Sri Lanka Institute of Information Technology

Individual Assignment

Bug Bounty Report

Web Security - IE2062

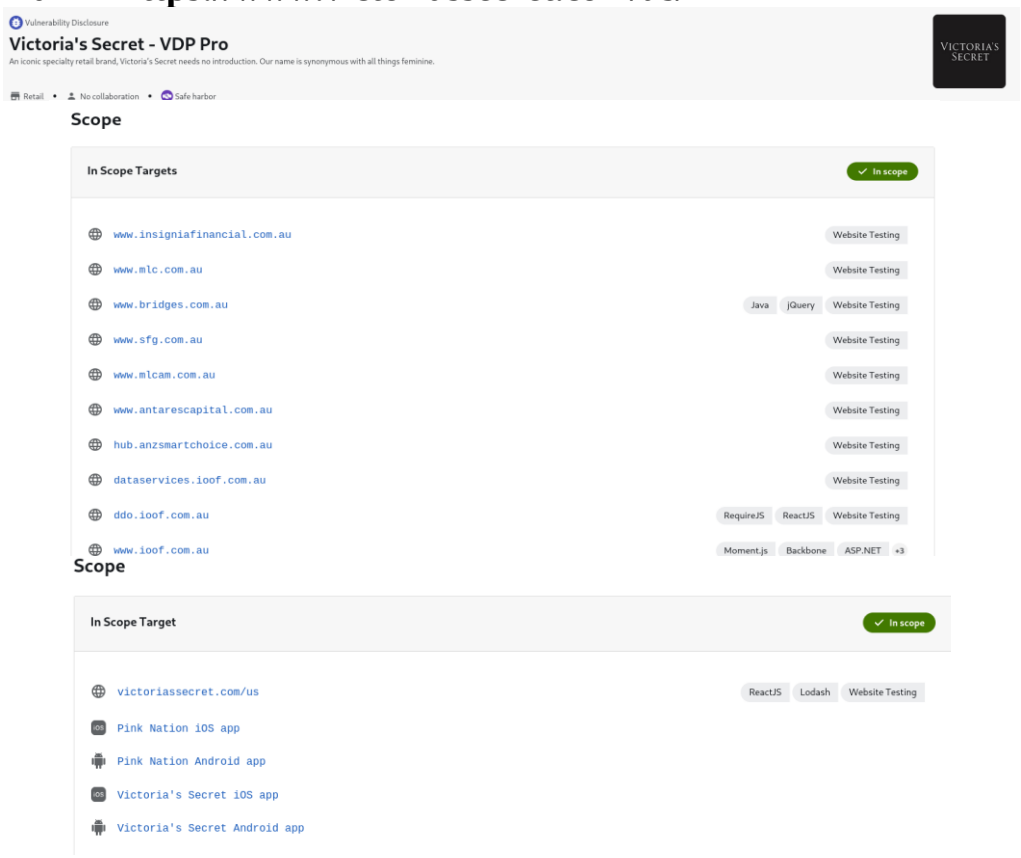
BSc Honors in Information Technology Specializing in Cyber Security

