Sri Lanka Institute of Information Technology

Bug Bounty Report - 05

Module: IE2062

Web Security

Year 2, Semester 2

Aazaf Ritha. J – IT23151710

B.Sc. (Hons) in Information Technology

Specialized in Cyber Security

Table of Contents

Introduction 3

Vulnerability Title 5

Description 6

Affected Component 7

Impact Assessment 8

Steps to Reproduce 9

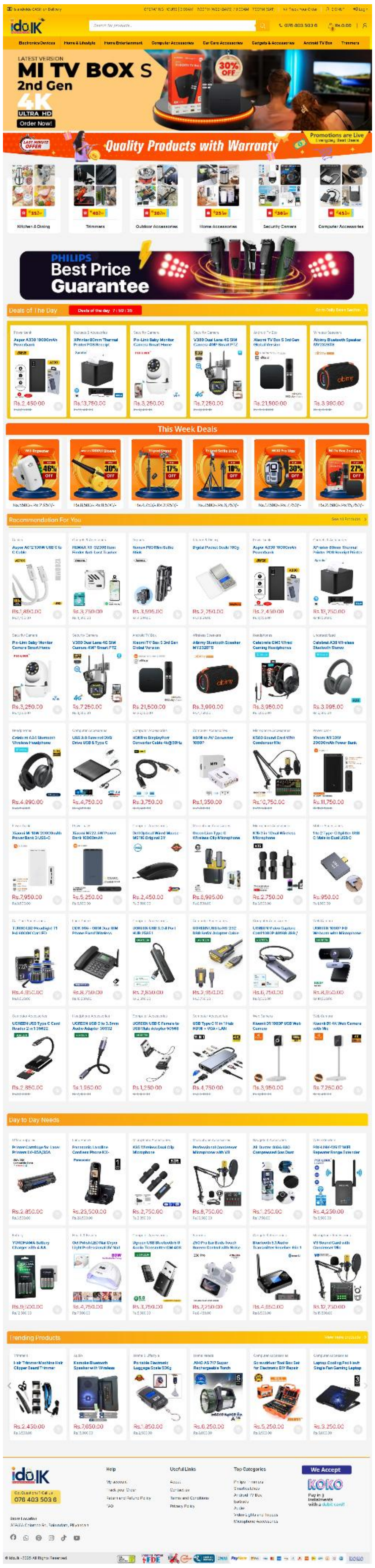
Proof of Concept (Screenshots) 11

Proposed Mitigation or Fix 14

Conclusion 16

Introduction

This report outlines a vulnerability discovered on a Sri Lankan website, https://www.ido.lk. During security testing, a Cross-Domain Misconfiguration vulnerability was identified across multiple components of the site, primarily due to insecure CORS (Cross-Origin Resource Sharing) settings. The issue was confirmed through both manual testing and automated tools, and this report provides technical details, clear reproduction steps, a proof of concept, and practical mitigation recommendations.


Islandwide CASH on Delivery
Track Your Order
Search for products...
076 403 503 6
Rs.0.00
Electronics Devices
Home & Lifestyle
Home Entertainment
Computer Accessories
Car Care Accessories
Gadgets & Accessories
Android TV Box
Trimmers
LATEST VERSION
MI TV BOX S
2nd Gen
4K
ULTRA HD
Order Now!
30% OFF
LAST MINUTE OFFER
Quality Products with Warranty
Promotions are Live
Everyday Best Deals
Kitchen & Dining
Trimmers
Outdoor Accessories
Home Accessories
Security Camera
Computer Accessories
PHILIPS
Best Price
Guarantee
Deals of The Day
Deals of the day 7 : 50 : 35
This Week Deals
Recommendation For You
Day to Day Needs
Trending Products
Hair Trimmer Machine Hair Clipper Beard Trimmer
Rs.2,450.00
Karaoke Bluetooth Speaker with Wireless
Rs.7,650.00
Portable Electronic Luggage Scale 50Kg
Rs.1,850.00
AKO AS 717 Super Rechargeable Torch
Rs.6,250.00
Screwdriver Tool Box Set for Electronic DIY Repair
Rs.5,250.00
Laptop Cooling Pad Hawk Single Fan Gaming Laptop
Rs.3,250.00
Help
My account
Track your Order
Return and Refund Policy
FAQ
Useful Links
About
Contact us
Terms and Conditions
Privacy Policy
Top Categories
Philips Trimmers
Smartwatches
Android TV Box
Earbuds
Audio
Video Lights and Tripods
Microphone Accessories
We Accept
KOKO
Pay in 3 instalments with a debit card!
Got Questions? Call us
076 403 503 6
Store Location
© ido.lk - 2025 All Rights Reserved

