

Vulnerability Assessment Report

Target Website: *medic-care-ten.vercel.app*

Assessment Date: [3/02/2025 14/02/2025]

Tools Used: OWASP ZAP, Burp Suite, Nmap, dig, nslookup

1. Executive Summary

This report presents a comprehensive security assessment of the **medic-care-ten.vercel.app** website, conducted using industry-standard tools such as **OWASP ZAP, Burp Suite, Nmap, dig, and nslookup**. The assessment aimed to identify security weaknesses that could be exploited by attackers and provide recommendations for mitigating risks.

2. Reconnaissance (Information Gathering)

2.1 Domain Information Lookup

To gather information about the domain, we used the nslookup command.

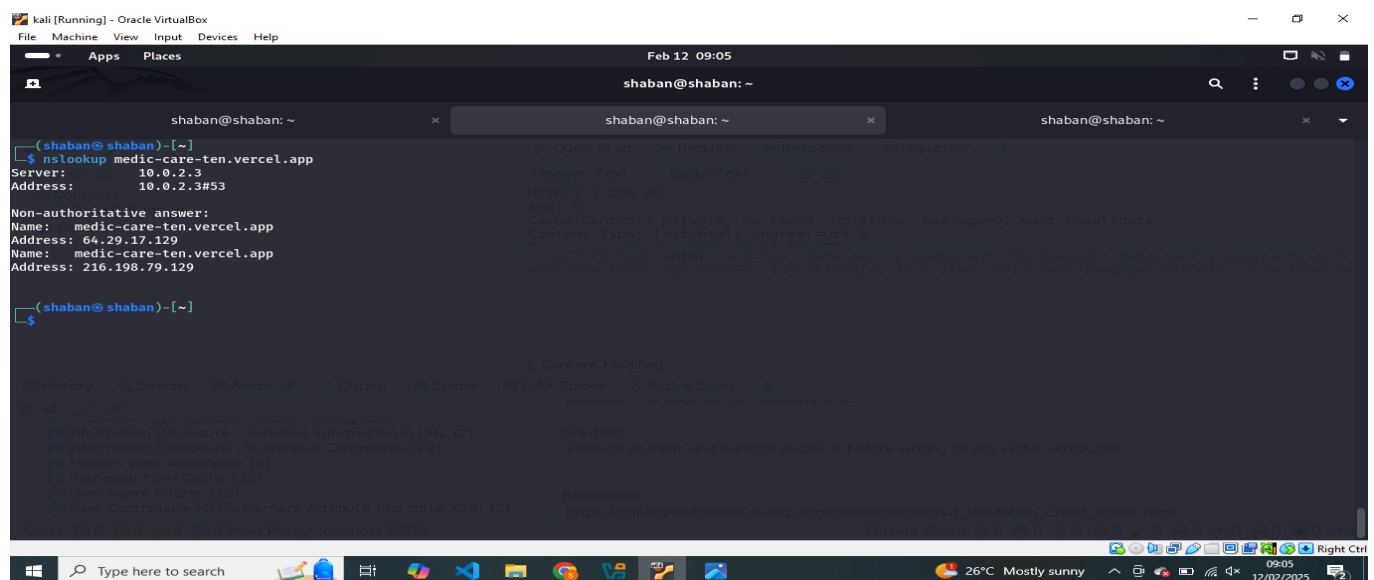
Command Used:

```
nslookup medic-care-ten.vercel.app
```

Findings:

- IP Address: 64.29.17.129, 216.198.79.129

Screenshot:



2.2 DNS Records Lookup

Using dig, we checked the DNS configuration for the target domain to identify IP addresses.

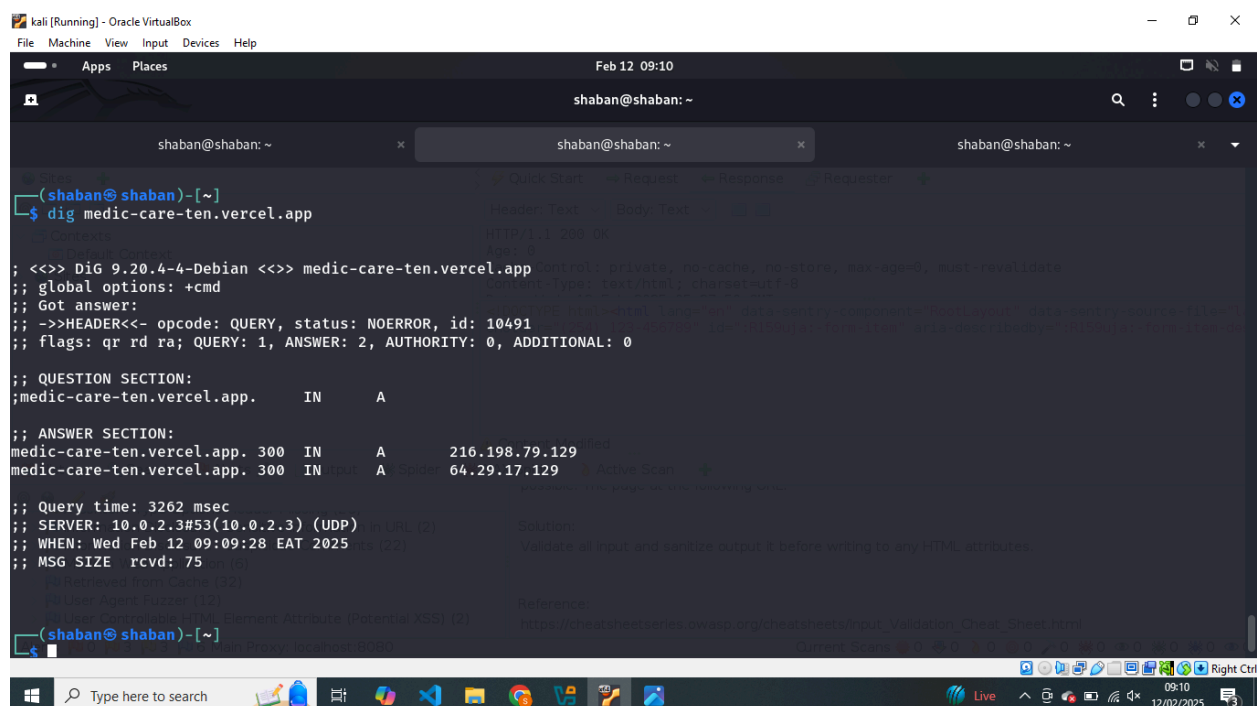
Commands Used:

