SLIIT UNI
THE KNOWLEDGE UNIVERSITY

Sri Lanka Institute of Information Technology

Bug Bounty Report - 04

Module: IE2062

Web Security

Year 2, Semester 2

Aazaf Ritha. J – IT23151710

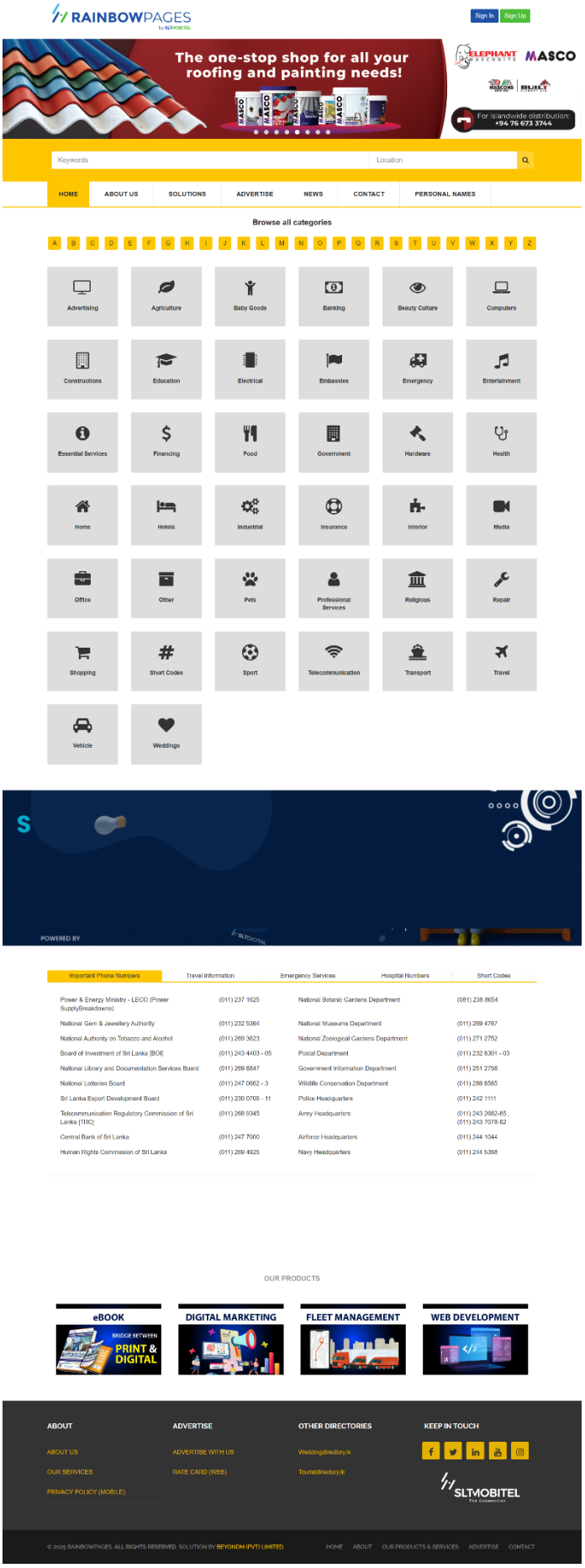
B.Sc. (Hons) in Information Technology

Specialized in Cyber Security

Table of Contents

Introduction

This report outlines a vulnerability discovered on a Sri Lankan e-commerce website, https://meatmart.lk. During security testing, a Clickjacking vulnerability was identified on multiple pages of the site. The issue was confirmed through both manual and automated methods, and this report provides technical details, reproduction steps, and mitigation recommendations.