CASE STUDY NAME	BUG BOUNTY Report 08
CAMPUS/CENTER	SLIIT KANDY UNI

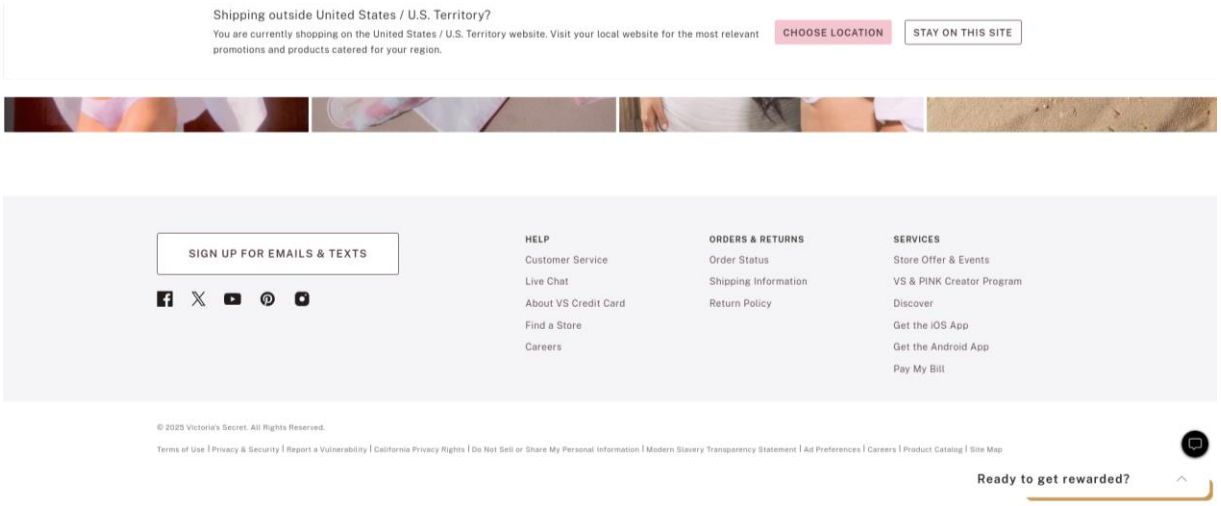
Student Details

	Student Registration Number	Student Name
1	IT23222854	JAYASINGHE B. I

Domain – https://www.victoriassecret.com/us/

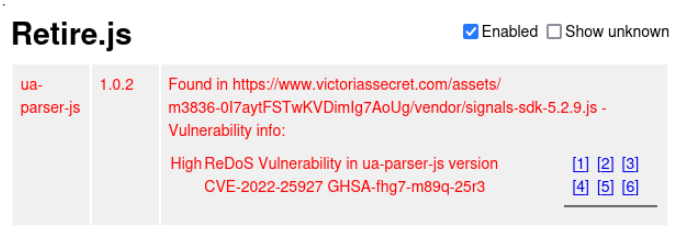


- Link - https://www.victoriassecret.com/us/
- Category – Vulnerability Disclosure Program (VDP)
- Type – Retail Company



1. Sensitive Data Exposure

1.1 Retire.js



Retire.js ☑ Enabled ☐ Show unknown

ua-parser-js	1.0.2	Found in https://www.victoriassecret.com/assets/m3836-0l7aytFSTwKVDimIg7AoUg/vendor/signals-sdk-5.2.9.js - Vulnerability info: High ReDoS Vulnerability in ua-parser-js version CVE-2022-25927 GHSA-fhg7-m89q-25r3 [1] [2] [3] [4] [5] [6]

Summary fo the above vulnerabilities.

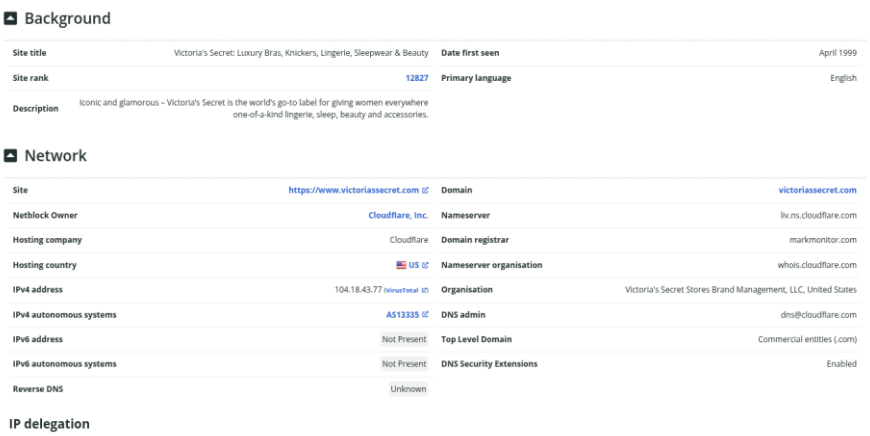
Library	Version	Vulnerability Description	Severity	Mitigation Strategy
ua-parser-js	**1.0.2**	Regular Expression Denial of Service (ReDoS)	High	Upgrade to **ua-parser-js 1.0.35 or later** for patched security.
				Limit input complexity for regex processing to prevent excessive load.
				Consider alternative parsing libraries if upgrading is not feasible.