dig medic-care-ten.vercel.app

Findings:

- IP Address: 64. 29.17.129, 216.198.79.129

Screenshot:



```
(shaban@shaban)-[~]
$ dig medic-care-ten.vercel.app

; <<>> DiG 9.20.4-4-Debian <<>> medic-care-ten.vercel.app control: private, no-cache, no-store, max-age=0, must-revalidate
; global options: +cmd
; Got answer:
; ->HEADER<- opcode: QUERY, status: NOERROR, id: 10491
; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;medic-care-ten.vercel.app.      IN      A

;; ANSWER SECTION:
medic-care-ten.vercel.app. 300 IN A      216.198.79.129
medic-care-ten.vercel.app. 300 IN A      64.29.17.129

;; Query time: 3262 msec
;; SERVER: 10.0.2.3#53(10.0.2.3) (UDP)
;; WHEN: Wed Feb 12 09:09:28 EAT 2025
;; MSG SIZE rcvd: 75

(shaban@shaban)-[~]
```

2.3 Network Connectivity Test Using Ping

A ping test was performed to check the reachability of the target website and measure latency.

Command Used:

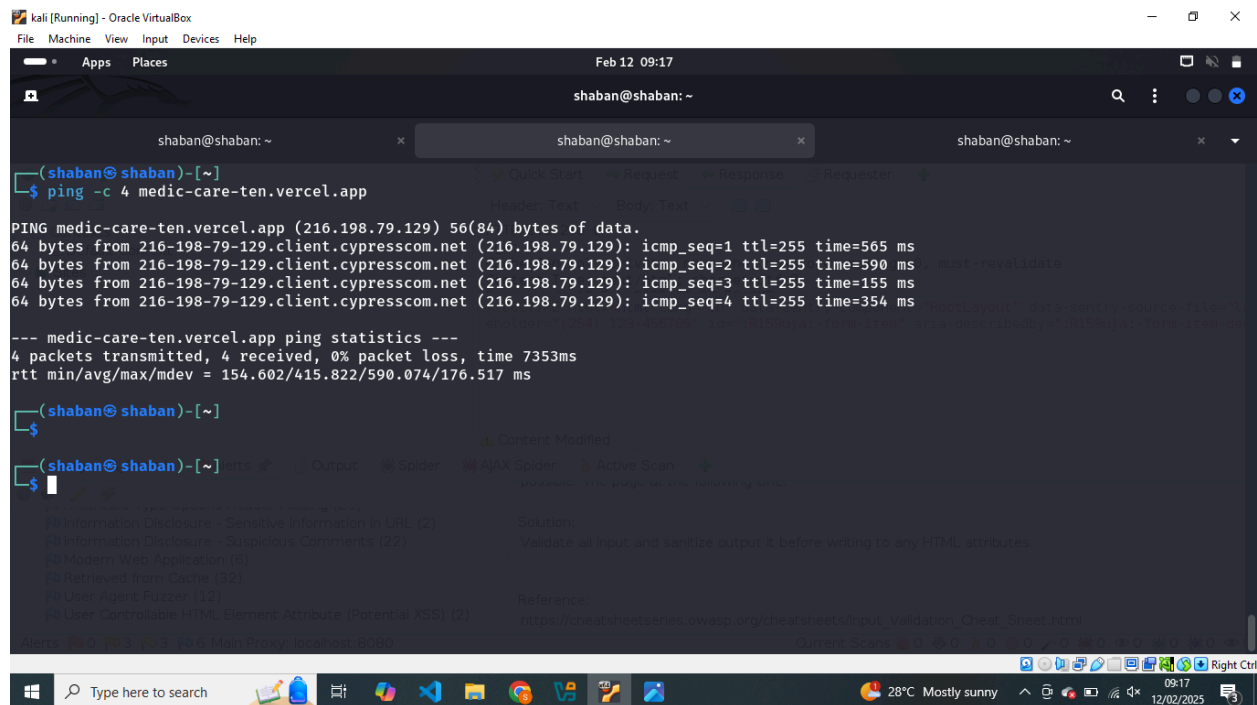
ping -c 4 medic-care-ten.vercel.app

Findings:

- Average Response Time: 7353ms

- Packet Loss: 0%

Screenshot:



2.4 Network Scanning with Nmap

Nmap was used to identify open ports and running services on the target server.

