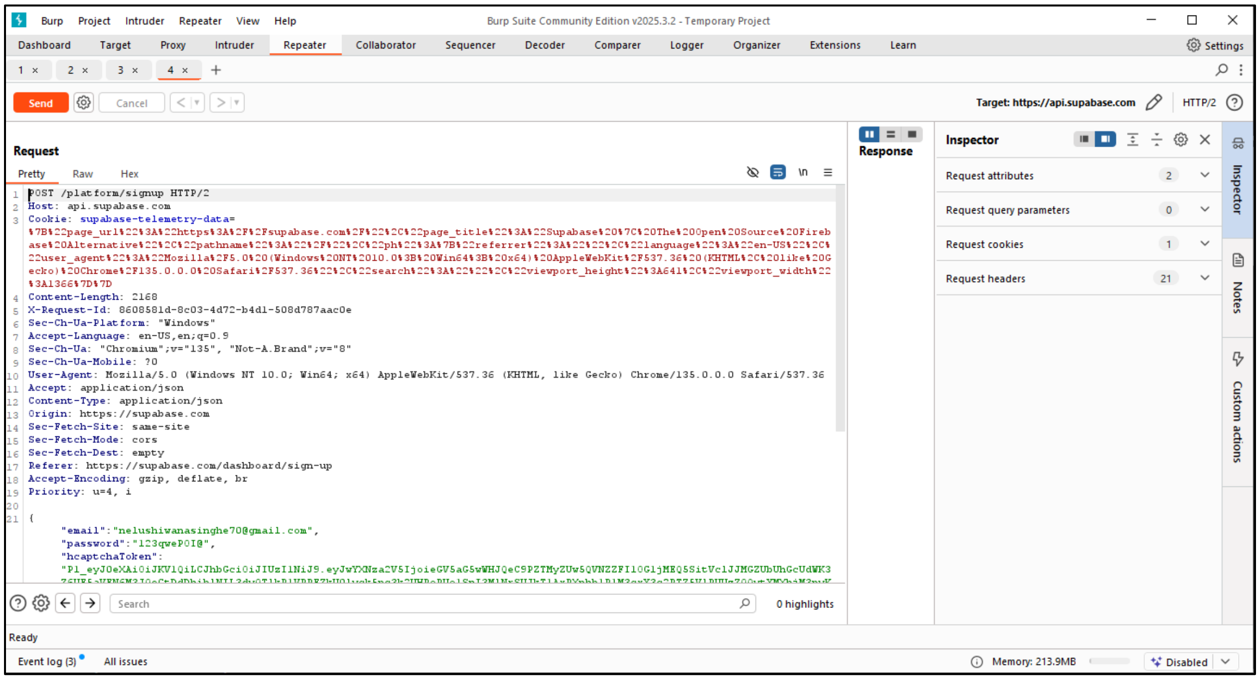
In HTTP Headers

Somet frameworks like Angular, Django send CSRF tokens in custom headers,

```
POST /settings/update HTTP/1.1
Host: spabase.com
X-CSRF-Token: 9f3a5b2adf9e4a...
```

