# 10 Bug Bounty reports

Table of Contents

# 1. Acknowledgment

An ever-more-critical demand exists for ethical, skilled, and watchful hackers as the cybersecurity landscape changes. Aware of this necessity, well-known companies all over the world have set up strong bug bounty programs that rely on the knowledge of white hat hackers to find weaknesses in their digital systems. These initiatives are crucial for strengthening an organization's security defenses and making sure they can withstand any threats.

I got interested in bug bounty hunting as a curious student who wanted to learn about the complexities of web applications. Motivated by an intense interest in cybersecurity and a strong desire to enhance safety in the digital realm, I thoroughly investigated the designs, protocols, and security measures of web-based software. This fundamental knowledge turned out to be quite helpful, giving me the ability I needed to appropriately find and report important flaws.

# 2.Objective

Finding possible security flaws in the infrastructure of a well-known website is the main goal of this security evaluation. Such vulnerabilities carry a high risk of causing breaches that jeopardize the availability, confidentiality, and integrity of the data and services on the website. For college cybersecurity students in particular, this research is relevant since it offers a real-world implementation of the theoretical ideas covered in the classroom.

# 3.Introduction

The widespread use of information technology in organizational operations in the current digital era is both a strategic advantage and a possible liability. Regrettably, the exponential rise in data generation has coincided with a rise in cyber dangers due to the swift development and uptake of digital technology. These risks include everything from sophisticated attacks that target financial, intellectual, and operational assets to data breaches and unauthorized access. In this context, IT security auditing becomes an essential instrument for protecting digital resources as well as a recommended practice. The significance of IT security auditing in stopping cybercrime and improving data security in businesses is emphasized in this research.

Cybercrime costs the world economy significant damages every year as it becomes more sophisticated and devastating. According to the most recent statistics, cyberattacks are occurring more frequently against enterprises globally, and the average cost of a data breach is rising to previously unheard-of heights. These events undermine confidence and jeopardize the reputation of the company in addition to causing financial harm. As a result, IT security audits play a crucial role as a preventive measure by methodically assessing how resilient an organization's IT environment is to possible threats.

A thorough evaluation of an organization's information systems, policies, and procedures is part of an IT security audit. The objectives of this assessment are to find weaknesses, make sure that national and international standards are followed, and confirm that IT controls protect the availability, integrity, and confidentiality of data. Organizations can adopt strategic improvements by detecting anomalies and holes in their security frameworks through thorough audits. In addition to reducing risks, this proactive approach maximizes the effectiveness and dependability of IT operations.

Furthermore, the value of IT security audits goes beyond only reducing risks. Audits guarantee compliance with legal and ethical standards, preventing serious fines and legal ramifications in an era where regulatory compliance (including GDPR, HIPAA, and others) progressively defines the operational landscapes of sectors. Additionally, they promote a security-conscious culture within the company by informing staff members about the most recent security dangers and best practices.

This paper will examine the approaches used in IT security audits, the difficulties auditors face, and the best strategies for resolving these issues. It will also look at case studies that show how important IT security audits are in preventing cyberattacks and bolstering data security measures. Organizations may safeguard their operational integrity and maintain their competitive edge in the digital marketplace by strengthening their defenses against the growing threat of cybercrime and developing a comprehensive understanding of IT security auditing.

# 4.Bug Bounty Reports

## 1.Session Cookie Not Marked as Secure

### What is Session Cookie Not Marked as Secure?

A website can configure a session cookie with different properties that specify how the browser should treat it. The Secure flag is one of these characteristics. The browser will only deliver a session cookie over an encrypted HTTPS connection rather than an unencrypted HTTP connection when the Secure attribute is set on the cookie.

When a session cookie's Secure flag is left unset, a vulnerability results. The cookie can be delivered over HTTP and HTTPS connections because to this absence. A user's browser may communicate session cookies in plain text when sending them over an HTTP connection. This includes potentially sensitive data like session identifiers.

Man-in-the-Middle (MitM) Attacks: Attackers may put themselves in the way of the user and the application server without using HTTPS, intercepting and altering the data in transit and maybe taking session cookies.