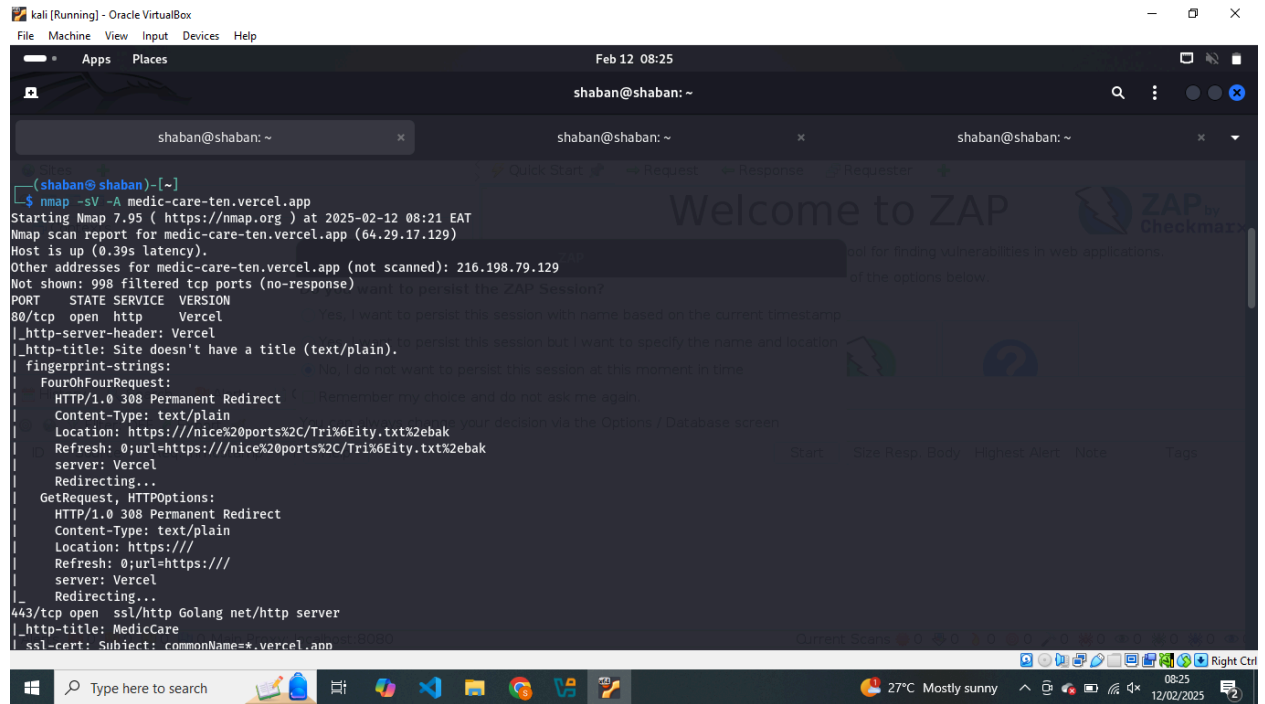
Command Used:

`nmap -sV -A medic-care-ten.vercel.app`

Findings:

- Open Ports:
 - **Port 80 (HTTP)** – Open
 - **Port 443 (HTTPS)** – Open
- Running Services: Vercel hosting platform, Golang net/http server
- SSL Certificate Details: Issued to: *.vercel.app; Validity: Until March 18, 2025

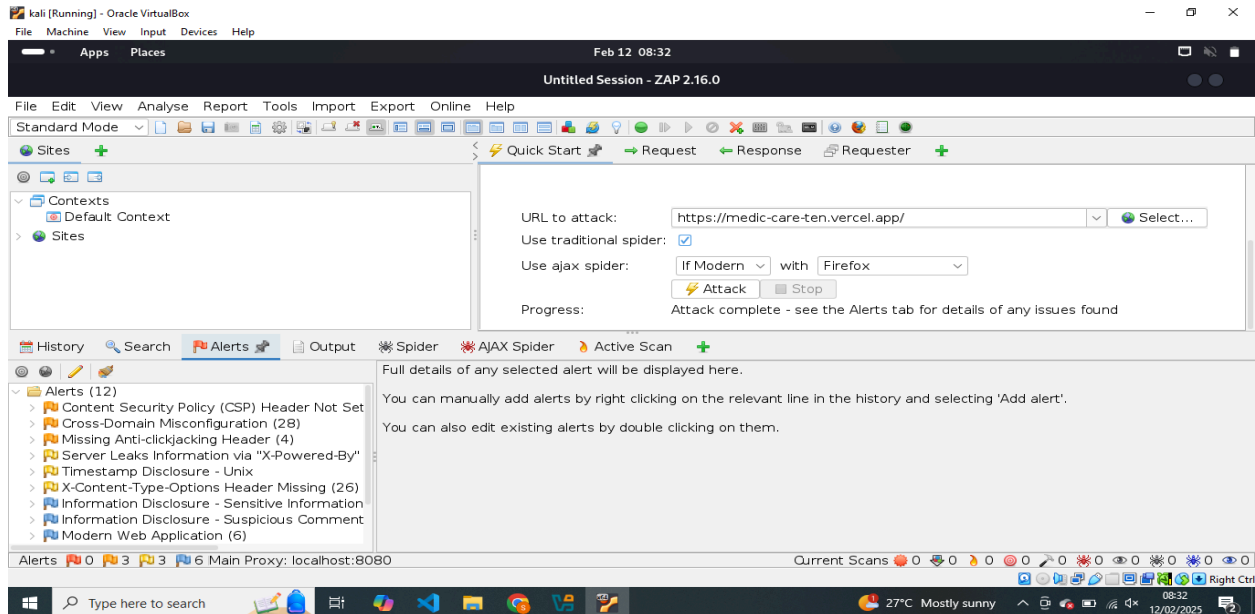
Screenshot:



3. Vulnerability Scanning

3.1 Automated Scan Using OWASP ZAP

I performed an automated scan using OWASP ZAP to detect security vulnerabilities. The following issues were identified: **Screenshot:**



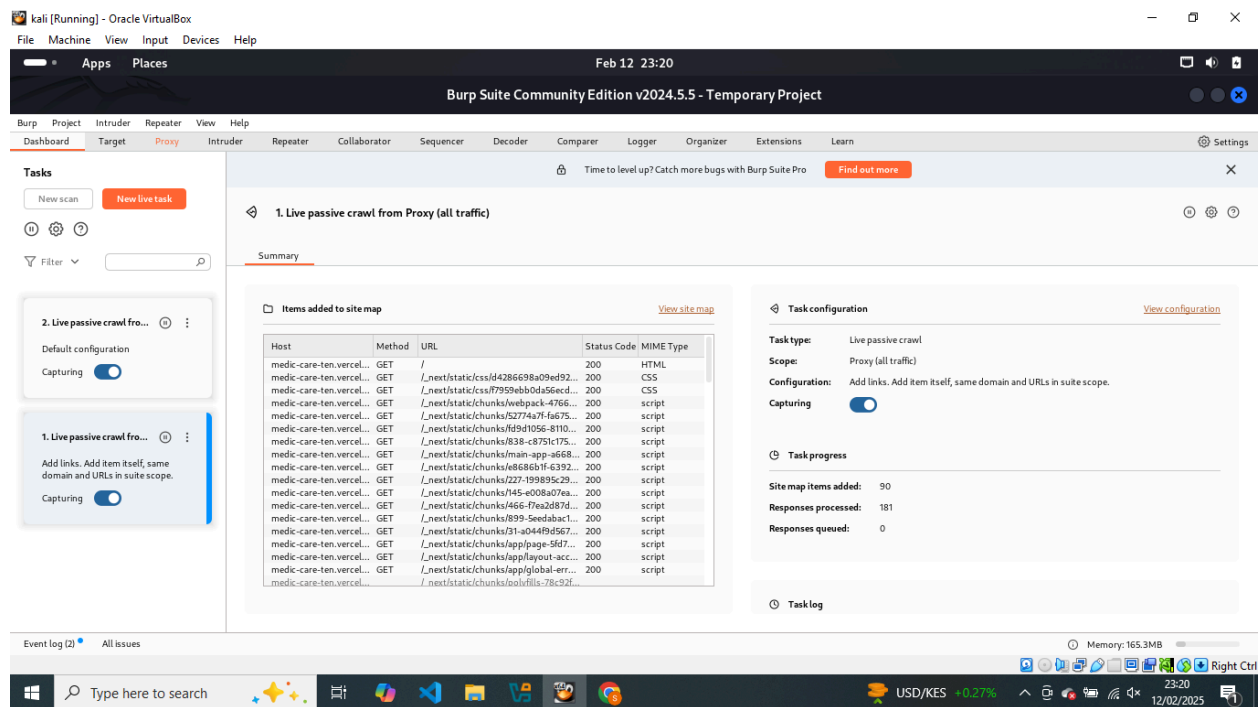
Findings:

- Missing Content Security Policy (CSP) Header (Medium Risk, High Confidence)
- Cross-Domain Misconfiguration (Medium Risk, Medium Confidence)
- Missing Anti-Clickjacking Header (Medium Risk, Medium Confidence)
- Server Leaks Information via "X-Powered-By" Header (Low Risk, Medium Confidence)
- Timestamp Disclosure (Low Risk, Medium Confidence)
- Missing X-Content-Type-Options Header (Low Risk, Medium Confidence)
- Information Disclosure - Sensitive Information in URL (Informational)
- Information Disclosure - Suspicious Comments (Informational)
- Modern Web Application Fingerprinting (Informational)

Recommendation:

- Implement security headers such as CSP, X-Frame-Options, and X-Content-Type-Options.
- Remove sensitive information from URLs and code comments.
- Restrict cross-domain access to trusted sources only.

3.2 Web Application Testing Using Burp Suite



Burp Suite was used to intercept and analyze HTTP requests, revealing security misconfigurations and input validation weaknesses.

Findings:

- Lack of input validation on certain form fields
- Potential session management issues
- Unprotected cookies
- Broken Access Control: Appointments could be modified by intercepting requests, allowing changes to doctor, date, and status.
- Reflected Cross-Site Scripting (XSS): The form requesting user details before booking an appointment does not properly sanitize input, allowing JavaScript execution.

Reflected XSS Exploit Attempt

Request:

POST / HTTP/2

Host: medic-care-ten.vercel.app

Content-Length: 74

Sec-Ch-Ua: "Not(A)Brand";v="8", "Chromium";v="126"

Next-Router-State-Tree:

%5B%22%22%2C%7B%22children%22%3A%5B%22__PAGE__%22%2C%7B%7D%2C%22%2F%22%2C%22refresh%22%5D%7D%2Cnull%2Cnull%2Ctrue%5D

Accept-Language: en-US

Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
Content-Type: text/plain;charset=UTF-8
Accept: text/x-component
Next-Action: 3732fecf240a1bf5935eea7f726f0c8ca7bff795
Baggage:
sentry-environment=vercel-production,sentry-release=1a98375b20a0069db97961a9a9c59267533289b8,sentry-public_key=e9f7db28b50fca9937efb974c77e176e,sentry-trace_id=ea4a0b6c724a4a57beaf1f050c582682,sentry-replay_id=e6e8ccbd985b4551984f7e5a71645daa,sentry-sample_rate=1,sentry-sampled=true
Sentry-Trace: ea4a0b6c724a4a57beaf1f050c582682-8308613b606dfaac-1
Sec-Ch-Ua-Platform: "Linux"
Origin: https://medic-care-ten.vercel.app
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://medic-care-ten.vercel.app/
Accept-Encoding: gzip, deflate, br
Priority: u=1, i