RAINBOWPAGES
Sign In
Sign Up
The one-stop shop for all your roofing and painting needs!
ELEPHANT HOUSE CEMENTS
MASCO
For islandwide distribution: +94 76 673 3744
Keywords
Location
HOME
ABOUT US
SOLUTIONS
ADVERTISE
NEWS
CONTACT
PERSONAL NAMES
Browse all categories
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Advertising
Agriculture
Baby Goods
Banking
Beauty Culture
Computers
Constructions
Education
Electrical
Embassies
Emergency
Entertainment
Essential Services
Financing
Food
Government
Hardware
Health
Home
Hotels
Industrial
Insurance
Interior
Media
Office
Other
Pets
Professional Services
Religious
Repair
Shopping
Short Codes
Sport
Telecommunication
Transport
Travel
Vehicle
Weddings
POWERED BY
Important Phone Numbers
Travel Information
Emergency Services
Hospital Numbers
Short Codes
Power & Energy Ministry - LECO (Power Supply/Breakdowns) (011) 237 1625
National Gem & Jewellery Authority (011) 232 5364
National Authority on Tobacco and Alcohol (011) 269 3623
Board of Investment of Sri Lanka [BOI] (011) 243 4400 - 05
National Library and Documentation Services Board (011) 269 8847
National Lotteries Board (011) 247 0662 - 3
Sri Lanka Export Development Board (011) 230 0705 - 11
Telecommunication Regulatory Commission of Sri Lanka [TRC] (011) 268 9345
Central Bank of Sri Lanka (011) 247 7000
Human Rights Commission of Sri Lanka (011) 269 4925
National Botanic Gardens Department (081) 238 8654
National Museums Department (011) 269 4767
National Zoological Gardens Department (011) 271 2752
Postal Department (011) 232 8301 - 03
Government Information Department (011) 251 2758
Wildlife Conservation Department (011) 288 8585
Police Headquarters (011) 242 1111
Army Headquarters (011) 243 2682-85 , (011) 243 7078-82
Airforce Headquarters (011) 244 1044
Navy Headquarters (011) 244 5368
OUR PRODUCTS
eBOOK
BRIDGE BETWEEN PRINT & DIGITAL
DIGITAL MARKETING
FLEET MANAGEMENT
WEB DEVELOPMENT
ABOUT
ABOUT US
OUR SERVICES
PRIVACY POLICY (MOBILE)
ADVERTISE
ADVERTISE WITH US
RATE CARD (WEB)
OTHER DIRECTORIES
Weddingdirectory.lk
Touristdirectory.lk
KEEP IN TOUCH
SLTMOBITEL
© 2025 RAINBOWPAGES. ALL RIGHTS RESERVED. SOLUTION BY BEYONDM (PVT) LIMITED
HOME ABOUT OUR PRODUCTS & SERVICES ADVERTISE CONTACT

Vulnerability Title

Title – Open Redirect Vulnerability

Risk Level- High

Domain - https://rainbowpages.lk/

Description

An Open Redirect vulnerability occurs when a web application accepts unvalidated input that specifies a URL to which users are redirected. In this case, the vulnerable endpoint /out.php on the domain rainbowpages.lk accepts a user-controlled parameter url and redirects users to that value without proper validation or sanitization.

This vulnerability is dangerous because it allows attackers to craft links that appear to originate from the trusted rainbowpages.lk domain but actually lead to external, potentially malicious websites. This can be exploited for phishing attacks, malware delivery, and social engineering.

This issue falls under the OWASP Top 10 – A01:2021 Broken Access Control and is also related to A10:2021 – Server-Side Request Forgery (SSRF) in broader contexts, but most directly aligns with unvalidated redirects and forwards, which are part of OWASP's broader concern regarding improper input validation and lack of output encoding.

The vulnerability arises due to:

- Lack of input validation: The application does not verify that the url parameter points to a trusted or internal domain.
- Direct use of user input in redirects: The value of url is used directly in a Location header, enabling arbitrary redirection.

If exploited, this can lead users to believe they are visiting a trusted site, thereby increasing the success rate of phishing campaigns or credential harvesting attempts.

Affected Component

URL redirection handlers

Login/logout redirection pages

Endpoint: /out.php

Query Parameter: url

Component Type: Redirection handler

Impact Assessment

Security Risks: Attackers can craft malicious links to redirect users to phishing sites or sites that deliver malware, thereby compromising user data (like credentials, personal info, etc.).

Trust Issues: Users might be tricked into trusting a website that they thought was legitimate.

Reputation Damage: If exploited in a public-facing web application, the company's reputation could suffer due to user trust issues and potential breaches.

Financial Impact: Successful exploitation may lead to phishing attacks, account takeovers, or other financial fraud scenarios.

Steps to Reproduce

Step 01: Set up Burp Suite:

- Open **Burp Suite** and ensure the **Proxy** is enabled.
- Configure your browser to use **Burp Suite** as the proxy. This is typically done by setting your browser's proxy settings to **127.0.0.1** and port **8080**.

Step 02: Intercept the Request:

- Visit the vulnerable web application in your browser (the one that might be susceptible to open redirects).
- Perform the action that triggers the redirect functionality.
- Burp Suite will intercept this HTTP request if the **Intercept** function is turned on under the **Proxy** tab.

Step 03: Capture the Redirect URL in Burp Suite:

- Once you trigger a redirect, Burp Suite will show the HTTP request and response in the **Intercept** tab.
- Look for a parameter that might control the redirect. Common parameters include redirect_url, next, return_to, etc.