Vulnerability Title

Title – Cross Domain Misconfiguration Vulnerability

Risk Level- Medium

Domain – www.ido.lk

Description

A **Cross-Domain Misconfiguration Vulnerability** was discovered on the Sri Lankan website https://www.ido.lk. This issue arises from improperly configured **Cross-Origin Resource Sharing (CORS)** policies that allow untrusted domains to access protected resources on the server. Specifically, the server responds with an overly permissive Access-Control-Allow-Origin header—either set to a wildcard (*) or configured to accept arbitrary origins—thereby enabling unauthorized cross-origin requests.

This vulnerability is directly related to **OWASP Top 10 – A07:2021 – Identification and Authentication Failures** and **A05:2021 – Security Misconfiguration**, as it reflects improper implementation of security headers and access control policies, which may allow attackers to bypass origin-based protections.

If exploited, a malicious actor hosting content on an external domain could craft requests that interact with the vulnerable application as if they were a trusted origin. This can lead to severe consequences including unauthorized access to sensitive user data, session hijacking, or performing actions on behalf of a victim without their consent.

Affected Component

CORS configuration: Servers that misconfigure the Access-Control-Allow-Origin header, allowing any domain to access protected resources.

Subdomain security: When sensitive data or functions are hosted on subdomains that are improperly configured or share resources without proper restrictions.

API endpoints: Misconfigured API endpoints allowing access from external or untrusted sources.

Cookies or storage: Misconfigured cookies or session management on cross-domain requests (e.g., failing to set proper SameSite attributes).

Cloud or third-party services: Improper configurations in cloud environments or third-party services that can lead to unintended access across domains.

Impact Assessment

Unauthorized Data Access: Attackers can access sensitive data or resources that are meant to be restricted to specific domains or users.

Cross-Site Request Forgery (CSRF): Malicious websites could make requests on behalf of users to perform actions they did not authorize, particularly if authentication or session data is available across domains.

Session Hijacking: If cross-domain requests are improperly configured, an attacker could exploit the vulnerability to steal cookies, session tokens, or authentication headers from users.

Identity Theft: Attackers could gain access to personal data, login credentials, or other sensitive information that should be restricted to specific origins.

Reputation Damage: Organizations may face a loss of trust from users or clients if this vulnerability is exploited publicly. Regulatory bodies may also impose penalties for improper handling of cross-origin requests.

Financial Loss: Exploitation of cross-domain vulnerabilities in banking or e-commerce applications could lead to financial fraud or theft.

Steps to Reproduce

Step 01: Identify Cross-Domain Resources:

Review the application to find resources or API endpoints that are accessed across different domains or subdomains.

Look for AJAX requests, API calls, or embedded content from third-party domains.

Step 02: Check CORS Configuration:

Using browser developer tools (e.g., Chrome DevTools), examine the **HTTP response headers** from requests to external resources or APIs.

Look for the Access-Control-Allow-Origin header. If the value is * or allows domains you don't trust, the CORS configuration is misconfigured.

Step 03: Test Malicious Domain Access:

Create a malicious webpage hosted on a different domain or subdomain. You can use simple HTML and JavaScript to make requests to the vulnerable domain.

Code

```
fetch('http://ido-api.com/api/secret-data', {
method: 'GET',
 headers: {
                'Authorization': 'Bearer victim-token'
         }
      })
.then(response => response.json())
.then(data => console.log(data));
```

If the server is vulnerable, it will respond with sensitive data, even though the request is made from an untrusted domain.

Step 04: Check for Insecure Cookies:

Inspect cookies and authentication headers using browser developer tools to ensure they are being sent across domains properly and securely.