Look for headers like,

- X-CSRF-Token
- X-XSRF-TOKEN
- X-CSRF

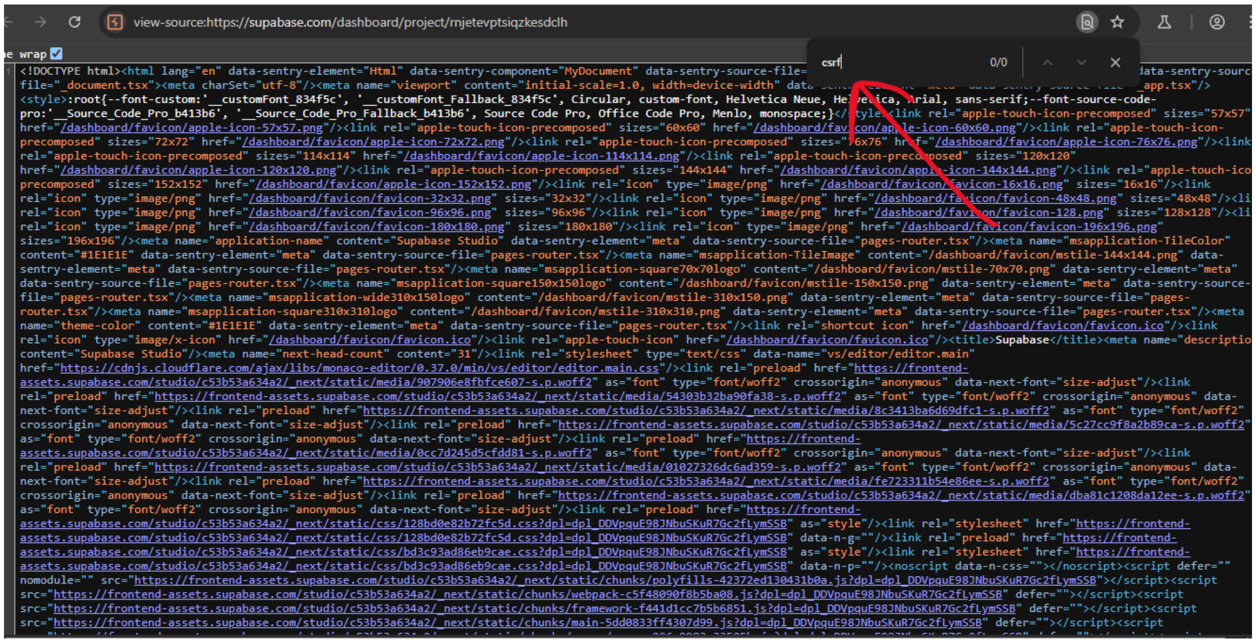

Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.3.2 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Send Cancel
Target: https://api.supabase.com HTTP/2
Request Pretty Raw Hex
Response
Inspector
Request attributes 2
Request query parameters 0
Request cookies 1
Request headers 21
Notes Custom actions
Search 0 highlights
Ready
Event log (3) All issues
Memory: 213.9MB Disabled

```
POST /platform/signup HTTP/2
Host: api.supabase.com
Cookie: supabase-telemetry-data=
%7B%22page_url%22%3A%22https%3A%2F%2Fsupabase.com%2F%22%2C%22page_title%22%3A%22Supabase%20%7C%20The%20Open%20Source%20Fireb
ase%20Alternative%22%2C%22pathname%22%3A%22%2F%22%2C%22ph%22%3A%7B%22referrer%22%3A%22%22%2C%22language%22%3A%22en-US%22%2C%
22user_agent%22%3A%22Mozilla%2F5.0%20(Windows%20NT%2010.0%3B%20Win64%3B%20x64)%20AppleWebKit%2F537.36%20(KHTML%2C%20like%20G
ecko)%20Chrome%2F135.0.0.0%20Safari%2F537.36%22%2C%22search%22%3A%22%22%2C%22viewport_height%22%3A641%2C%22viewport_width%22
%3A1366%7D%7D
Content-Length: 2168
X-Request-Id: 8608581d-8c03-4d72-b4d1-508d787aac0e
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Sec-Ch-Ua: "Chromium";v="135", "Not-A.Brand";v="8"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
Accept: application/json
Content-Type: application/json
Origin: https://supabase.com
Sec-Fetch-Site: same-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://supabase.com/dashboard/sign-up
Accept-Encoding: gzip, deflate, br
Priority: u=4, i

{
    "email":"nelushiwanasinghe70@gmail.com",
    "password":"123qweP0I@",
    "hcaptchaToken":
    "P1_eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJwYXNza2V5IjoieGV5aG5wWHJQeC9PZTMyZUw5QVNZZFI1OG1jMEQ5SitVclJJMGZUbUhGcUdWK3
```

Typically, CSRF tokens are stored in a hidden input field in a form like this,

<input type="hidden" name="csrf_token" value="RANDOMCSRFVALUE">

So, I searched for CSRF keyword in the source code and I got 0 matches.