2.1. Netcraft

The Netcraft tool provides information about a website's infrastructure including,

- Web server
- Operating system
- Hosting provider
- SSL certificate details.

Attackers can use this information for reconnaissance purposes to identify potential vulnerabilities and plan targeted attacks. Mitigation strategies include keeping software up to date, using secure hosting providers, and implementing strong SSL/TLS configurations.



Background

Site title	Victoria's Secret: Luxury Bras, Knickers, Lingerie, Sleepwear & Beauty	Date first seen	April 1999
Site rank	12827	Primary language	English
Description	Iconic and glamorous – Victoria's Secret is the world's go-to label for giving women everywhere one-of-a-kind lingerie, sleep, beauty and accessories.		

Network

Site	https://www.victoriassecret.com	Domain	victoriassecret.com
Netblock Owner	Cloudflare, Inc.	Nameserver	liv.ns.cloudflare.com
Hosting company	Cloudflare	Domain registrar	markmonitor.com
Hosting country	US	Nameserver organisation	whois.cloudflare.com
IPv4 address	104.18.43.77 (VirusTotal)	Organisation	Victoria's Secret Stores Brand Management, LLC, United States
IPv4 autonomous systems	AS13335	DNS admin	dns@cloudflare.com
IPv6 address	Not Present	Top Level Domain	Commercial entities (.com)
IPv6 autonomous systems	Not Present	DNS Security Extensions	Enabled
Reverse DNS	Unknown		

IP delegation

SSL/TLS

Assurance	Organisation validation	**Perfect Forward Secrecy**	Yes
Common name	*.victoriassecret.com	**Supported TLS Extensions**	RFC8446 key share, RFC8446 supported versions, RFC4366 server name, RFC7301 application-layer protocol negotiation, RFC4366 status request
Organisation	VS Service Company, LLC	**Application-Layer Protocol Negotiation**	h2
State	Ohio	**Next Protocol Negotiation**	Not Present
Country	US	**Issuing organisation**	Sectigo Limited
Organisational unit	Not Present	**Issuer common name**	Sectigo RSA Organization Validation Secure Server CA
Subject Alternative Name	*.victoriassecret.com, victoriassecret.com	**Issuer unit**	Not Present
Validity period	From Feb 12 2025 to Feb 12 2026 (12 months)	**Issuer location**	Salford
Matches hostname	Yes	**Issuer country**	GB
Server	cloudflare	**Issuer state**	Greater Manchester
Public key algorithm	rsaEncryption	**Certificate Revocation Lists**	http://crl.sectigo.com/SectigoRSAOrganizationValidationSecureServerCA.crl
Protocol version	TLSv1.3	**Certificate Hash**	twexOncLCNoi2P9aYNY9iWuSefw
Public key length	2048	**Public Key Hash**	e46b13b29cd8648536b7baa044f733e63c45232e55167e0d886d0896b3c4c785
Certificate check	ok	**OCSP servers**	http://ocsp.sectigo.com
Signature algorithm	sha256WithRSAEncryption	**OCSP stapling response**	Certificate valid
Serial number	0x535c2c7d7772dc8c8c38d1cada913fcb	**OCSP data generated**	Apr 27 23:43:05 2025 GMT
Cipher	TLS_AES_256_GCM_SHA384	**OCSP data expires**	May 4 23:43:04 2025 GMT
Version number	0x02		

Certificate Transparency

Signed Certificate Timestamps (SCTs)