Step 04: Modify the URL for Malicious Redirect:

- Right-click on the intercepted HTTP request and click **"Send to Repeater"**. This will allow you to modify the request and send it multiple times.
- In the **Repeater** tab, locate the URL parameter (url=) and modify its value to a malicious site.

Step 05: Send the Request:

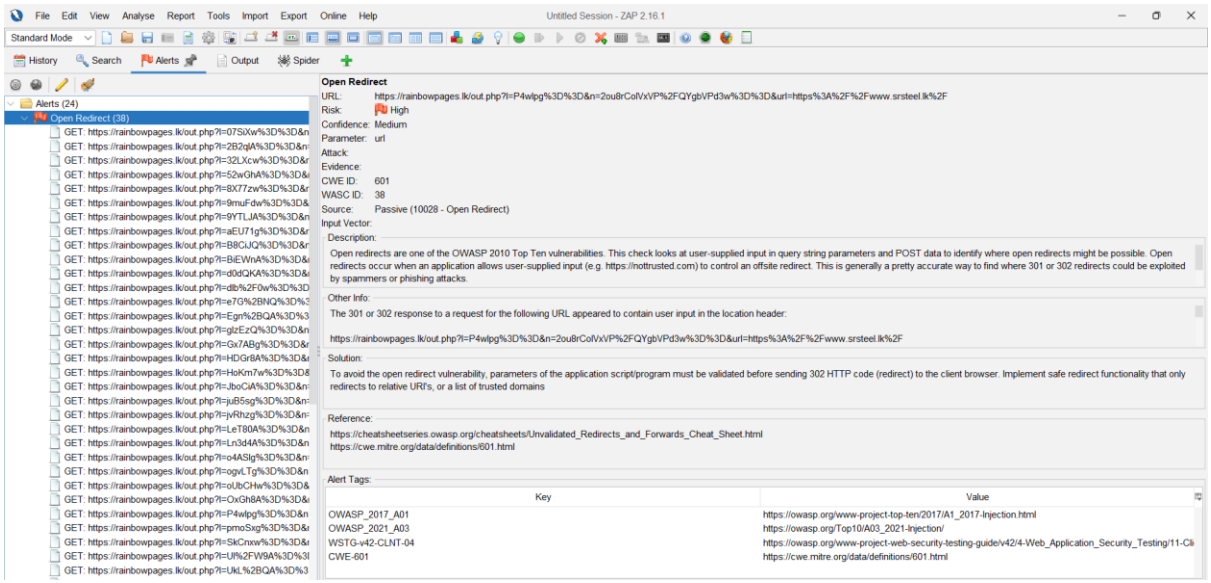
- In the **Repeater** tab, click **"Send"** to send the modified request to the server.
- If the vulnerability exists, the server should redirect you to http://malicious-website.com instead of the original destination.

Step 06: Verify the Redirect:

- Once the request is sent, verify that the browser is redirected to the malicious website.
- If the redirection occurs, the application is vulnerable to **Open Redirect**.

Proof of Concept (Screenshots)

OWSAP ZAP Scan



This screenshot is taken from **OWASP ZAP** and shows a high-risk **Open Redirect vulnerability** detected on the https://rainbowpages.lk/out.php endpoint. The scanner confirms that the url parameter can be manipulated to redirect users to external websites (e.g., https://www.srsteel.lk). The alert description links the issue to the **OWASP Top 10** and includes the evidence of the vulnerable URL, highlighting that no validation is being performed on the redirect target.

