**Sri Lanka Institute of Information Technology**

**IE2062 - Web Security**

**Final Assignment**

**Bug Bounty Report 09**

**Name:** Peiris W.S.S.N

**Registration Number:** IT23227286

Contents

[1. Introduction 3](#_Toc196572244)

[2. Reconnaissance 4](#_Toc196572245)

[3. Vulnerability 6](#_Toc196572246)

[4. Vulnerability description 6](#_Toc196572247)

[5. Affected Components 6](#_Toc196572248)

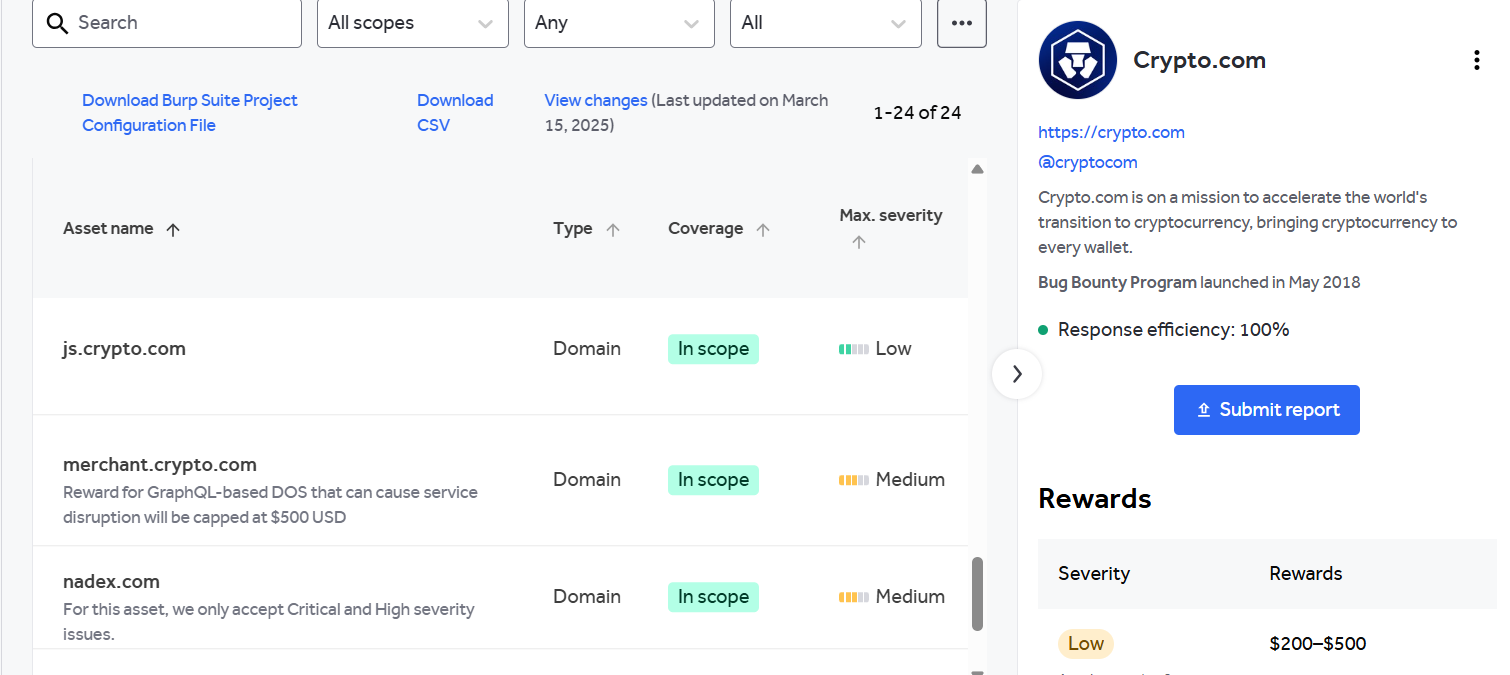
[6. Impact Assessment 7](#_Toc196572249)

[7. Steps to reproduce 7](#_Toc196572250)

[8. Proof of concept 8](#_Toc196572251)

[9. Proposed mitigation or fix 8](#_Toc196572252)