Source	Log	Timestamp	Signature Verification
Certificate	*Unknown* lpdkv1VYl63304doNwhCd+nwOtX2pPM2bkakPw/KqcY=	2025-02-12 09:16:52	*Unknown*
Certificate	*Unknown* GYbUxyiqb/66A294Kk08kar0LXIxD670XXBBLSVMx9Q=	2025-02-12 09:16:52	*Unknown*
Certificate	*Unknown* yzj3FYl8hKFEX1vB3fvJbvKaWc1HCmkFhbDLFMMUWOc=	2025-02-12 09:16:52	*Unknown*

SSLv3/POODLE

This site does not support the SSL version 3 protocol.

More information about SSL version 3 and the POODLE vulnerability.

Heartbleed

The site did not offer the Heartbeat TLS extension prior to the Heartbleed disclosure, and so was not exploitable.

This test does not exploit the Heartbleed vulnerability but uses information from conventional HTTPS requests. More information about Heartbleed detection.

Web Browser Targeting

Web browser targeting enables software applications to make use of specific functions of the browser as well as optimizing the application for specific browser versions.

Technology	Description	Popular sites using this technology
Strict-Transport-Security (including subdomains)	*No description*	chatgpt.com, accounts.google.com, docs.google.com
X-XSS-Protection Block	Block pages on which cross-site scripting is detected	teams.microsoft.com
Document Compatibility Mode	A meta-tag used in Internet Explorer 8 to enable compatibility mode	chat.deepseek.com
X-Content-Type-Options	Browser MIME type sniffing is disabled	
Strict Transport Security	Web security policy mechanism whereby a web server declares that complying user agents are to interact with it using only secure HTTP connections	
X-Frame-Options Same Origin	Do not allow this site to be rendered within an iframe	
Content Security Policy	Detect and mitigate attacks in the browser	

Doctype

A Document Type Declaration, or DOCTYPE, is an instruction that associates a particular SGML or XML document (for example, a webpage) with a Document Type Definition (DTD).

Technology	Description	Popular sites using this technology
HTML5	Latest revision of the HTML standard, the main markup language on the web	www.amazon.com, webmail.vinccihoteles.com, mail.google.com

HTML 5

HTML5 is a markup language for structuring and presenting content for the World Wide Web and a core technology of the Internet. It is the fifth revision of the HTML standard.