**Affected components for this vulnerability:**

1.User Browser:

The endpoint responsible for managing and storing the session cookie is the browser. Respecting the Secure property and enforcing the rule that these cookies should only be sent over secure connections are requirements for browsers. Even though most modern browsers handle this effectively, part of general security hygiene is making sure that every user has a browser that complies with these requirements.

2. Network Infrastructure:

Network architecture contributes to ensuring that HTTPS is utilized for all connections where cookies may be communicated, even though it is not directly related to cookie settings. This entails configuring SSL/TLS correctly and, if necessary, utilizing VPN (Virtual Private Networks) technology to provide further data transfer security, particularly in settings handling sensitive data.

**Impacted assessment:**

1. Risk of Sniffing and Eavesdropping Attacks :

Session cookies may be sent via unsecured HTTP connections if they are not tagged as secure. This means that anyone able to keep an eye on network traffic can intercept them. This is especially likely to occur in situations where network security has been breached or on public Wi-Fi networks. Without requiring the user's password, attackers can access the web application to the same degree as authorized users if they manage to intercept these cookies.

2.Data Breach:

An attacker may gain access to private or sensitive data once they have seized control of a user's session. This information may include details about the user and possibly other users as well. This could result in serious data breaches that expose private or sensitive information, including financial and personal data as well as confidential company information, depending on the attacker's access level and the application's design.

3.Lost user trust and reputation:

Users may lose trust if they find out that a security lapse, like using insecure cookies, allowed their information to be compromised. This can result in a tarnished reputation for companies, which can have a lasting impact on client loyalty and overall economic success. It frequently takes a lot of work and money to recover from such harm in the areas of customer service, public relations, and enhanced security measures.

4.Financial Impact:

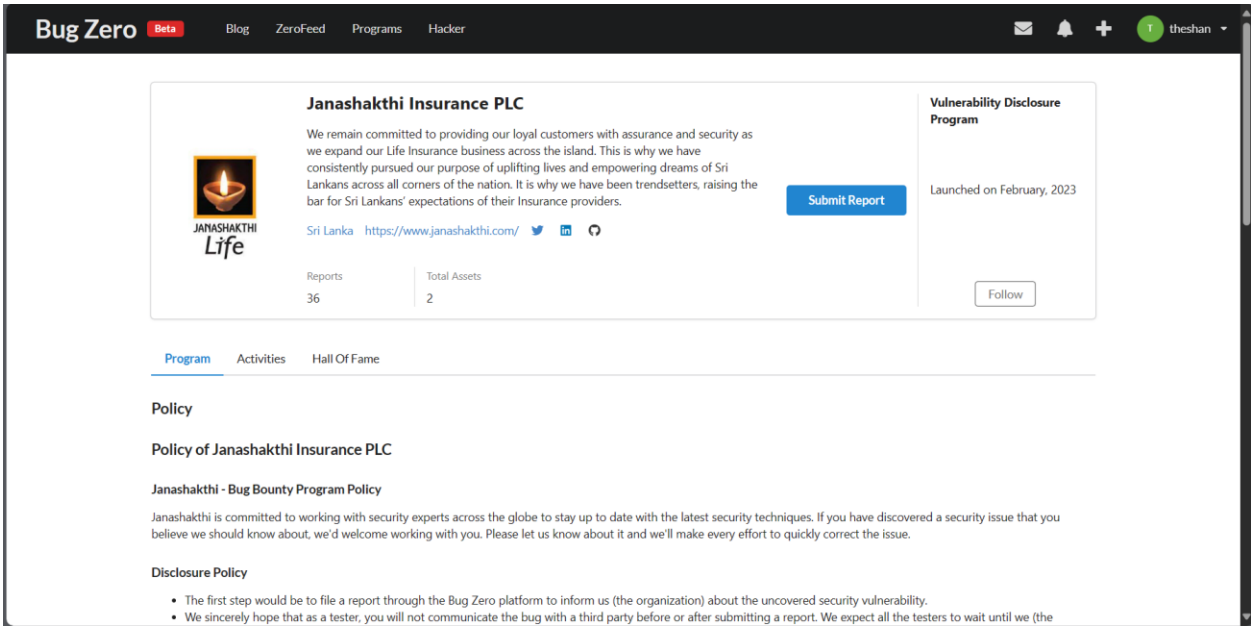
The financial ramifications of a security breach resulting from a session cookie that was not tagged as safe can extend beyond fines and legal fees. These may include expenses related to forensic investigations, bolstering security postures, informing impacted users, and perhaps compensating affected parties. Indirect expenses from lost revenue and downtime could also exist.