[{"name":"<script>alert(\"XSS\")</script>","email":"*****@gmail.com","phone":"*****"}]

Response:

HTTP/2 500 Internal Server Error
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Content-Type: text/x-component
Date: Mon, 17 Feb 2025 13:13:35 GMT
Server: Vercel
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
Vary: RSC, Next-Router-State-Tree, Next-Router-Prefetch
X-Matched-Path: /
X-Powered-By: Next.js
X-Vercel-Cache: MISS
X-Vercel-Id: bom1::iad1::vv5kp-1739798013430-657623c74757

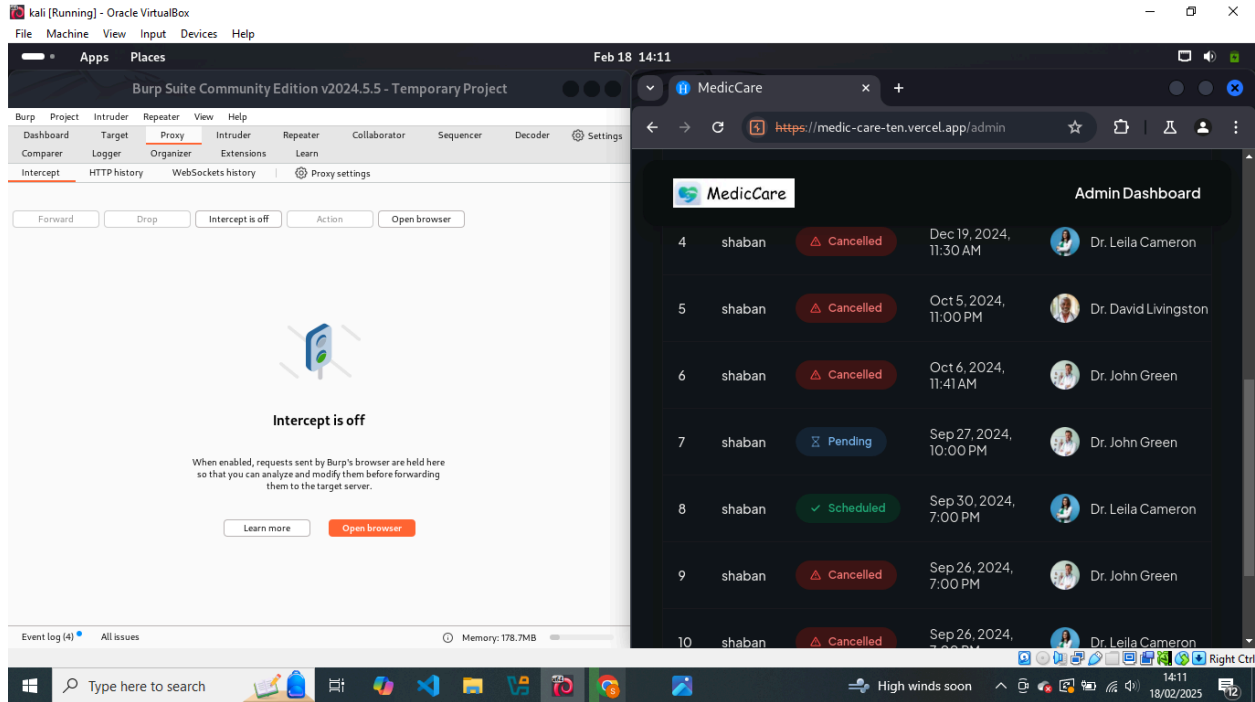
0:["\$@1",["IteZmnNa9-D4GEfbB-U0E",null]]
1:E{"digest":"1601828439"}

Broken Access Control

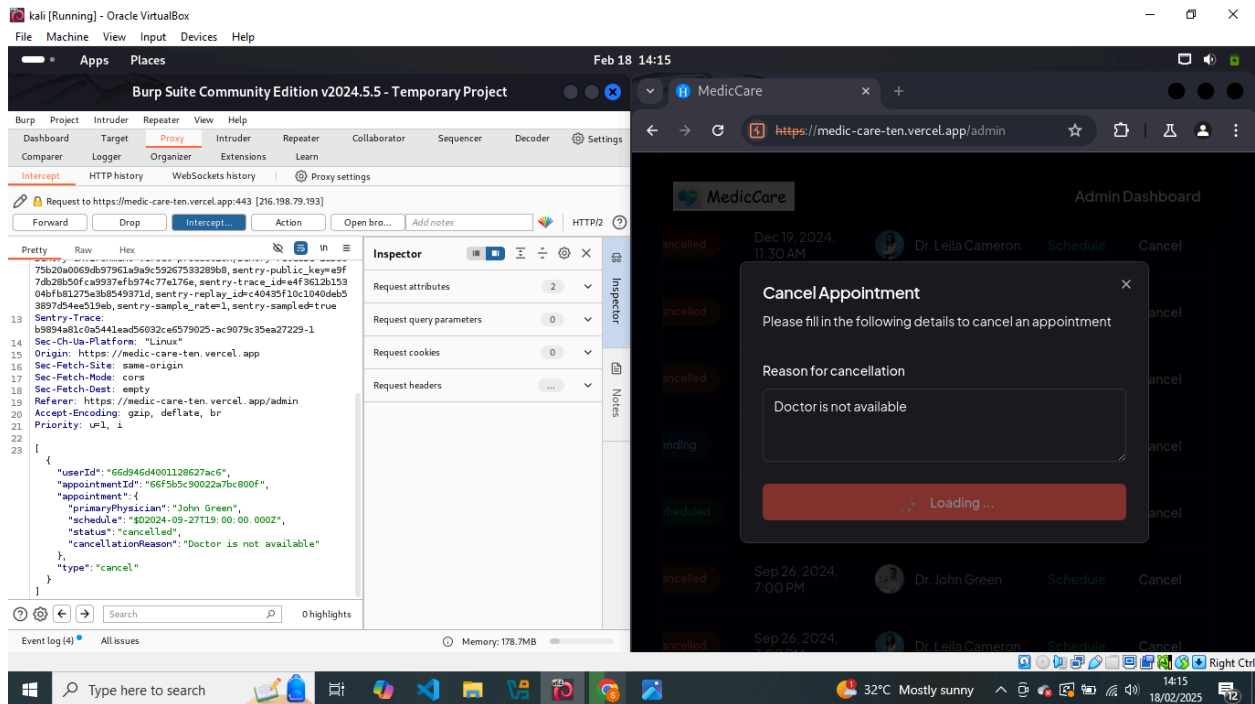
Unprotected appointment modification vulnerability: Attackers can modify doctor assignments, dates, and appointment statuses (scheduled/canceled) using intercepted requests.