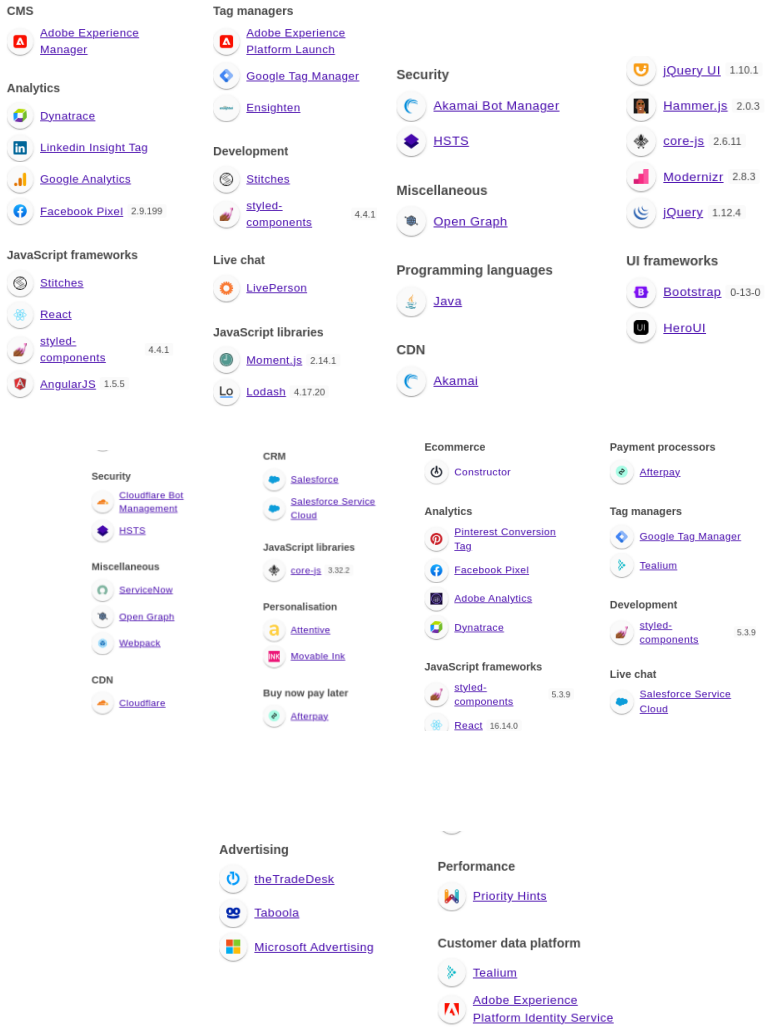
CSS Usage

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language (such as XHTML).

Technology	Description	Popular sites using this technology
External	Styles defined within an external CSS file	mail.yahoo.com
Embedded	Styles defined within a webpage	www.amazon.it, www.amazon.es, www.amazon.ca

2.2 Wappalyzer

Here are the results of the Wappalyzer detect

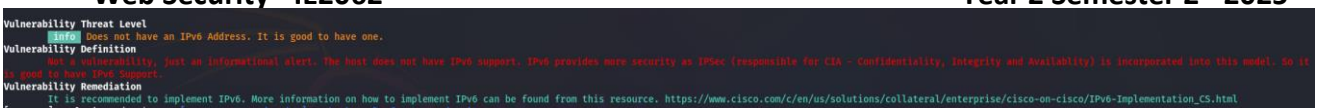


2. Multi Tool Web Vulnerability Scanner

2.1 Rapidscan

```
┌──(binosh㉿BINZ)-[~/Desktop/WS Assingment/Tools]
└─$ cd rapidscan/

┌──(binosh㉿BINZ)-[~/Desktop/WS Assingment/Tools/rapidscan]
└─$ sudo python3 rapidscan.py https://www.victoriassecret.com/us/
```

Out of 80 vulnerabilities checked for https://www.victoriassecret.com/us/ **4 vulnerabilities were detected**



2.2 OWASP

Detected Vulnerability: CSP Header Not Set

The missing Content-Security-Policy Header introduces a medium security risk, making the application vulnerable to Cross-Site Scripting (XSS) and content injection threats. Immediate implementation of CSP is recommended to enforce security measures and reduce exposure to attacks.

- Risk Level: Medium
- Confidence Level: High
- Vulnerability Type: Missing Content Security Policy Header
- CWE ID: 693 – Protection Mechanism Failure
- WASC ID: 15 – Application Misconfiguration
- Alert Reference: 10038-1

Details of Vulnerability

- Key Parameter Affected: Content-Security-Policy
- Attack Vector: None
- Evidence: CSP missing from HTTP headers
- Source: Passive scan (10038 - CSP Header Not Set)
- Issue Description: The web server lacks a CSP header, increasing risks related to content injection and malicious code execution

Impact Assessment

Potential Risk

- Increased vulnerability to Cross-Site Scripting (XSS) attacks
- Higher risk of Clickjacking and unauthorized content framing
- Possible exposure to data injection attacks

Recommended Remediation

Short-Term Fixes

- Define and set a Content-Security-Policy header to restrict allowed content sources
- Enforce strict policies to block unauthorized JavaScript and iframe embedding

Long-Term Fixes

- Conduct periodic security audits to validate CSP configurations
- Strengthen CSP directives such as default-src 'none' for improved protection
- Implement secure response headers to mitigate common web attacks

Content Security Policy (CSP) Header Not Set

URL: https://www.victoriassecret.com/us/
Risk: Medium
Confidence: High
Parameter:
Attack:
Evidence:
CWE ID: 693
WASC ID: 15
Source: Passive (10038 - Content Security Policy (CSP) Header Not Set)
Alert Reference: 10038-1
Input Vector:

Description:

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including 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,

The missing Content-Security-Policy Header introduces a medium security risk, increasing exposure to Cross-Site Scripting (XSS) and unauthorized resource loading vulnerabilities. Immediate CSP implementation is highly recommended to strengthen security defenses.