Check if the cookies are marked with **SameSite=Strict** or **SameSite=Lax**. If **SameSite=None** is used without Secure, cookies may be exposed to cross-domain attacks.

Step 05: Verify Access Control in APIs:

Review API endpoints to ensure that they do not accept requests from untrusted origins. If an API lacks proper authentication or authorization checks, it may allow attackers to perform unauthorized actions.

Try sending a crafted request from a malicious domain using tools like **Postman** or **Burp Suite** to check if the server is vulnerable to cross-domain requests.

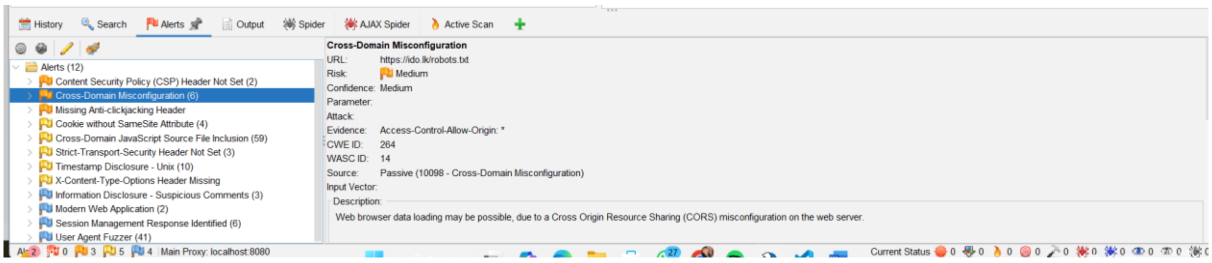
Step 06: Subdomain Misconfiguration:

If the application has subdomains that share resources, verify that these subdomains are properly isolated and secured.

Ensure that sensitive operations on one subdomain do not inadvertently allow cross-domain access from other subdomains.

Proof of Concept (Screenshots)

OWASP ZAP Scan



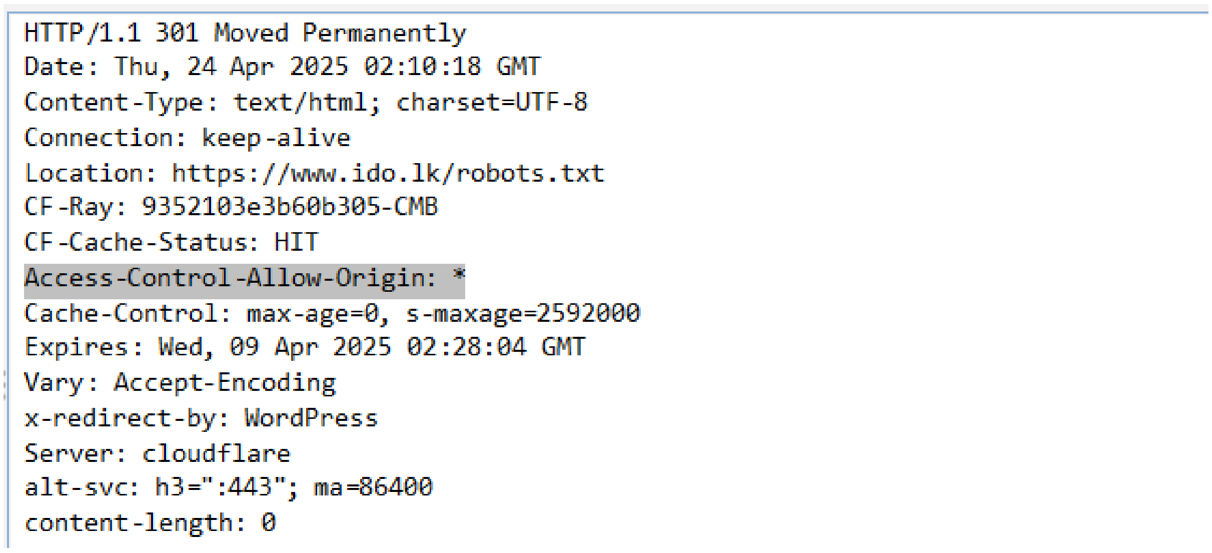
```
Alerts (12)
  Content Security Policy (CSP) Header Not Set (2)
  Cross-Domain Misconfiguration (6)
  Missing Anti-clickjacking Header
  Cookie without SameSite Attribute (4)
  Cross-Domain JavaScript Source File Inclusion (59)
  Strict-Transport-Security Header Not Set (3)
  Timestamp Disclosure - Unix (10)
  X-Content-Type-Options Header Missing
  Information Disclosure - Suspicious Comments (3)
  Modern Web Application (2)
  Session Management Response Identified (6)
  User Agent Fuzzer (41)

Cross-Domain Misconfiguration
URL:          https://ido.lk/robots.txt
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.

0  3  5  4  Main Proxy: localhost:8080
```