**Steps to reproduce:**

You must watch how cookies are handled during data transmission between a client (often a web browser) and a server over a network in order to replicate the vulnerability when a session cookie is not recognized as secure. Verifying that the session cookie can be transferred via an insecure connection (HTTP) and potentially be intercepted is the aim of this test.

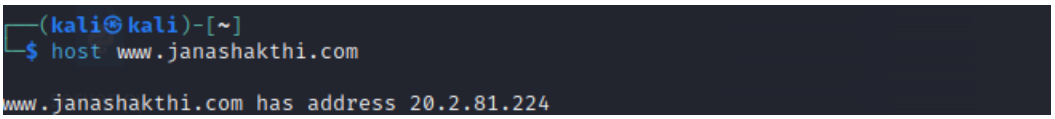
**Proof of concept (if applicable):**

Use "**netcraft**" to find information's,

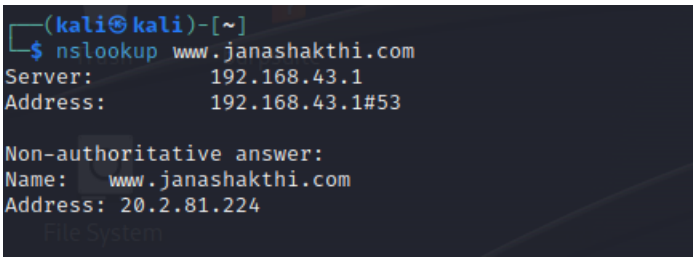


**host** and **nslookup** tools are programs that collect data from the Domain Name System (DNS) using command lines. The DNS assists in converting human-readable domain names into IP addresses that computers use to identify one another on a network.