Detected Vulnerability: CSP Header Not Set

- **Risk Level:** Medium
- **Confidence Level:** High
- **Vulnerability Type:** Missing Content Security Policy Header
- **CWE ID:** 693 – Protection Mechanism Failure
- **WASC ID:** 15 – Application Misconfiguration
- **Alert Reference:** 10309-3

Details of Vulnerability

- **Key Parameter Affected:** Content-Security-Policy
- **Attack Vector:** Not applicable
- **Evidence:** CSP missing from HTTP headers
- **Source:** Passive scan (10309 - CSP Header Not Set)
- **Issue Description:** CSP is an additional security layer that helps detect and mitigate certain types of attacks, including **Cross-Site Scripting (XSS)** and **data injection attacks**. The absence of this header increases the risk of **malicious content execution and unauthorized resource loading**.

Impact Assessment

Potential Risk

- Increased vulnerability to **Cross-Site Scripting (XSS)** attacks
- Higher risk of **Clickjacking** and unauthorized content framing
- Possible exposure to **data injection attacks and unauthorized scripts**

Recommended Remediation

Short-Term Fixes:

- Implement and set a **Content-Security-Policy** header to restrict allowed content sources
- Define explicit directives to block unauthorized script execution
- Use frame ancestors 'none' to prevent Clickjacking risks

Long-Term Fixes

- Conduct periodic **CSP audits** to validate configurations
- Strengthen CSP directives such as default-src 'none' for improved protection
- Implement **secure response headers** to mitigate common web attacks

Detected Vulnerability: Anti-Clickjacking Header Not Set

The absence of anti-clickjacking headers introduces a medium security risk, potentially allowing malicious UI manipulations that could lead to fraudulent interactions and user deception. Immediate remediation through X-Frame-Options and CSP enhancements is advised to mitigate vulnerability.

- **Risk Level:** Medium
- **Confidence Level:** Medium
- **Vulnerability Type:** Lack of Clickjacking Protection
- **CWE ID:** 1021 – Clickjacking
- **WASC ID:** 15 – Application Misconfiguration
- **Alert Reference:** 10020-1

Details of Vulnerability

- **Key Parameter Affected:** X-Frame-Options
- **Attack Type:** Clickjacking
- **Evidence:** Missing X-Frame-Options or Content-Security-Policy: frame-ancestors directive
- **Source:** Passive scan (10020 - Anti-Clickjacking Header)
- **Issue Description:** The response does not protect against **Clickjacking attacks**, which can allow malicious sites to **embed the page in an iframe** and trick users into interacting with concealed UI elements.

Impact Assessment

Potential Risk

- Unauthorized interaction leading to **account compromise**
- Manipulation of user inputs for **phishing and social engineering attacks**
- Increased risk of **fraudulent transactions** through embedded malicious interfaces

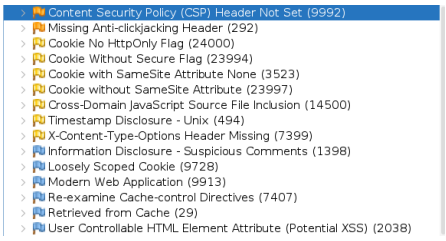
Recommended Remediation

Short-Term Fixes:

- Set **X-Frame-Options** HTTP response header to DENY or SAMEORIGIN
- Implement **Content-Security-Policy (CSP) with frame-ancestors** to restrict embedding