# Introduction



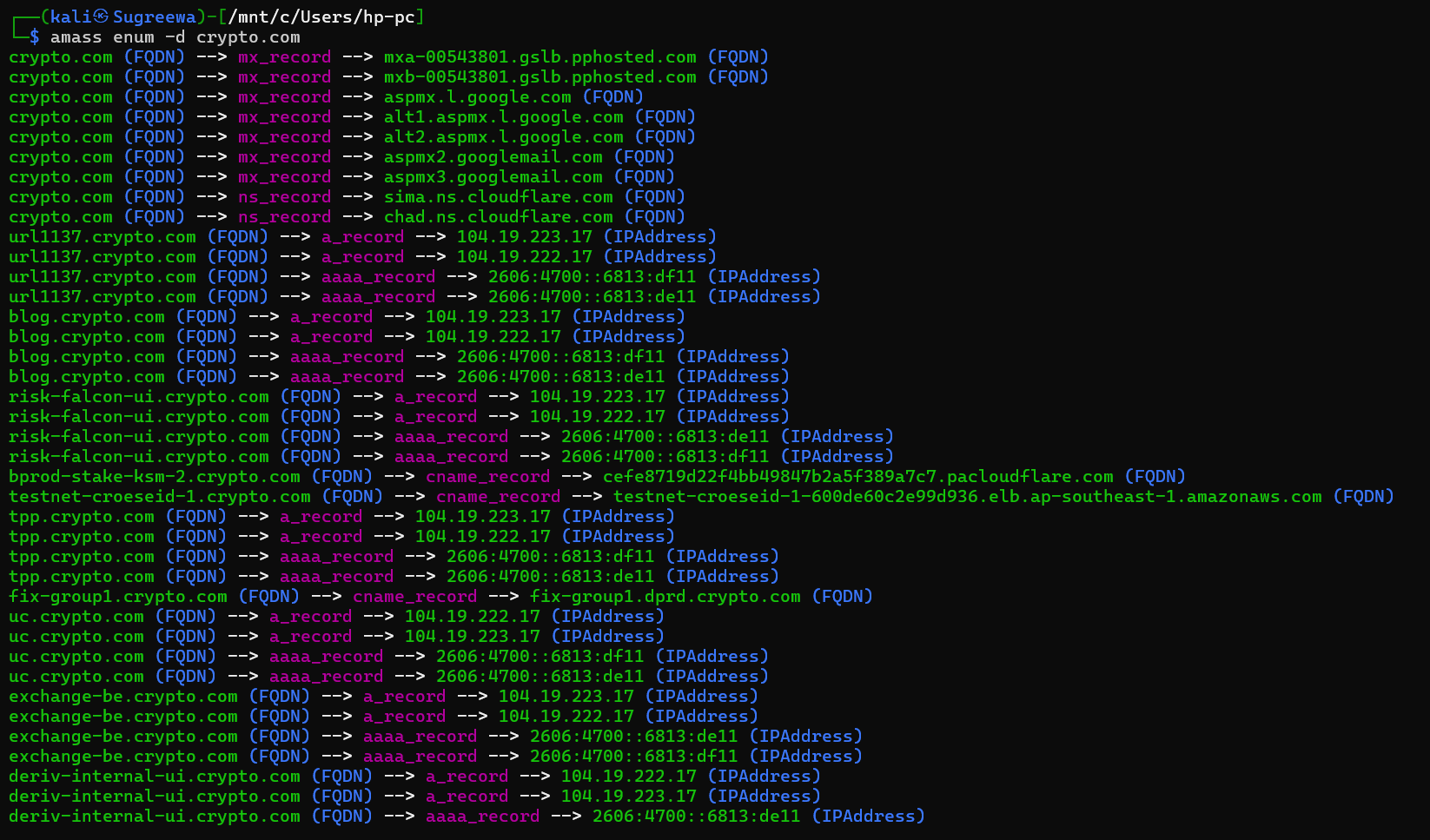
**Website:** <https://crypto.com/>

**Sub-domain:** <https://merchant.crypto.com/users/sign_in>

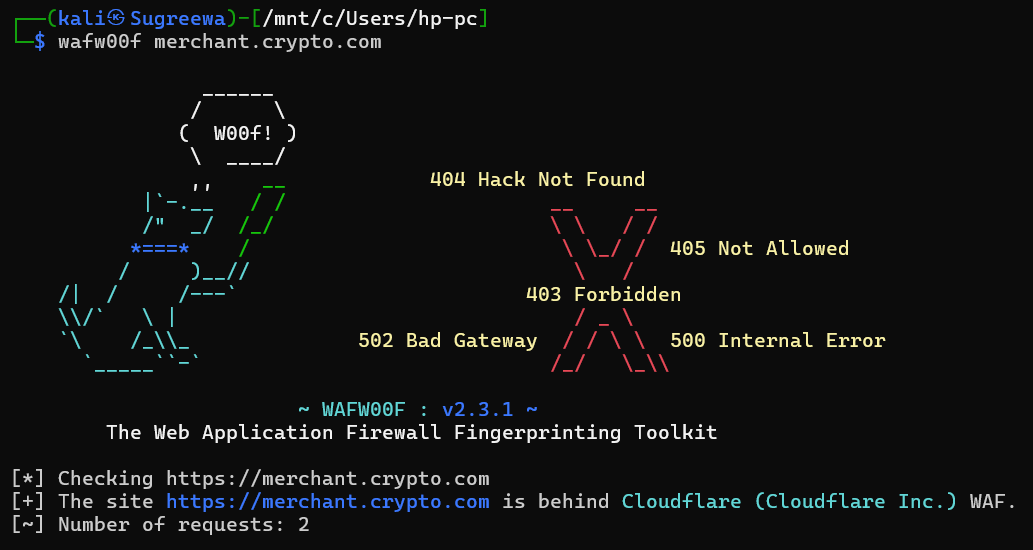
**Listed by:** Crypto.com

# Reconnaissance

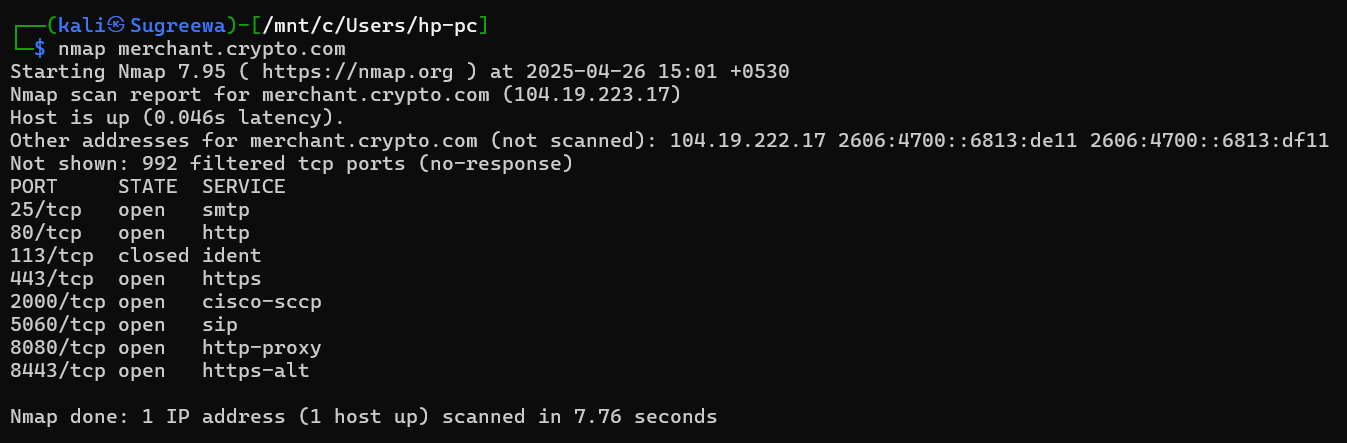
* **Subdomain enumeration using Amass**