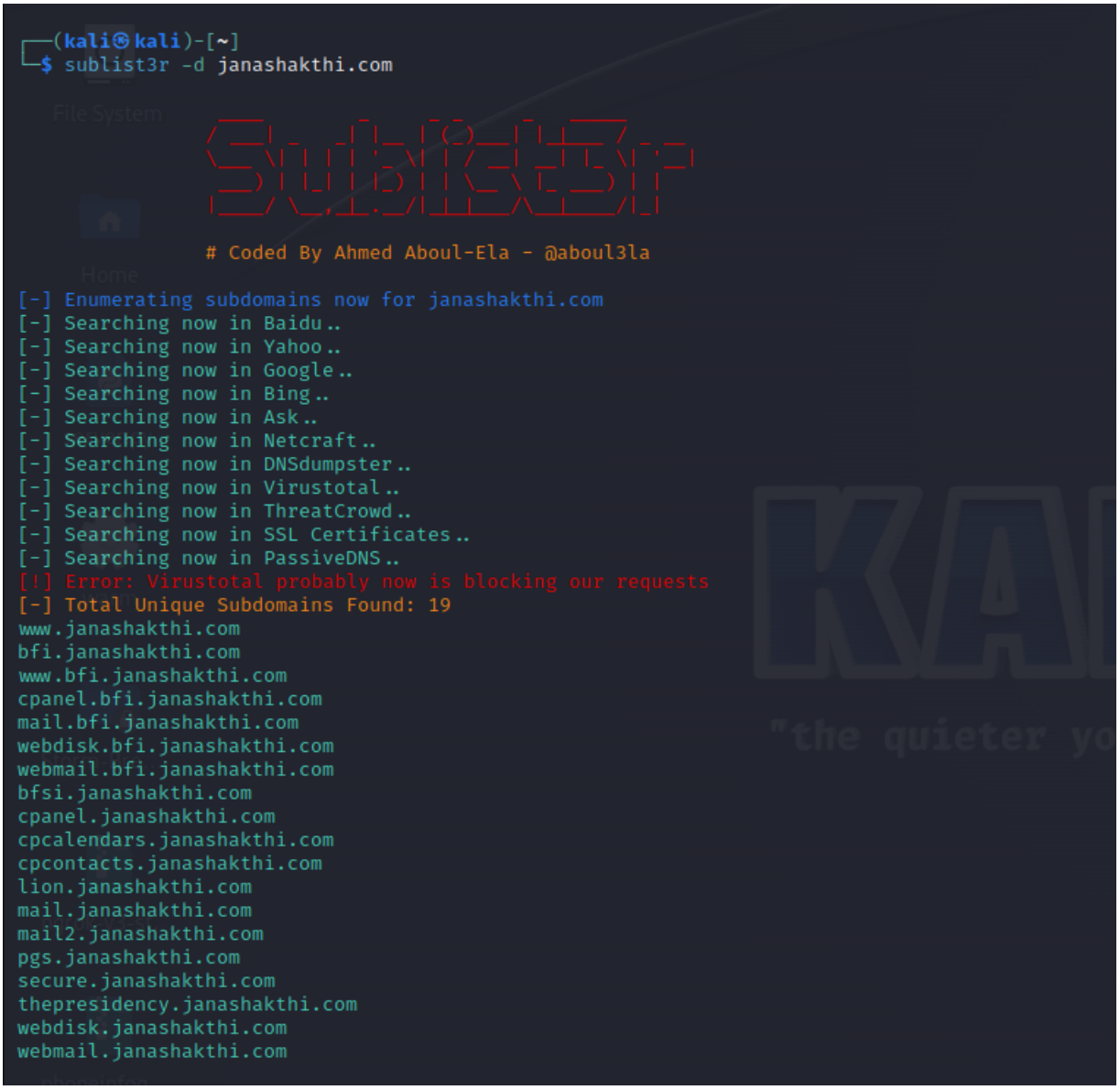
## Host




## Nslookup



By using “**sublist3r**” like tool we can find subdomains of the main,



## Nmap

```
┌──(kali㉿kali)-[~]
└─$ nmap 20.2.81.224
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-18 11:53 EDT
Nmap scan report for 20.2.81.224
Host is up (0.14s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE
22/tcp  open  ssh
53/tcp  open  domain
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 36.28 seconds
```

## OWASP ZAP Report

### Alert counts by site and risk

This table shows, for each site for which one or more alerts were raised, the number of alerts raised at each risk level.

Alerts with a confidence level of "False Positive" have been excluded from these counts.

(The numbers in brackets are the number of alerts raised for the site at or above that risk level.)

| | | Risk | | | |
|---|---|---|---|---|---|
| | | High (= High) | Medium (>= Medium) | Low (>= Low) | Informational (>= Informational) |
| Site | https://www.janashakthi.com | 0 (0) | 3 (3) | 3 (6) | 7 (13) |

## Cookie Without Secure Flag (1)

▼ GET https://www.janashakthi.com/about

| | |
|---|---|
| **Alert tags** | ▪ OWASP_2021_A05<br>▪ WSTG-v42-SESS-02<br>▪ OWASP_2017_A06 |
| **Alert description** | A cookie has been set without the secure flag, which means that the cookie can be accessed via unencrypted connections. |
| **Request** | ▼ Request line and header section (295 bytes)<br><br>GET https://www.janashakthi.com/about HTTP/1.1<br>host: www.janashakthi.com<br>user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36<br>pragma: no-cache<br>cache-control: no-cache<br>referer: https://www.janashakthi.com/sitemap.xml<br><br>▼ Request body (0 bytes) |

| | |
|---|---|
| **Response** | ▾ Status line and header section (375 bytes)<br><br>HTTP/1.1 200<br>Server: nginx<br>Date: Tue, 23 Apr 2024 03:37:35 GMT<br>Content-Type: text/html;charset=ISO-8859-1<br>Connection: keep-alive<br>Set-Cookie: JSESSIONID=09A9ACC342A81CD9FA7F4E4829CE0870;<br>Path=/; HttpOnly<br>Strict-Transport-Security: max-age=31536000<br>X-Content-Type-Options: nosniff<br>X-Frame-Options: SAMEORIGIN<br>X-XSS-Protection: 1; mode=block<br>content-length: 111928<br><br>▸ Response body (111928 bytes) |
| **Parameter** | JSESSIONID |
| **Evidence** | Set-Cookie: JSESSIONID |

### Cookie Without Secure Flag

| | |
|---|---|
| **Source** | raised by a passive scanner (Cookie Without Secure Flag) |
| **CWE ID** | 614 |
| **WASC ID** | 13 |
| **Reference** | ▪ https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html |

**Proposed mitigation or fix**

1. Set the Secure Attribute: Make that the Secure property is set for session cookies in web apps. Usually, this can be completed at the cookie's configuration or setting point.