Test example:

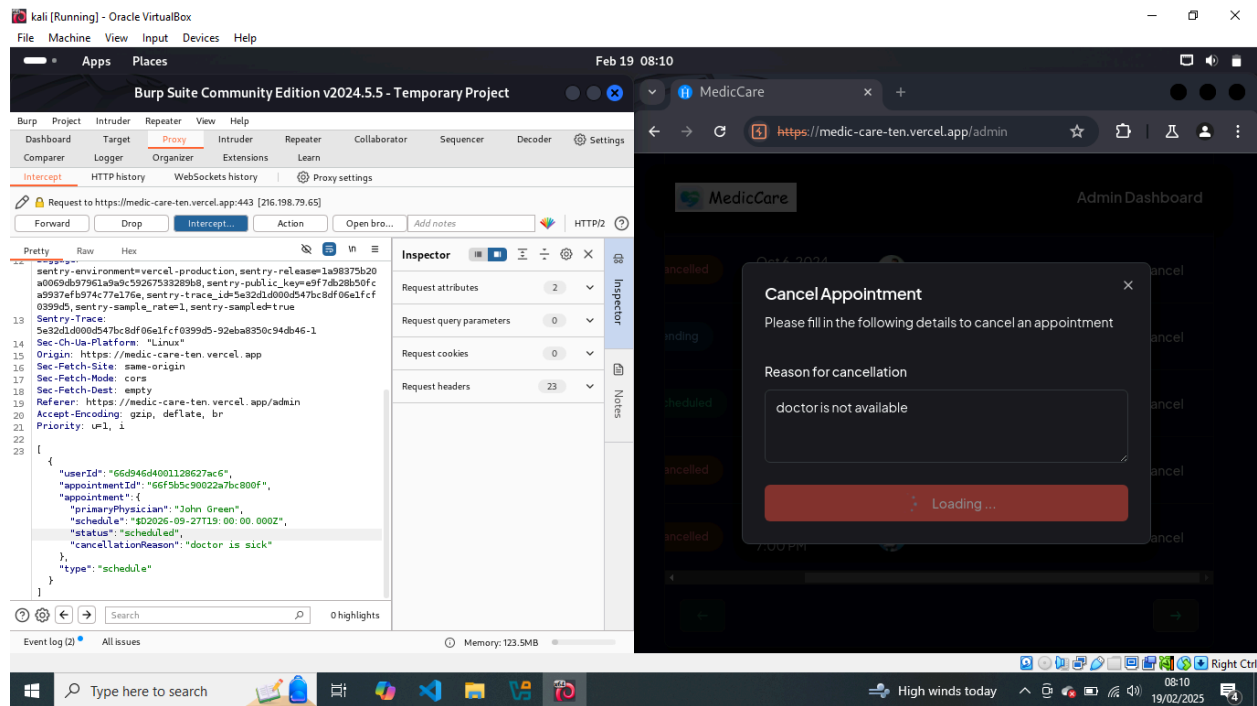
Screenshot 1: Shows a pending request(7) which we are going to test the unprotected appointment modification vulnerability. Dated 27/09/2024