****

* **Firewall detection**

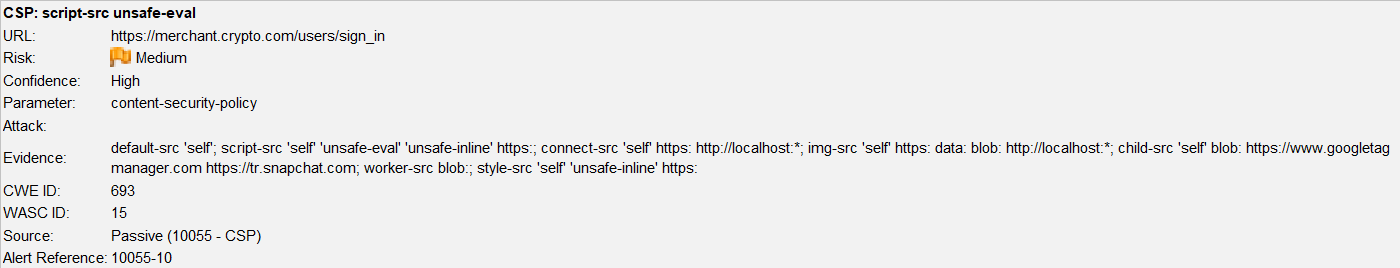
****

* **Nmap Scan**

****

# Vulnerability

* **CSP: script-src unsafe-eval**

****

# Vulnerability description

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) Cross Site Scripting (XSS), and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.