Long-Term Fixes:

- Conduct **regular security audits** to verify correct header configurations
- Strengthen CSP directives for **cross-origin iframe embedding protection**
- Ensure clickjacking defenses align with **OWASP security best practices**



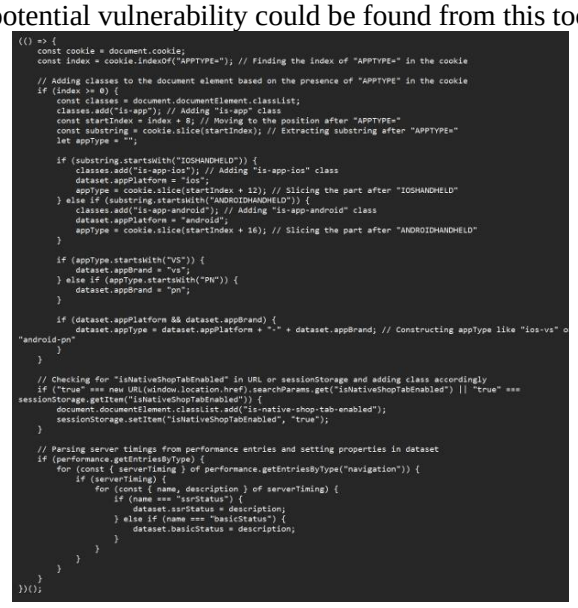
3. Injection

3.1. XSSStrike

Need to install requirements before uing the xssstrike.py.

Python3 xssstrike.py -u https://www.vicctoriassecret.com/us/ --crawl

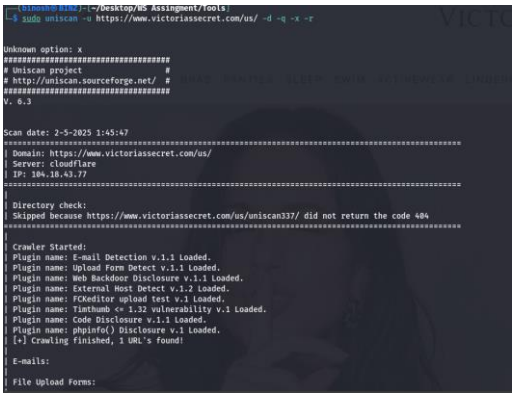
A one potential vulnerability could be found from this tool. The JavaScript code had resulted.



```
(() => {
    const cookie = document.cookie;
    const index = cookie.indexOf("APPTYPE="); // Finding the index of "APPTYPE=" in the cookie

    // Adding classes to the document element based on the presence of "APPTYPE" in the cookie
    if (index >= 0) {
        const classes = document.documentElement.classList;
        classes.add("is-app"); // Adding "is-app" class
        const startIndex = index + 8; // Moving to the position after "APPTYPE="
        const substring = cookie.slice(startIndex); // Extracting substring after "APPTYPE="
        let appType = "";

        if (substring.startsWith("IOSHANDHELD")) {
            classes.add("is-app-ios"); // Adding "is-app-ios" class
            dataset.appPlatform = "ios";
            appType = cookie.slice(startIndex + 12); // Slicing the part after "IOSHANDHELD"
        } else if (substring.startsWith("ANDROIDHANDHELD")) {
            classes.add("is-app-android"); // Adding "is-app-android" class
            dataset.appPlatform = "android";
            appType = cookie.slice(startIndex + 16); // Slicing the part after "ANDROIDHANDHELD"
        }

        if (appType.startsWith("VS")) {
            dataset.appBrand = "vs";
        } else if (appType.startsWith("PN")) {
            dataset.appBrand = "pn";
        }

        if (dataset.appPlatform && dataset.appBrand) {
            dataset.appType = dataset.appPlatform + "-" + dataset.appBrand; // Constructing appType like "ios-vs" o
"android-pn"
        }
    }

    // Checking for "isNativeShopTabEnabled" in URL or sessionStorage and adding class accordingly
    if ("true" === new URL(window.location.href).searchParams.get("isNativeShopTabEnabled") || "true" ===
sessionStorage.getItem("isNativeShopTabEnabled")) {
        document.documentElement.classList.add("is-native-shop-tab-enabled");
        sessionStorage.setItem("isNativeShopTabEnabled", "true");
    }

    // Parsing server timings from performance entries and setting properties in dataset
    if (performance.getEntriesByType) {
        for (const { serverTiming } of performance.getEntriesByType("navigation")) {
            if (serverTiming) {
                for (const { name, description } of serverTiming) {
                    if (name === "ssrStatus") {
                        dataset.ssrStatus = description;
                    } else if (name === "basicStatus") {
                        dataset.basicStatus = description;
                    }
                }
            }
        }
    }
})();
```