The screenshot shows that OWASP ZAP detected a Cross-Domain Misconfiguration on the site, where the server allows requests from any origin (Access-Control-Allow-Origin: *). This weak CORS policy can let attackers access sensitive data from external domains, posing a medium security risk.

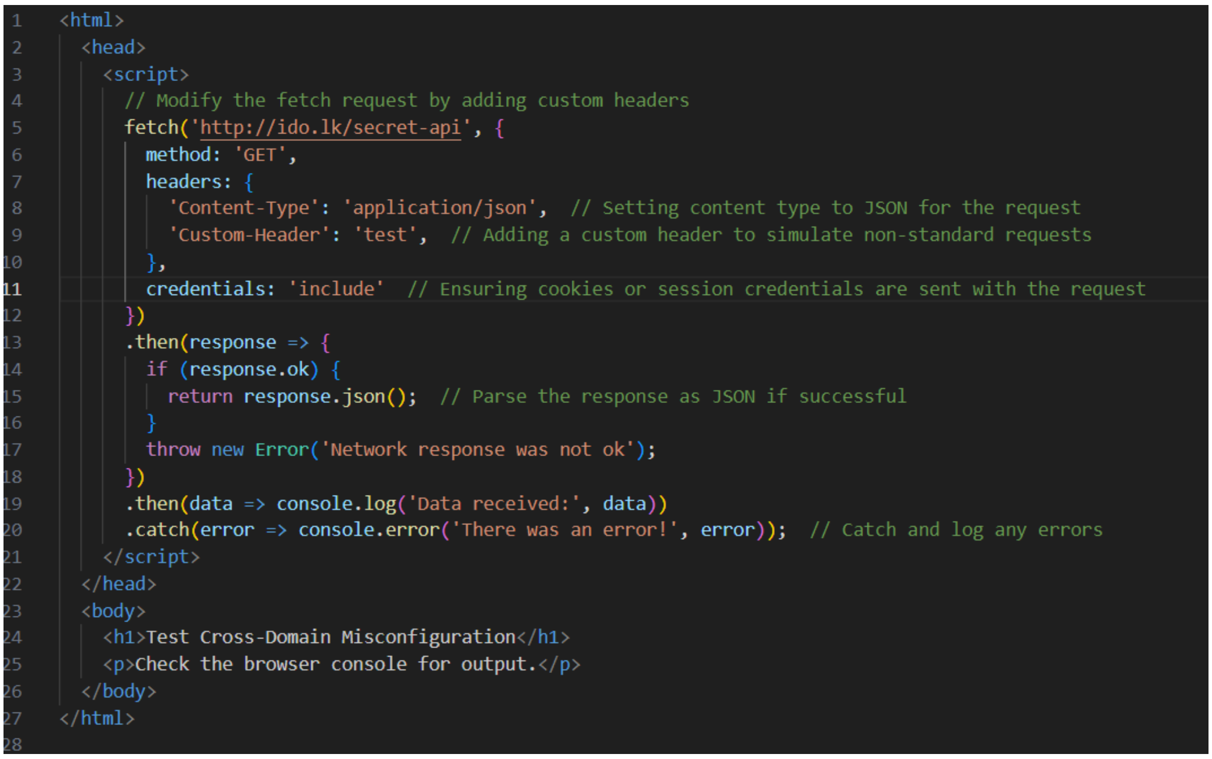
Web Developer Tools



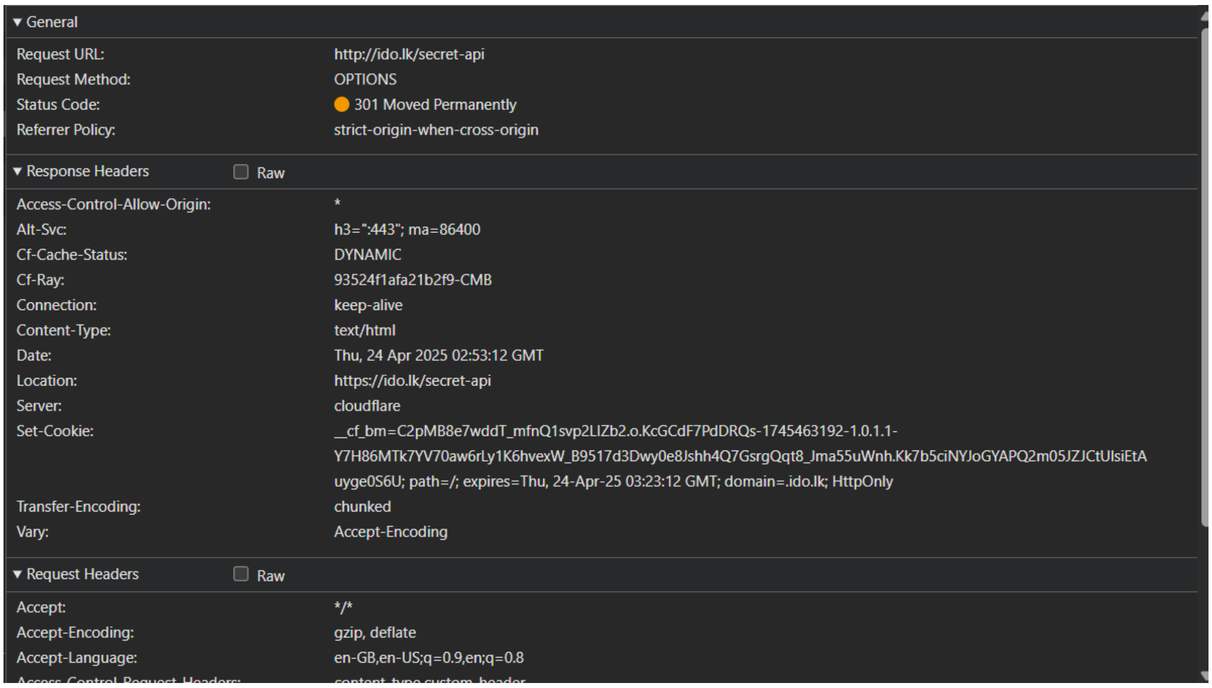
```
HTTP/1.1 301 Moved Permanently
Date: Thu, 24 Apr 2025 02:10:18 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
Location: https://www.ido.lk/robots.txt
CF-Ray: 9352103e3b60b305-CMB
CF-Cache-Status: HIT
Access-Control-Allow-Origin: *
Cache-Control: max-age=0, s-maxage=2592000
Expires: Wed, 09 Apr 2025 02:28:04 GMT
Vary: Accept-Encoding
x-redirect-by: WordPress
Server: cloudflare
alt-svc: h3=":443"; ma=86400
content-length: 0
```

This header allows any external domain to access the resource, which is a CORS misconfiguration. Even though the request is redirected, the presence of Access-Control-Allow-Origin: * indicates that the server is improperly configured to trust all origins, making it vulnerable to cross-origin attacks such as data theft or unauthorized API access.