2. Enforce HTTPS: To avoid sending any data over unencrypted networks, make sure your website or application is only provided over HTTPS. This involves configuring HTTP Strict Transport Security (HSTS), which instructs browsers to communicate with your server only over HTTPS.

3. Scan and Audit: Use both automated tools and manual review to regularly check your application for configuration errors in cookie settings and other security headers.

4. Use Frameworks that Encourage Secure Defaults: Use web frameworks that enforce secure defaults whenever possible to lower the chance of developer errors.

# 2.Out of date version (Lodash)

**What is Out of date version Vulnerability:**

It appears that the "Out-of-date Version (Lodash) vulnerability" you mentioned is a flaw in an earlier version of the Lodash library. With the aid of functions, you can work with arrays, objects, strings, and more using Lodash, a well-known JavaScript toolkit. An application's security may be jeopardized if an outdated version of Lodash is utilized in it or on its website. Tools that are outdated frequently include known security flaws that hackers might use to gain unauthorized access, execute malicious code, and carry out other malicious actions. Depending on where and how the outdated version is utilized in your software stack, the Out-of-date Version (Lodash) vulnerability may impact different areas of your program.

**Affected components for this vulnerability:**

1. Operating Systems: The fundamental program that controls computer hardware and software resources is called an operating system (OS), and examples include Windows, macOS, Linux, and others. Operating systems that are outdated may include security flaws that can be exploited and are patched in more recent versions. Updating an operating system (OS) is essential for preserving functionality and security because updates frequently bring fixes for serious security flaws as well as improvements and new features.

2. Software Applications: Computer programs, be they web browsers, productivity tools, or specialized applications, are constantly updated to address security flaws, add new features, and correct errors. If a program is not updated, it might be exposed to many types of attacks, such as data leakage and remote code execution, among others.

3. Web Browsers: Because web browsers are so widely used and have access to so much important data, they are often targets for attackers. Browser updates frequently address compatibility and security flaws. Users that use outdated browsers are vulnerable to various dangers such as malicious scripts, drive-by downloads, and phishing attempts.

**Impacted assessment:**

Increased Risk of Exploits: Vulnerabilities in outdated software are frequently known and can be used by attackers. Because exploits for these vulnerabilities are frequently easily accessible and well-documented, it is simpler for attackers to compromise systems.

Easier Target for Attackers: Because obsolete systems have known vulnerabilities, attackers frequently search networks for them. This makes them easier targets.

Compromised Data Integrity and Confidentiality: In the event that these vulnerabilities are successfully exploited, critical data integrity and confidentiality may be jeopardized by unauthorized access and data breaches.