Vulnerable URL

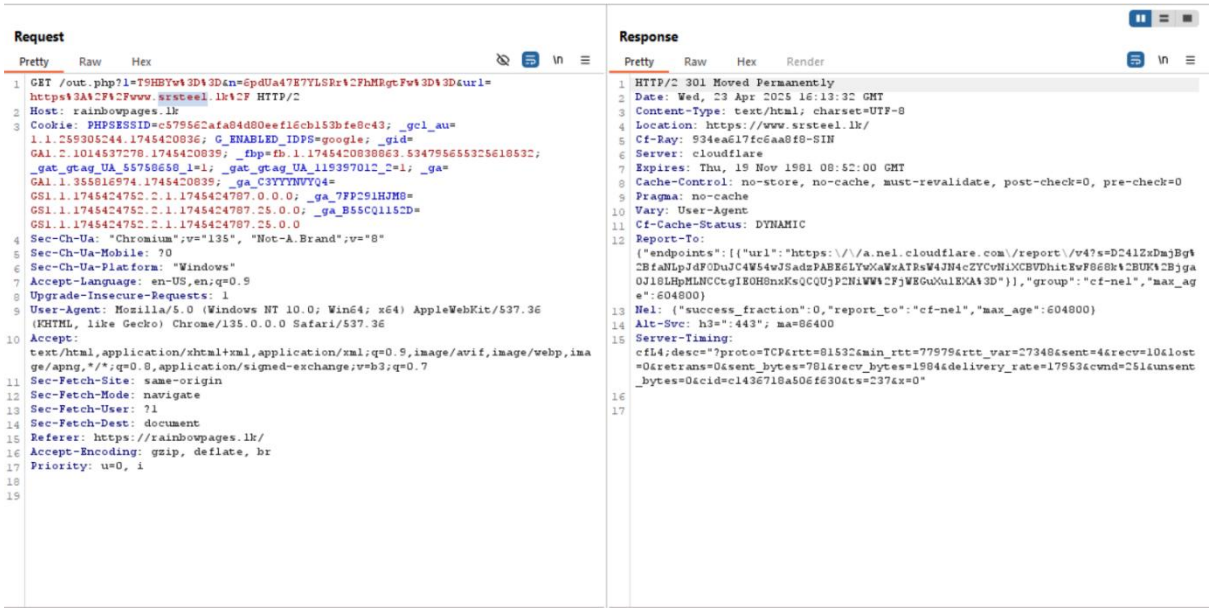
/out.php?l=7N10xA%3D%3D&n=n1CGss7sSbAR%2FpmPLYUJJQ%3D%3D&url=https%3A%2F%2Fwww.srsteel.lk%2F HTTP/2

Decoded - https://rainbowpages.lk/out.php?url=https://www.srsteel.lk

Burp Suite Interception Request and response

The following screenshot from Burp Suite demonstrates the intercepted GET request and corresponding HTTP response for the vulnerable endpoint.

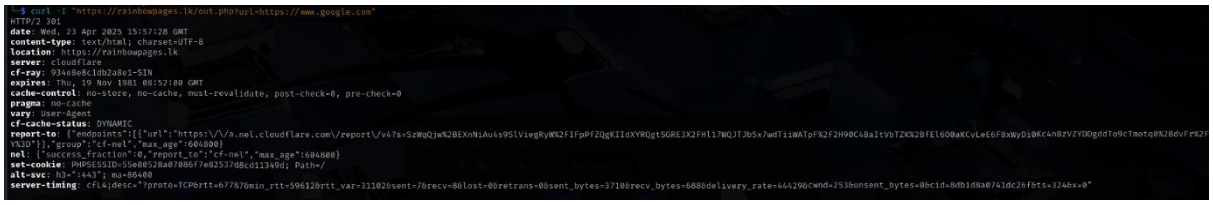
This Burp Suite capture shows the HTTP **request and response** for the vulnerable /out.php endpoint on rainbowpages.lk. The request includes an url parameter pointing to https://www.srsteel.lk, and the server responds with a **301 Moved Permanently** status and a Location header set to the attacker-controlled URL. This confirms that the application blindly redirects users to any external domain without validation—demonstrating a clear case of an **Open Redirect vulnerability**.



Poc (GET request)

curl -I "https://rainbowpages.lk/out.php?url=https://www.google.com"

Response



What This Confirms

- The 302 status code means **Found (Temporary Redirect)**.
- The Location header contains the URL you passed in the url parameter.
- No validation is being done to check whether https://www.google.com/ is a safe or internal URL.

Proposed Mitigation or Fix

URL Validation: Ensure that any URL used for redirection is validated to only redirect users to trusted domains. Implement a whitelist of allowed domains or specific URLs for redirection.

Use Relative Paths for Redirection: Instead of using full URLs for redirection, use relative paths (e.g., /dashboard) which are inherently safer.

Use HTTP Referer Header Checks: For sensitive operations, verify that the redirect originates from a valid source page by checking the Referer header.

Encode and Escape User Inputs: Make sure user inputs are sanitized by encoding or escaping before using them in redirect URLs.

Tokenization: Generate a unique token for each redirect that can be matched with a list of allowed redirections on the server side.

Notify Users: When performing redirection, provide clear notifications to the user about where they are being redirected, especially if the redirect destination is external.

Conclusion

The open redirect vulnerability identified in rainbowpages.lk poses a significant security risk by allowing attackers to redirect users to untrusted, malicious websites. This issue can be exploited for phishing, malware distribution, and other social engineering attacks. Addressing this vulnerability through proper input validation and implementing a whitelist of safe redirect targets is essential to protect users and maintain the integrity and trustworthiness of the platform.