Use Malicious Script



```html
<html>
  <head>
    <script>
      // Modify the fetch request by adding custom headers
      fetch('http://ido.lk/secret-api', {
        method: 'GET',
        headers: {
          'Content-Type': 'application/json',  // Setting content type to JSON for the request
          'Custom-Header': 'test',  // Adding a custom header to simulate non-standard requests
        },
        credentials: 'include'  // Ensuring cookies or session credentials are sent with the request
      })
      .then(response => {
        if (response.ok) {
          return response.json();  // Parse the response as JSON if successful
        }
        throw new Error('Network response was not ok');
      })
      .then(data => console.log('Data received:', data))
      .catch(error => console.error('There was an error!', error));  // Catch and log any errors
    </script>
  </head>
  <body>
    <h1>Test Cross-Domain Misconfiguration</h1>
    <p>Check the browser console for output.</p>
  </body>
</html>
```



```
▼ General
Request URL:                     http://ido.lk/secret-api
Request Method:                  OPTIONS
Status Code:                     ● 301 Moved Permanently
Referrer Policy:                 strict-origin-when-cross-origin

▼ Response Headers    ☐ Raw
Access-Control-Allow-Origin:     *
Alt-Svc:                         h3=":443"; ma=86400
Cf-Cache-Status:                 DYNAMIC
Cf-Ray:                          93524f1afa21b2f9-CMB
Connection:                      keep-alive
Content-Type:                    text/html
Date:                            Thu, 24 Apr 2025 02:53:12 GMT
Location:                        https://ido.lk/secret-api
Server:                          cloudflare
Set-Cookie:                      __cf_bm=C2pMB8e7wddT_mfnQ1svp2LIZb2.o.KcGCdF7PdDRQs-1745463192-1.0.1.1-
                                 Y7H86MTk7YV70aw6rLy1K6hvexW_B9517d3Dwy0e8Jshh4Q7GsrgQqt8_Jma55uWnh.Kk7b5ciNYJoGYAPQ2m05JZJCtUIsiEtA
                                 uyge0S6U; path=/; expires=Thu, 24-Apr-25 03:23:12 GMT; domain=.ido.lk; HttpOnly
Transfer-Encoding:               chunked
Vary:                            Accept-Encoding

▼ Request Headers    ☐ Raw
Accept:                          */*
Accept-Encoding:                 gzip, deflate
Accept-Language:                 en-GB,en-US;q=0.9,en;q=0.8
```

Key Details:

1. **301 Moved Permanently:** This means the resource has been moved to a new URL permanently, and the correct URL for the resource is https://ido.lk/secret-api.

2. ****Access-Control-Allow-Origin:* :** The CORS header still allows any origin to access the resource, which is a good sign if you're testing for cross-domain misconfigurations

3. **Location Header**: This is the important part of the response:

   ***Location:* https://ido.lk/secret-api**

   This tells you that the resource has been moved from http://ido.lk/secret-api to https://ido.lk/secret-api.

Proposed Mitigation or Fix

Proper CORS Configuration:

- Configure **CORS headers** to only allow trusted origins. Avoid using a wildcard (*) for Access-Control-Allow-Origin. Instead, list specific domains that are allowed to interact with your application.
- Additionally, restrict HTTP methods (e.g., GET, POST, PUT) that can be used in cross-domain requests by setting the Access-Control-Allow-Methods header.

Use SameSite Cookies:

- Ensure cookies are properly configured with SameSite=Strict or SameSite=Lax to limit cross-site cookie usage.
- Set the Secure flag to ensure cookies are only sent over HTTPS.

Subdomain Isolation:

- Avoid sharing sensitive resources or authentication mechanisms between subdomains unless absolutely necessary. If sharing resources is required, use strict **CORS settings** and **subdomain-specific access controls** to prevent unauthorized access.
- Implement **subdomain isolation** by securing APIs and sensitive data endpoints against access from untrusted subdomains.

Implement Origin and Referer Header Checks:

- Validate the **Origin** and **Referer** headers to ensure that requests are coming from authorized domains.
- Reject requests with missing or invalid Origin/Referer headers, particularly when dealing with sensitive operations or APIs.

Enforce Authentication and Authorization:

- Ensure that sensitive operations and API endpoints require proper authentication and authorization, checking tokens or session cookies before processing requests.

- Implement proper **OAuth2** or **JWT** token handling, ensuring tokens are bound to specific domains or client applications.

Conclusion

A Cross-Domain Misconfiguration vulnerability was identified on https://www.ido.lk, allowing unauthorized access from external domains due to insecure CORS settings. This issue could lead to data leaks or unauthorized actions. Immediate corrective measures, such as restricting CORS to trusted domains, are recommended to secure the application.