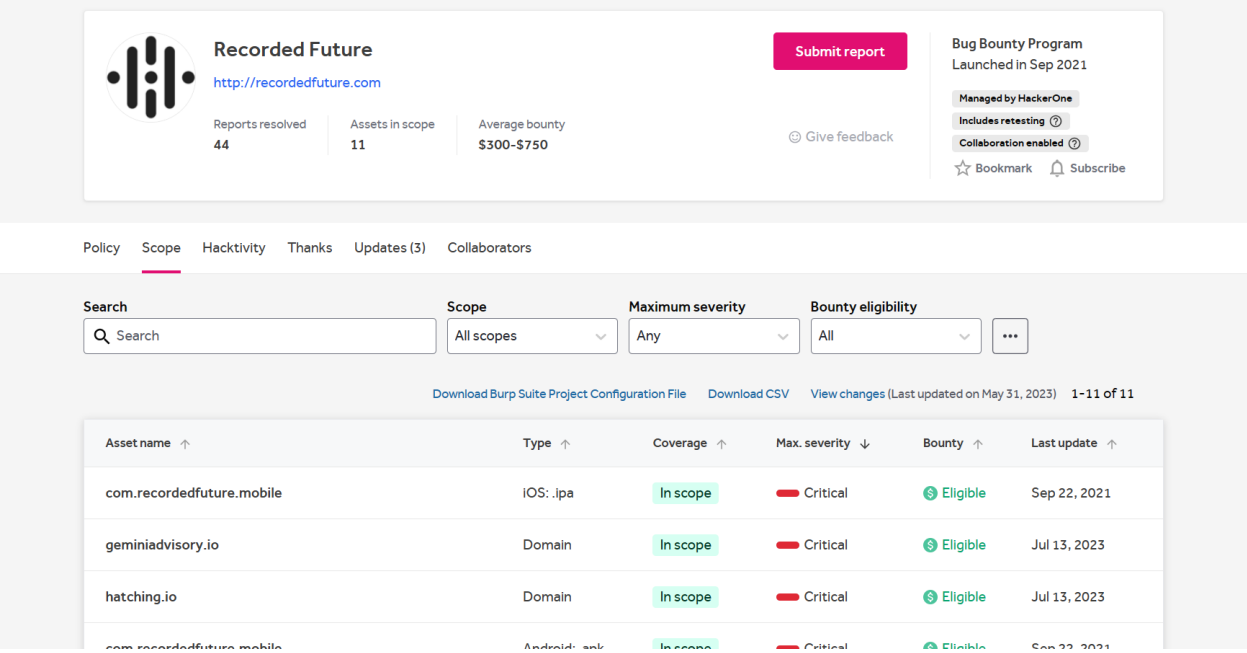
**Steps to reproduce:**

Throughout its operations, an organization uses a variety of software solutions, such as operating systems, third-party apps, and specially created software. Software providers update their products over time to fix newly found security flaws, enhance functionality, or keep them compatible with other developments in technology.

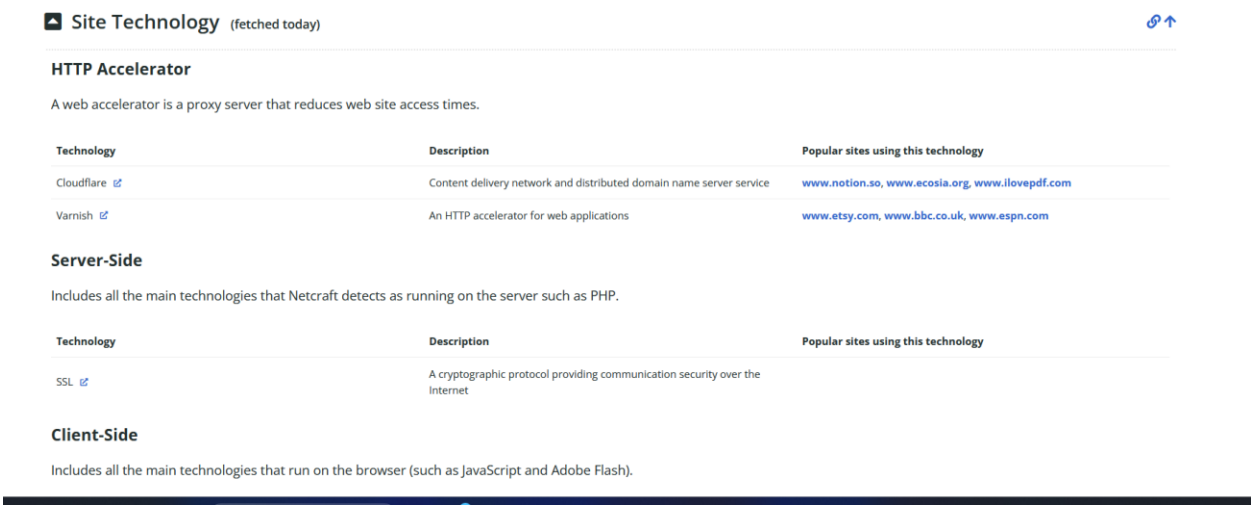
Certain software components may continue to be at previous versions if the organization does not update its software in a systematic manner. This could be the

result of a lack of IT personnel, worries about possible disruptions from updates, or simple oversight. These out-of-date versions might have security flaws that have been fixed in more recent versions. Attackers can take advantage of these flaws when they are aware of them, which are frequently listed in open vulnerability databases or made public in security bulletins. Because attackers can leverage pre-existing tools, scripts, or techniques created especially to target these known vulnerabilities, this exploitation is rather simple.