This JavaScript code is designed to modify the behavior and appearance of a website based on certain conditions, such as the presence of specific cookies or URL parameters, and performance metrics.

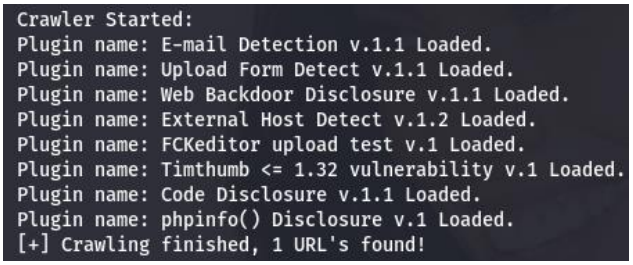
3.2 Uniscan



uniscan -u https://www.victoriassecret.com/us/ -qd

qd → To enable directory and dynamic checks.

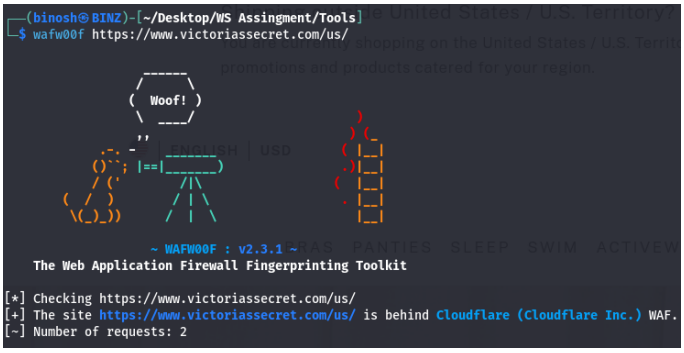
The uniscan results are as below.



The above result is about detected plugins or features that may be vulnerable to exploitation. It is noticeable that older versions of some services are being used in this domain. And, the below potential issues were found suggesting that the website may expose email addresses insecurely, potentially leading to spam or phishing attacks. It also indicates the presence of an upload form that could be exploited to upload malicious files, as well as making requests to external hosts, which could pose a security risk if not properly managed.

4. Firewall Detection

4.1. Wafw00f



Could be detected that Cloudflare is being used as the firewall in this site. Ther is a known vulnerability for Cloudflare Firewall. Need to be alert about this and have to update the firewall regularaly.

How to mitigate the Vulnerability

- Apply security patches by updating to the most recent version of Angular.
- Verify and clean user input to avoid doing too much backtracking.
- To stop unwanted script execution, use stringent Content Security Policy (CSP) headers.
- Switch to an updated, security-patched version of Angular.
- Update to jQuery UI 1.13.2 or later, as this problem has been fixed.
- Before transferring user input to UI elements, sanitize it.
- Update to at least jQuery 3.5.0.
- For security patches, update to Moment.js 2.29.2 or later.
- Before sending locale inputs to Moment.js routines, clean them up.
- Limit modifications to dynamic locales to trusted values only.
- Before sending user input to DOM manipulation routines, clean it up.
- Use CSP headers to prevent malicious scripts from running.

Proof of report submission.