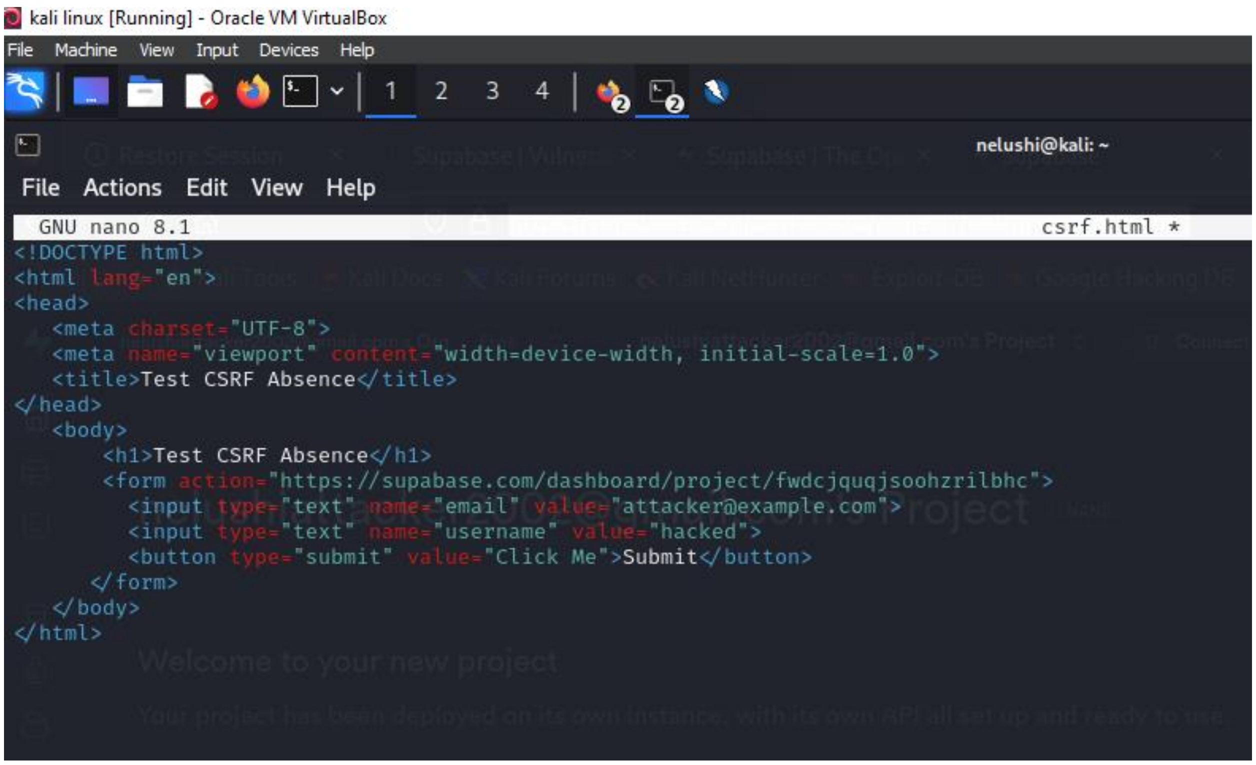

view-source:https://supabase.com/dashboard/project/rnjetevptsiqzkesdclh
wrap
csrf
0/0

```
<!DOCTYPE html><html lang="en" data-sentry-element="Html" data-sentry-component="MyDocument" data-sentry-source-file= data-sentry-sourc
file="_document.tsx"><meta charSet="utf-8"/><meta name="viewport" content="initial-scale=1.0, width=device-width" data _app.tsx"/>
<style>:root{--font-custom:'__customFont_834f5c', '__customFont_Fallback_834f5c', Circular, custom-font, Helvetica Neue, Helvetica, Arial, sans-serif;--font-source-code-
pro:'__Source_Code_Pro_b413b6', '__Source_Code_Pro_Fallback_b413b6', Source Code Pro, Office Code Pro, Menlo, monospace;}</style><link rel="apple-touch-icon-precomposed" sizes="57x57"
href="/dashboard/favicon/apple-icon-57x57.png"/><link rel="apple-touch-icon-precomposed" sizes="60x60" href="/dashboard/favicon/apple-icon-60x60.png"/><link rel="apple-touch-icon-
precomposed" sizes="72x72" href="/dashboard/favicon/apple-icon-72x72.png"/><link rel="apple-touch-icon-precomposed" sizes="76x76" href="/dashboard/favicon/apple-icon-76x76.png"/><link
rel="apple-touch-icon-precomposed" sizes="114x114" href="/dashboard/favicon/apple-icon-114x114.png"/><link rel="apple-touch-icon-precomposed" sizes="120x120"
href="/dashboard/favicon/apple-icon-120x120.png"/><link rel="apple-touch-icon-precomposed" sizes="144x144" href="/dashboard/favicon/apple-icon-144x144.png"/><link rel="apple-touch-ico
precomposed" sizes="152x152" href="/dashboard/favicon/apple-icon-152x152.png"/><link rel="icon" type="image/png" href="/dashboard/favicon/favicon-16x16.png" sizes="16x16"/><link
rel="icon" type="image/png" href="/dashboard/favicon/favicon-32x32.png" sizes="32x32"/><link rel="icon" type="image/png" href="/dashboard/favicon/favicon-48x48.png" sizes="48x48"/><li
rel="icon" type="image/png" href="/dashboard/favicon/favicon-96x96.png" sizes="96x96"/><link rel="icon" type="image/png" href="/dashboard/favicon/favicon-128.png" sizes="128x128"/><li
rel="icon" type="image/png" href="/dashboard/favicon/favicon-180x180.png" sizes="180x180"/><link rel="icon" type="image/png" href="/dashboard/favicon/favicon-196x196.png"
sizes="196x196"/><meta name="application-name" content="Supabase Studio" data-sentry-element="meta" data-sentry-source-file="pages-router.tsx"/><meta name="msapplication-TileColor"
content="#1E1E1E" data-sentry-element="meta" data-sentry-source-file="pages-router.tsx"/><meta name="msapplication-TileImage" content="/dashboard/favicon/mstile-144x144.png" data-
sentry-element="meta" data-sentry-source-file="pages-router.tsx"/><meta name="msapplication-square70x70logo" content="/dashboard/favicon/mstile-70x70.png" data-sentry-element="meta"
data-sentry-source-file="pages-router.tsx"/><meta name="msapplication-square150x150logo" content="/dashboard/favicon/mstile-150x150.png" data-sentry-element="meta" data-sentry-source-
file="pages-router.tsx"/><meta name="msapplication-wide310x150logo" content="/dashboard/favicon/mstile-310x150.png" data-sentry-element="meta" data-sentry-source-file="pages-
router.tsx"/><meta name="msapplication-square310x310logo" content="/dashboard/favicon/mstile-310x310.png" data-sentry-element="meta" data-sentry-source-file="pages-router.tsx"/><meta
name="theme-color" content="#1E1E1E" data-sentry-element="meta" data-sentry-source-file="pages-router.tsx"/><link rel="shortcut icon" href="/dashboard/favicon/favicon.ico"/><link
rel="icon" type="image/x-icon" href="/dashboard/favicon/favicon.ico"/><link rel="apple-touch-icon" href="/dashboard/favicon/favicon.ico"/><title>Supabase</title><meta name="descriptio