## Proof of concept (if applicable):



Finding information by using “**netcraft**”



Site Technology (fetched today)

**HTTP Accelerator**

A web accelerator is a proxy server that reduces web site access times.

| Technology | Description | Popular sites using this technology |
|---|---|---|
| Cloudflare | Content delivery network and distributed domain name server service | www.notion.so, www.ecosia.org, www.ilovepdf.com |
| Varnish | An HTTP accelerator for web applications | www.etsy.com, www.bbc.co.uk, www.espn.com |

**Server-Side**

Includes all the main technologies that Netcraft detects as running on the server such as PHP.

| Technology | Description | Popular sites using this technology |
|---|---|---|
| SSL | A cryptographic protocol providing communication security over the Internet | |

**Client-Side**

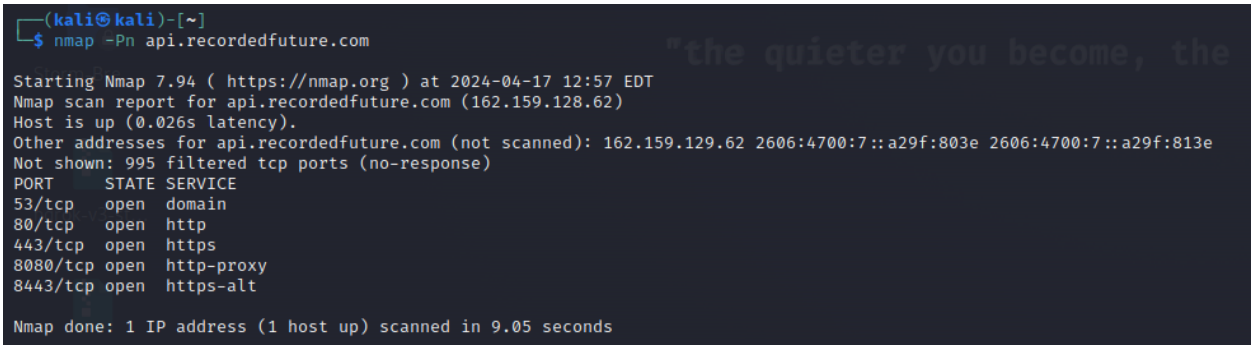
Includes all the main technologies that run on the browser (such as JavaScript and Adobe Flash).

By using “**whatweb”** can find which type of technology used in the websit.

```
┌──(kali㉿kali)-[~]
└─$ whatweb http://recordedfuture.com
http://recordedfuture.com [301 Moved Permanently] Cookies[_cfuvid], Country[UNITED STATES][US], HTTPServer[cloudflare], HttpOnly[_cfuvid], IP[104.18.43.111], RedirectLocation[https://recordedfuture.com/], Title[301 Moved Permanently],
UncommonHeaders[x-content-type-options,cf-ray]
https://recordedfuture.com/ [301 Moved Permanently] Cookies[_cfuvid], Country[UNITED STATES][US], HTTPServer[cloudflare], HttpOnly[_cfuvid], IP[104.18.43.111], RedirectLocation[https://www.recordedfuture.com/], Strict-Transport-Securit
y[max-age=31536000; includeSubDomains], Title[301 Moved Permanently], UncommonHeaders[traceresponse,x-debug-info,x-platform-server,x-served-by,x-cache-hits,cf-cache-status,x-content-type-options,cf-ray], Via-Proxy[1.1 varnish]
https://www.recordedfuture.com/ [200 OK] Cookies[_cfuvid], Country[RESERVED][ZZ], Frame, HTML5, HTTPServer[cloudflare], HttpOnly[_cfuvid], IP[172.64.144.145], Open-Graph-Protocol[website], Script[application/json], Strict-Transport-Sec
urity[max-age=31536000; includeSubDomains], Title[Recorded Future: Securing Our World With Intelligence], UncommonHeaders[traceresponse,x-debug-info,x-nextjs-cache,x-platform-server,x-served-by,x-cache-hits,cf-cache-status,x-content-ty
pe-options,cf-ray], Via-Proxy[1.1 varnish], X-Powered-By[Next.js]
```

Checking subdomains in this web site using "**sublist3r**"

Checking vulnerabilities using **nmap**



# Netspacker report

## 1. Out-of-date Version (Lodash)

CRITICAL 1

Netsparker identified that the target web site is using Lodash and detected that it is out of date.