1. Affected Components

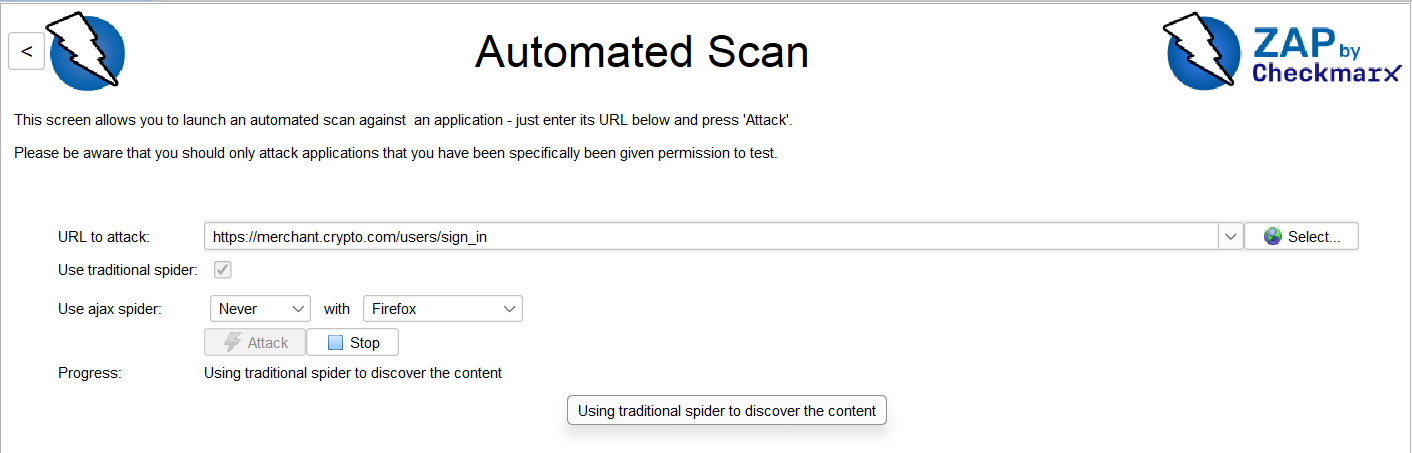
* **Component:** HTTP Response Header → Content-Security-Policy
* **Current Policy:**
  + script-src 'self' 'unsafe-eval' 'unsafe-inline' https:;
* **Directive Affected:** script-src
* **Risk Element:** Presence of 'unsafe-eval'

# Impact Assessment

* **Risk Level:** High
* **Potential Impacts:**
  + Enables execution of injected scripts using eval()-based functions
  + Increases susceptibility to XSS, especially in apps that manipulate or parse user input dynamically
  + May allow malicious scripts to bypass filtering mechanisms
  + Violates strong CSP enforcement, failing compliance checks (e.g., PCI-DSS, OWASP)

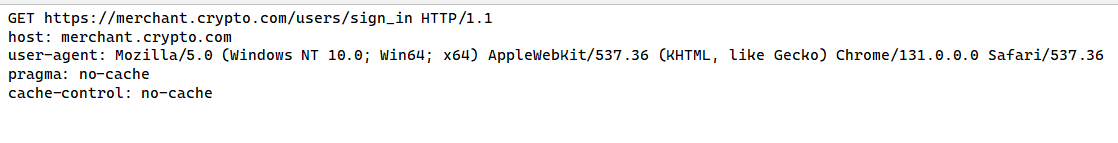
# Steps to reproduce

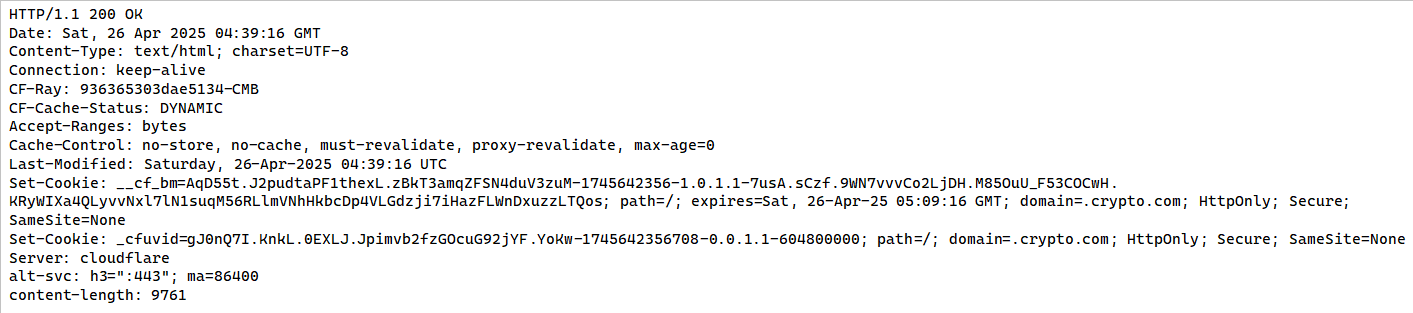
* Use browser **DevTools** or **ZAP/Burp** to inspect any page response.



* Locate the Content-Security-Policy response header.
* Confirm the presence of:
  + script-src 'self' 'unsafe-eval' ...
* (Optional) Test usage of eval() in inline or loaded scripts and verify it executes without errors.

# Proof of concept





# Proposed mitigation or fix

Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.