content="Supabase Studio"/><meta name="next-head-count" content="31"/><link rel="stylesheet" type="text/css" data-name="vs/editor/editor.main"
href="https://cdnjs.cloudflare.com/ajax/libs/monaco-editor/0.37.0/min/vs/editor/editor.main.css"/><link rel="preload" href="https://frontend-
assets.supabase.com/studio/c53b53a634a2/_next/static/media/907906e8fbfce607-s.p.woff2" as="font" type="font/woff2" crossorigin="anonymous" data-next-font="size-adjust"/><link
rel="preload" href="https://frontend-assets.supabase.com/studio/c53b53a634a2/_next/static/media/54303b32ba90fa38-s.p.woff2" as="font" type="font/woff2" crossorigin="anonymous" data-
next-font="size-adjust"/><link rel="preload" href="https://frontend-assets.supabase.com/studio/c53b53a634a2/_next/static/media/8c3413ba6d69dfc1-s.p.woff2" as="font" type="font/woff2"
crossorigin="anonymous" data-next-font="size-adjust"/><link rel="preload" href="https://frontend-assets.supabase.com/studio/c53b53a634a2/_next/static/media/5c27cc9f8a2b89ca-s.p.woff2"
as="font" type="font/woff2" crossorigin="anonymous" data-next-font="size-adjust"/><link rel="preload" href="https://frontend-
assets.supabase.com/studio/c53b53a634a2/_next/static/media/0cc7d245d5cfdd81-s.p.woff2" as="font" type="font/woff2" crossorigin="anonymous" data-next-font="size-adjust"/><link
rel="preload" href="https://frontend-assets.supabase.com/studio/c53b53a634a2/_next/static/media/01027326dc6ad359-s.p.woff2" as="font" type="font/woff2" crossorigin="anonymous" data-
next-font="size-adjust"/><link rel="preload" href="https://frontend-assets.supabase.com/studio/c53b53a634a2/_next/static/media/fe723311b54e86ee-s.p.woff2" as="font" type="font/woff2"
crossorigin="anonymous" data-next-font="size-adjust"/><link rel="preload" href="https://frontend-assets.supabase.com/studio/c53b53a634a2/_next/static/media/dba81c1208da12ee-s.p.woff2"
as="font" type="font/woff2" crossorigin="anonymous" data-next-font="size-adjust"/><link rel="preload" href="https://frontend-
assets.supabase.com/studio/c53b53a634a2/_next/static/css/128bd0e82b72fc5d.css?dpl=dpl_DDVpquE98JNbuSKuR7Gc2fLymSSB" as="style"/><link rel="stylesheet" href="https://frontend-
assets.supabase.com/studio/c53b53a634a2/_next/static/css/128bd0e82b72fc5d.css?dpl=dpl_DDVpquE98JNbuSKuR7Gc2fLymSSB" data-n-g=""/><link rel="preload" href="https://frontend-
assets.supabase.com/studio/c53b53a634a2/_next/static/css/bd3c93ad86eb9cae.css?dpl=dpl_DDVpquE98JNbuSKuR7Gc2fLymSSB" as="style"/><link rel="stylesheet" href="https://frontend-
assets.supabase.com/studio/c53b53a634a2/_next/static/css/bd3c93ad86eb9cae.css?dpl=dpl_DDVpquE98JNbuSKuR7Gc2fLymSSB" data-n-p=""/><noscript data-n-css=""></noscript><script defer=""
nomodule="" src="https://frontend-assets.supabase.com/studio/c53b53a634a2/_next/static/chunks/polyfills-42372ed130431b0a.js?dpl=dpl_DDVpquE98JNbuSKuR7Gc2fLymSSB"></script><script
src="https://frontend-assets.supabase.com/studio/c53b53a634a2/_next/static/chunks/webpack-c5f48090f8b5ba08.js?dpl=dpl_DDVpquE98JNbuSKuR7Gc2fLymSSB" defer=""></script><script
src="https://frontend-assets.supabase.com/studio/c53b53a634a2/_next/static/chunks/framework-f441d1cc7b5b6851.js?dpl=dpl_DDVpquE98JNbuSKuR7Gc2fLymSSB" defer=""></script><script
src="https://frontend-assets.supabase.com/studio/c53b53a634a2/_next/static/chunks/main-5dd0833ff4307d99.js?dpl=dpl_DDVpquE98JNbuSKuR7Gc2fLymSSB" defer=""></script><script
```

The form does not contain any hidden CSRF token field like the one above, the site might not be using CSRF protection.