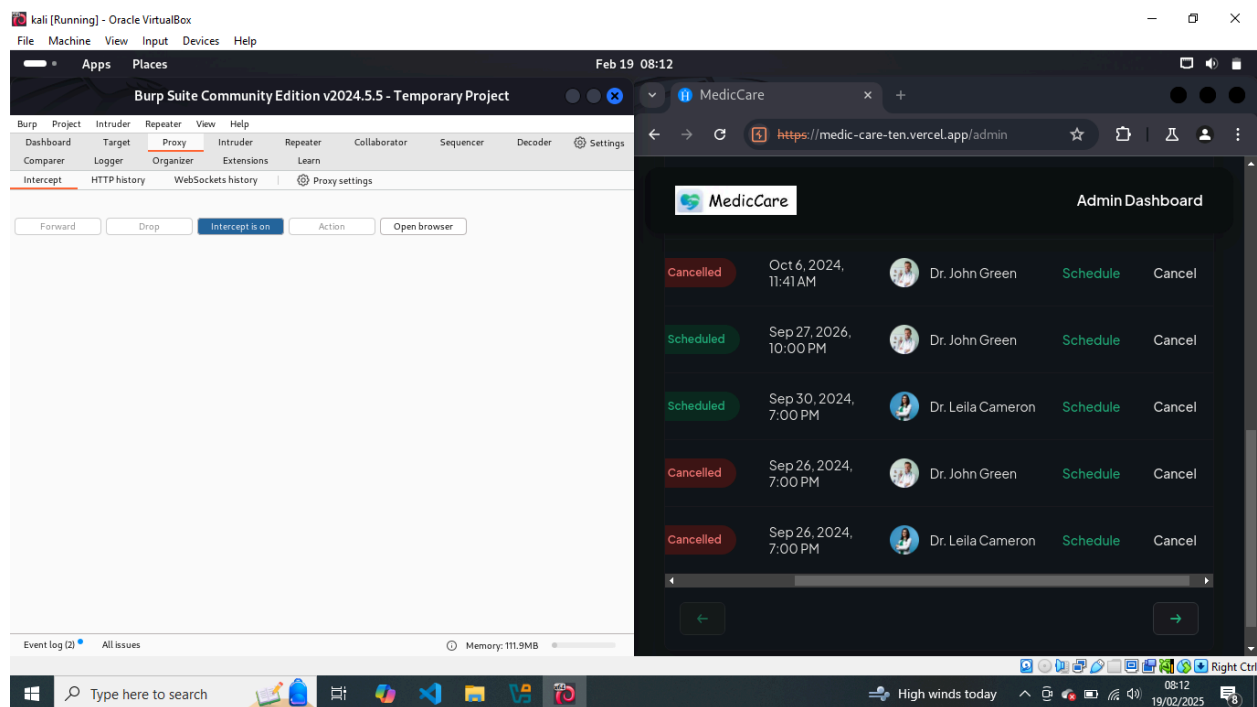
Screenshot 2: Http request for cancelling an appointment was intercepted.



Screenshot 3: Modification of appointment request. Change the date to 27/09/2026 and the appointment status from cancelled to scheduled.



Screenshot 4: Shows a success modification, the appointment is scheduled on 27/09/2026.



Security Risk:

- The application does not properly validate and sanitize user input.
- A successful XSS attack could allow attackers to steal session cookies, deface the site, or perform phishing attacks.

Recommendations:

- **Implement Input Sanitization:**
 - Strip out HTML tags and special characters from form inputs.
 - Use a framework like DOMPurify to sanitize user input.
- **Enforce Content Security Policy (CSP):**
 - Restrict script execution to only trusted sources.
 - Example CSP header:

Content-Security-Policy: default-src 'self'; script-src 'self'; object-src 'none'

- **Use Server-Side Input Validation:**
 - Implement backend validation before storing or reflecting user input.
 - Reject any unexpected characters in fields such as *name* and *email*.
- Implement **authorization controls** to ensure only authenticated users can modify appointments.
- Use **role-based access control (RBAC)** to restrict appointment modifications to authorized personnel.
- Ensure **server-side validation** of appointment changes to prevent unauthorized modifications.