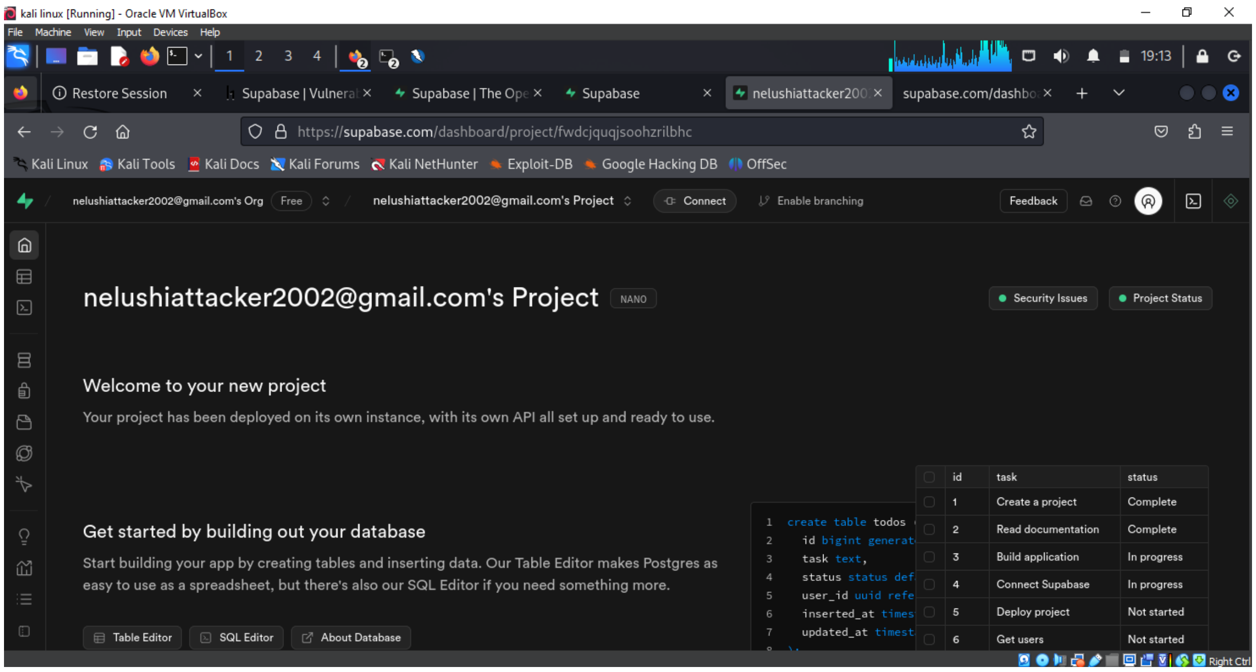
3. I logged into the site using my normal credentials and now I have an active session.

4. Next, I Created a simple HTML page that will submit data to the target website from another origin. This tests whether the website is vulnerable to CSRF attacks.



kali linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
nelushi@kali: ~
File Actions Edit View Help

```
 GNU nano 8.1                                                        csrf.html *
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Test CSRF Absence</title>
</head>
   <body>
       <h1>Test CSRF Absence</h1>
       <form action="https://supabase.com/dashboard/project/fwdcjquqjsoohzrilbhc">
         <input type="text" name="email" value="attacker@example.com">
         <input type="text" name="username" value="hacked">
         <button type="submit" value="Click Me">Submit</button>
      </form>
   </body>
</html>
```



nelushiattacker2002@gmail.com's Project NANO

Welcome to your new project

Your project has been deployed on its own instance, with its own API all set up and ready to use.

Get started by building out your database

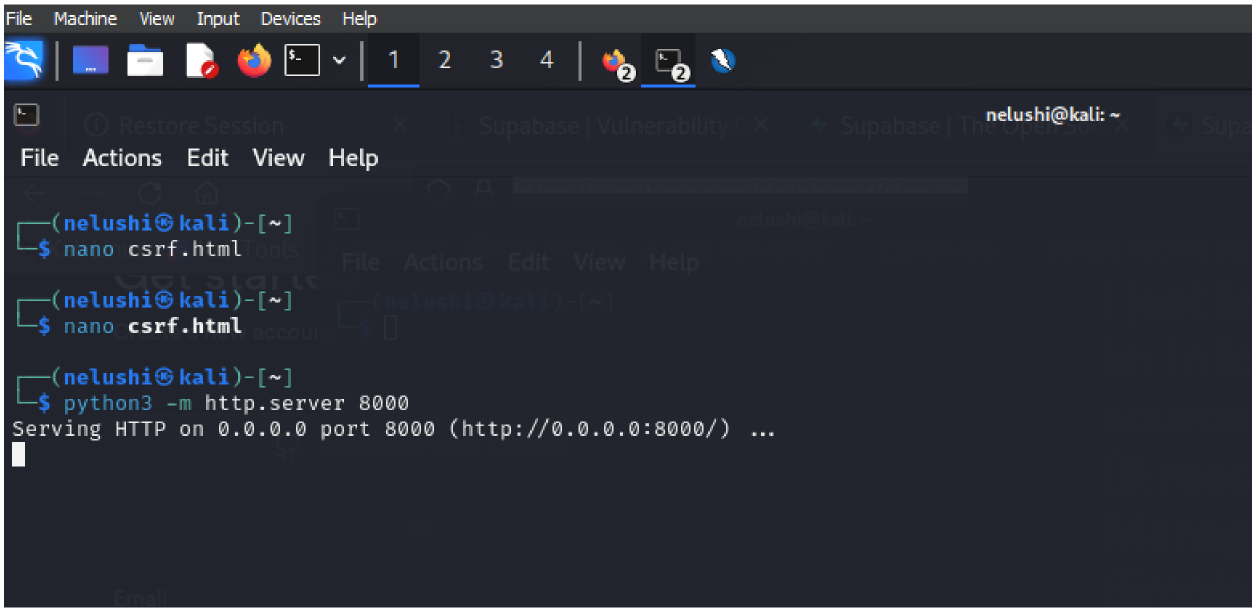
Start building your app by creating tables and inserting data. Our Table Editor makes Postgres as easy to use as a spreadsheet, but there's also our SQL Editor if you need something more.

```
create table todos
  id bigint generat
  task text,
  status status def
  user_id uuid refe
  inserted_at times
  updated_at timest
```

id	task	status
1	Create a project	Complete
2	Read documentation	Complete
3	Build application	In progress
4	Connect Supabase	In progress
5	Deploy project	Not started
6	Get users	Not started

5. After that, I started a local web server to run the html code.

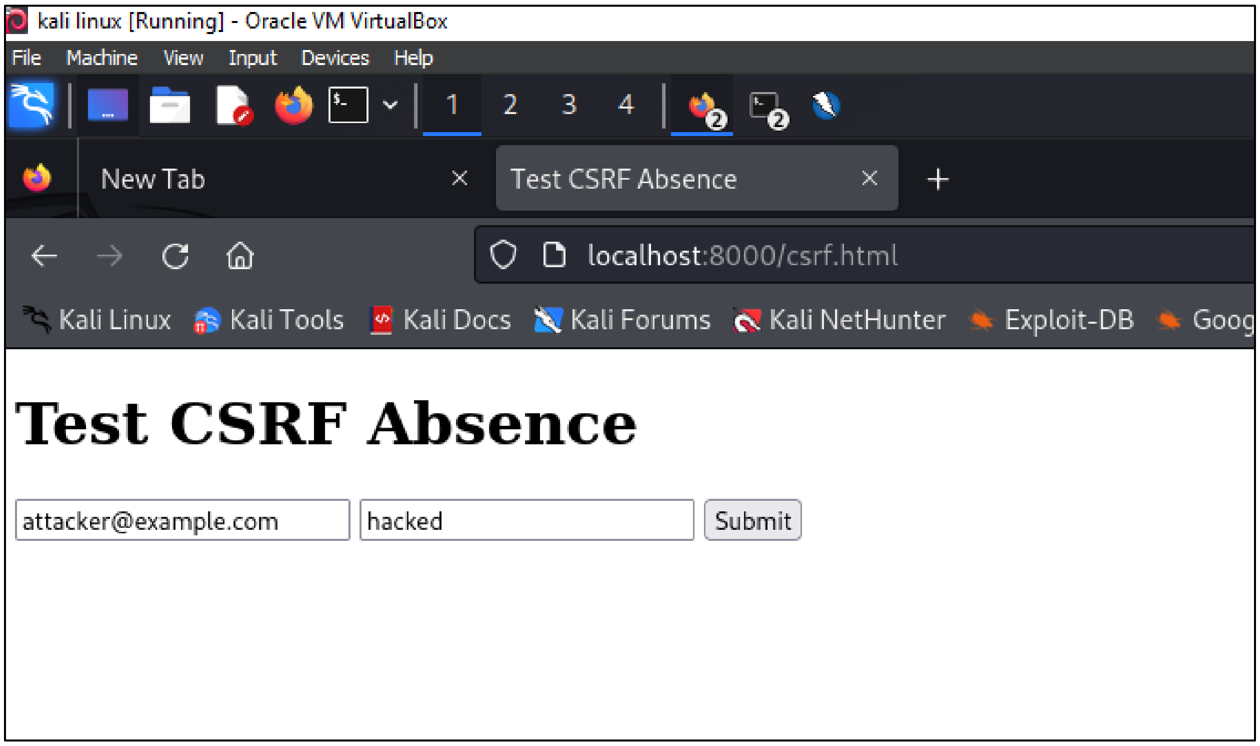
python3 -m http.server 8000


File Machine View Input Devices Help
nelushi@kali: ~
File Actions Edit View Help

```
┌──(nelushi㉿kali)-[~]
└─$ nano csrf.html

┌──(nelushi㉿kali)-[~]
└─$ nano csrf.html

┌──(nelushi㉿kali)-[~]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

6. Then I opened firefox and searched this to run the html code.

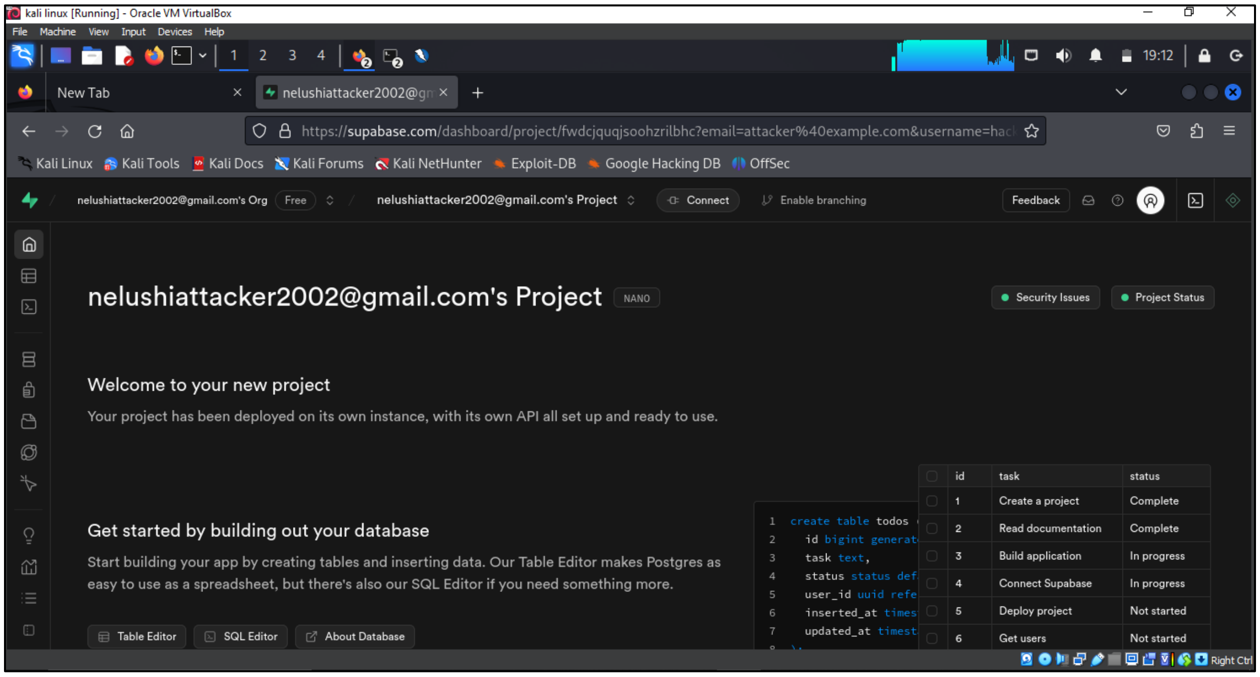
   [***http://localhost:8000/csrf.html***](http://localhost:8000/csrf.html)



Test CSRF Absence

attacker@example.com | hacked | Submit

After submitting this attacker credentials, it directed me to the logged page as I was the victim.


kali linux [Running] - Oracle VM VirtualBox