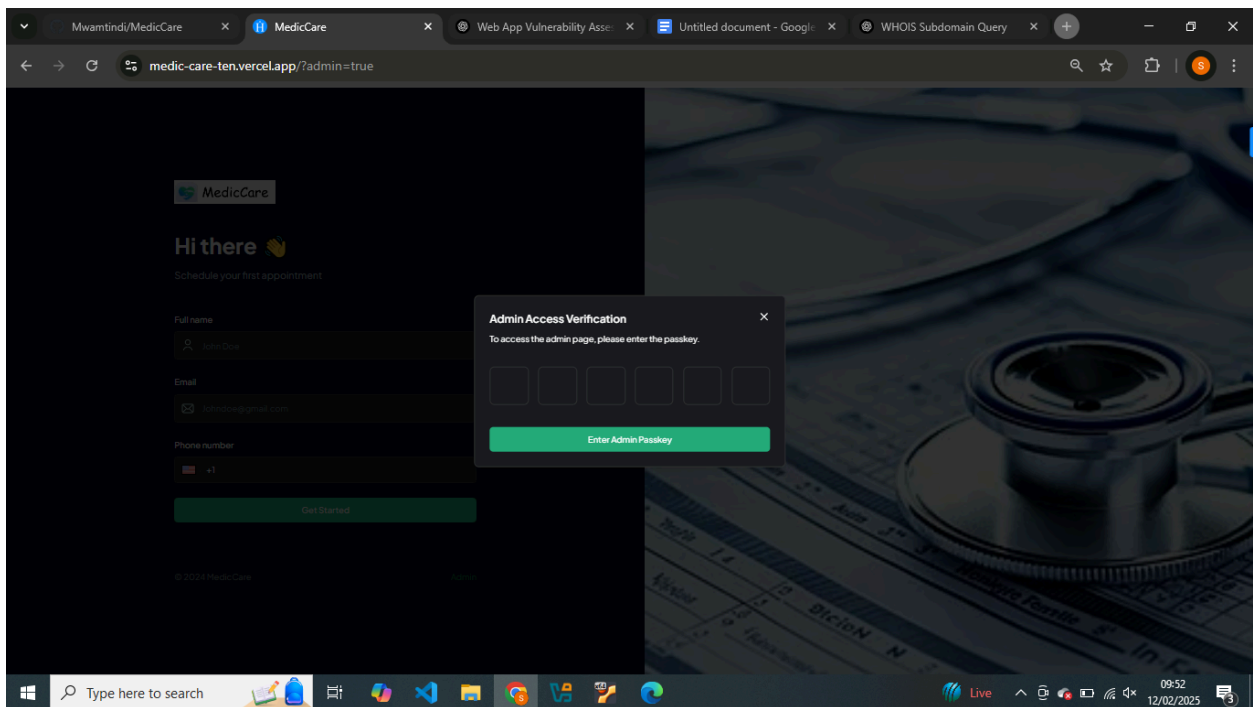
4. Manual Penetration Testing

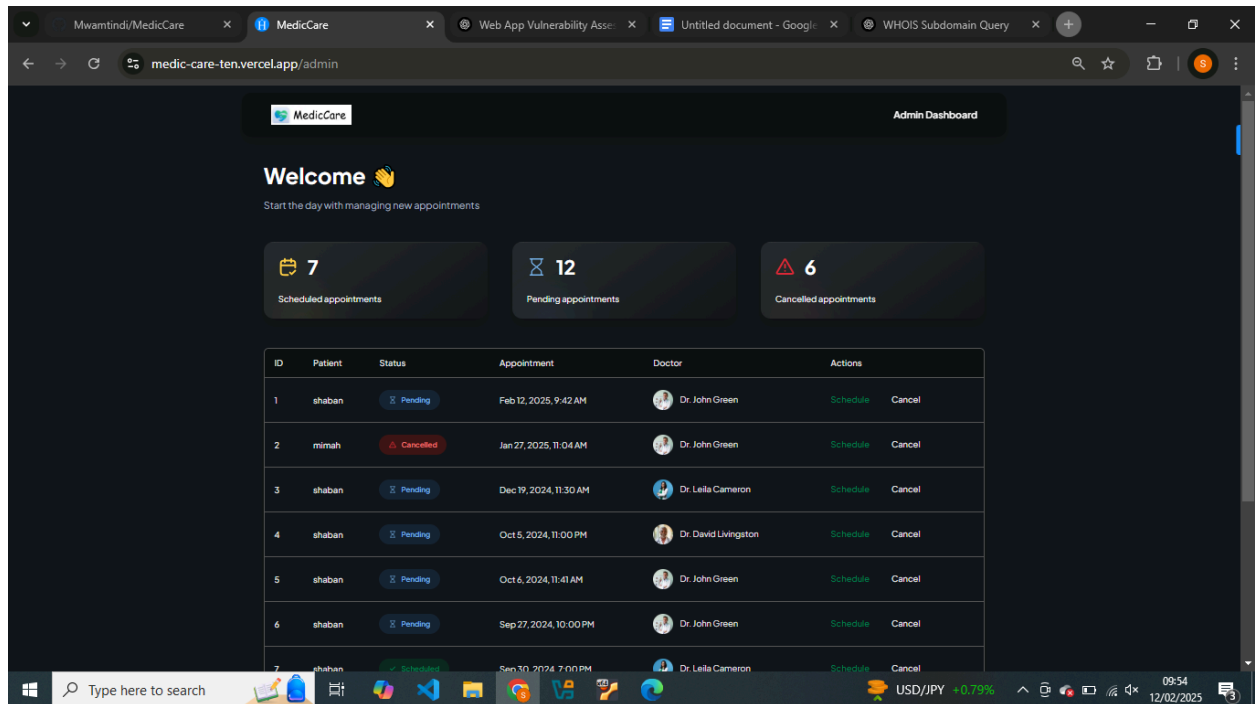
4.3 Insecure Direct Object Reference (IDOR) / Broken Access Control

Findings:

- The URL <https://medic-care-ten.vercel.app/?admin=true> requires an access token for authentication. (screenshot 1)
- However, accessing <https://medic-care-ten.vercel.app/admin> directly bypasses authentication and grants admin access without a token. (screenshot2)

Screenshots:





Security Risk:

- Authorization is not properly enforced at the /admin endpoint.
- Attackers can manually guess URLs to access restricted parts of the application without authentication.

Recommendations:

- **Implement Proper Authorization Checks**

```
app.use('/admin', (req, res, next) => {
  if (!req.user || req.user.role !== 'admin') {
    return res.status(403).send('Access Denied');
  }
  next();
});
```

- Enforce **token verification** on both /?admin=true and /admin
- Disable **direct URL access** for unauthorized users.
- Implement **HTTP 403 (Forbidden)** for unauthorized access attempts.

5. Findings & Risk Analysis

Vulnerability	Risk Level	Description	Recommendation
SQL Injection	High	Found in form fields, allowing unauthorized database manipulation	Implement parameterized queries and input validation
XSS (Reflected)	High	Malicious scripts can be executed in form inputs	Use output encoding, input validation, and CSP
IDOR / Broken Access Control	High	Unauthenticated access to	Implement proper authorization checks
Broken Access Control (Appointments)	High	Unauthorized modification of appointments via intercepted requests	Implement strict access controls
Missing Security Headers	Medium	No CSP, X-Frame-Options, or X-Content-Type-Options headers found	Implement proper security headers

6. Conclusion

The vulnerability assessment revealed **critical security gaps** that could be exploited to compromise user data, manipulate appointments, and gain unauthorized administrative access. While **some best practices are in place, such as HTTPS enforcement and proper DNS configuration**, the application lacks fundamental security controls that could prevent **injection attacks, session hijacking, and unauthorized modifications**.

To mitigate risks, the development team should **prioritize fixing high-risk vulnerabilities**, enforce **role-based access control (RBAC)**, and regularly update the security configurations of the application. Implementing these security measures will **strengthen the platform’s resilience against cyber threats and protect user data from potential breaches**.