https://supabase.com/dashboard/project/fwdcjququjsoohzrilbhc?email=attacker%40example.com&username=hac

nelushiattacker2002@gmail.com's Org · Free / nelushiattacker2002@gmail.com's Project · Connect · Enable branching · Feedback

nelushiattacker2002@gmail.com's Project NANO

Security Issues · Project Status

Welcome to your new project

Your project has been deployed on its own instance, with its own API all set up and ready to use.

Get started by building out your database

Start building your app by creating tables and inserting data. Our Table Editor makes Postgres as easy to use as a spreadsheet, but there's also our SQL Editor if you need something more.

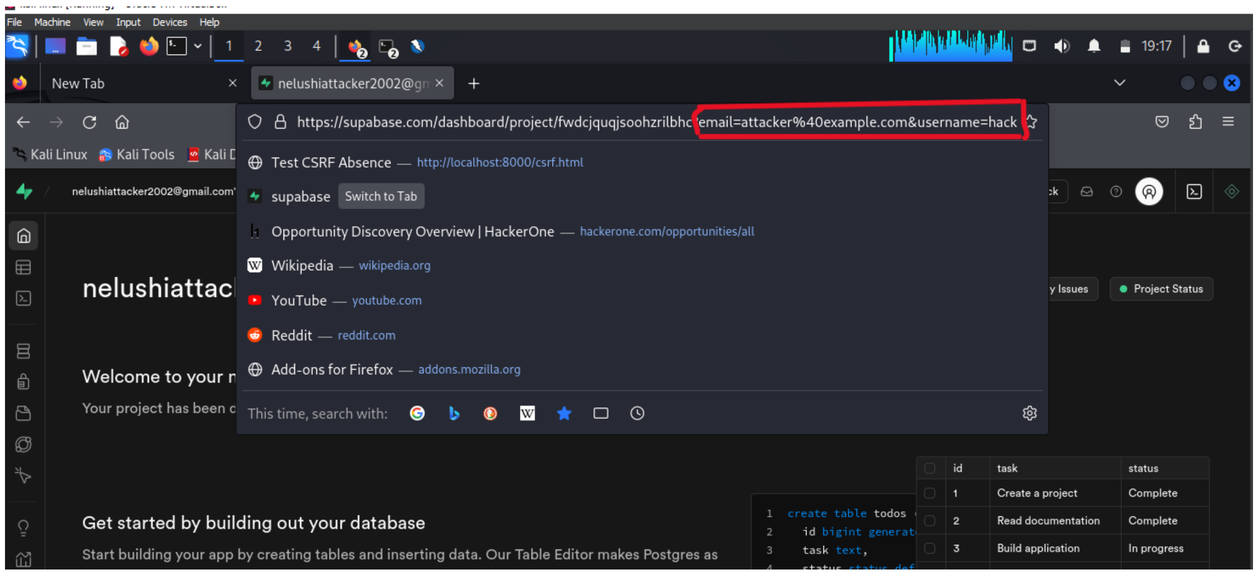
Table Editor · SQL Editor · About Database

```
create table todos
  id bigint generat
  task text,
  status status def
  user_id uuid refe
  inserted_at times
  updated_at timest
```

id	task	status
1	Create a project	Complete
2	Read documentation	Complete
3	Build application	In progress
4	Connect Supabase	In progress
5	Deploy project	Not started
6	Get users	Not started

The website's server procceded the request as it came from the victim.

The URL contains the attacker credentials, yet it directed me to the account I was logged it.


https://supabase.com/dashboard/project/fwdcjququjsoohzrilbhc?email=attacker%40example.com&username=hack

Test CSRF Absence — http://localhost:8000/csrf.html

supabase · Switch to Tab

Opportunity Discovery Overview | HackerOne — hackerone.com/opportunities/all

Wikipedia — wikipedia.org

YouTube — youtube.com

Reddit — reddit.com

Add-ons for Firefox — addons.mozilla.org

This time, search with:

An Example of Attack in Action

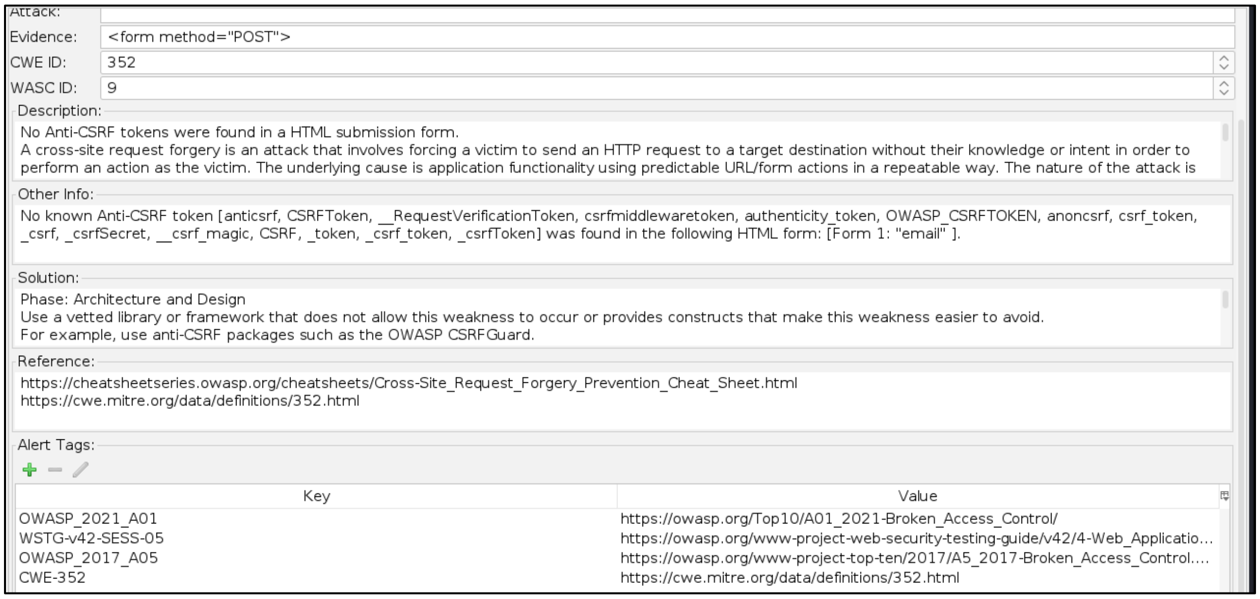
The victim is logged in to supabase.com login page.

The attacker sends a link to the victim directing them to the HTML page I created.

The victim will click the link, and the form submission occurs and the target website processes the form submission as if the victim submitted it, and at this point, the victim's account has been compromised without them knowing.

Lack of Anti-CSRF tokens is under **OWASP 2021 A01: Broken Access Control** and **WSTG-v42-SESS-05**, because it allows attackers to trick authenticated users to conduct unauthorized activities. Users may be manipulated into sending unintended requests by lack of CSRF protection, leading to account compromise or privilege escalation**.**

Keys and values of the alert



Attack:

Evidence: <form method="POST">

CWE ID: 352

WASC ID: 9

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

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, _csrfToken] was found in the following HTML form: [Form 1: "email"].

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

Alert Tags:

Key	Value
OWASP_2021_A01	https://owasp.org/Top10/A01_2021-Broken_Access_Control/
WSTG-v42-SESS-05	https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Applicatio…
OWASP_2017_A05	https://owasp.org/www-project-top-ten/2017/A5_2017-Broken_Access_Control.…
CWE-352	https://cwe.mitre.org/data/definitions/352.html

Proposed mitigation or fix

Leverage CSRF Tokens - Implement a unique CSRF token in every form and validate it server-side for all requests that modify the state of the server.

Leverage Same-Site Cookies - When cookies are set, set the SameSite attribute (Strict or Lax) to block requests that are cross-origin.

Use Header Checks - Verify the HTTP referer and origin header to make sure they fall to the domain you are serving.

Use CSRF Tokens in AJAX - Pass the CSRF token as an ajax header for AJAX requests.

Rotate Tokens - Changing tokens periodically helps limit reuse.

Regularly Test - Regularly test for CSRF vulnerabilities with something like OWASP ZAP.