### Impact

Since this is an old version of the software, it may be vulnerable to attacks.

**Lodash Improperly Controlled Modification of Object Prototype Attributes ('Prototype Pollution') Vulnerability**

Prototype pollution attack when using _.zipObjectDeep in lodash before 4.17.20.

### Affected Versions

0.1.0 to 4.17.19

### External References

- CVE-2020-8203

**Lodash Improper Neutralization of Special Elements used in a Command ('Command Injection') Vulnerability**

Lodash versions prior to 4.17.21 are vulnerable to Command Injection via the template function.

### Affected Versions

0.1.0 to 4.17.20

### External References

- CVE-2021-23337

**Lodash Other Vulnerability**

Lodash versions prior to 4.17.21 are vulnerable to Regular Expression Denial of Service (ReDoS) via the toNumber, trim and trimEnd functions.

### Affected Versions

0.1.0 to 4.17.20

### External References

- CVE-2020-28500

**Lodash Other Vulnerability**

Versions of lodash lower than 4.17.12 are vulnerable to Prototype Pollution. The function defaultsDeep could be tricked into adding or modifying properties of Object.prototype using a constructor payload.

### Affected Versions

0.1.0 to 4.17.11

**Proposed mitigation or fix**

Using an outdated version of a library, like Lodash, poses a security risk that can be reduced by taking a number of calculated precautions that are aimed at keeping software dependencies up to date. First, in order to find outdated libraries, it is crucial to routinely audit the dependencies in your project. For JavaScript projects, tools such as npm's npm audit or yarn audit can be very helpful because they identify and report vulnerabilities in installed npm packages. Once an out-of-date Lodash version has been found, review the release notes or change logs of the most recent iterations to learn about the enhancements and potentially disruptive changes they bring.

The Lodash library should then be updated to the most recent stable version that doesn't clash with any other components of your application. Usually, you can accomplish this by using your package manager using commands like yarn upgrade lodash or npm update lodash. Make sure the update hasn't introduced any new problems or interfered with any functionality by carefully testing your program after it has been updated. Before sending these upgrades to production, it is imperative that they are completed in a development or staging environment.

# 3.Vulnerable JS Library(jQuery)

## What is vulnerable JS library(jquery)?

A JavaScript library that contains known security flaws that an attacker could exploit is referred to as a "vulnerable JS library". When we discuss jQuery as a vulnerable JS library, we mean that vulnerabilities have been found in certain versions of jQuery.

For instance, there may be security flaws in earlier iterations of jQuery that let attackers use cross-site scripting (XSS) attacks. Through XSS attacks, attackers can insert malicious scripts into websites that other users are seeing, potentially resulting in data theft, user session hijacking, or other undesirable results.

**Affected components for this vulnerability:**

The jQuery Validation Library is frequently used in web applications to facilitate client-side form validation. Using an outdated version of jQuery validation puts your application at risk of several potential vulnerabilities that could impact different areas of it. Those with bad intentions could take advantage of these weaknesses. Some elements that are subject to change are as follows:
Verification of User Data Security vulnerabilities in outdated versions of jQuery validation could allow malicious users to get around client-side validation and send damaging or inaccurate data. Security issues like data manipulation or injection attacks could arise from this.

**Impacted assessment:**

1. Loss of Data Integrity and Confidentiality: Sensitive information can be intercepted or altered through the use of XSS and other vulnerabilities, resulting in data breaches.

2. Compromised User Trust: Users' confidence in your service may be severely damaged if they experience attacks as a result of security holes in your application.

3. Regulatory and Compliance Issues: Violating security requirements, such as by employing out-of-date libraries, may give rise to fines and legal problems, contingent upon your industry and jurisdiction.

**Steps to reproduce.**

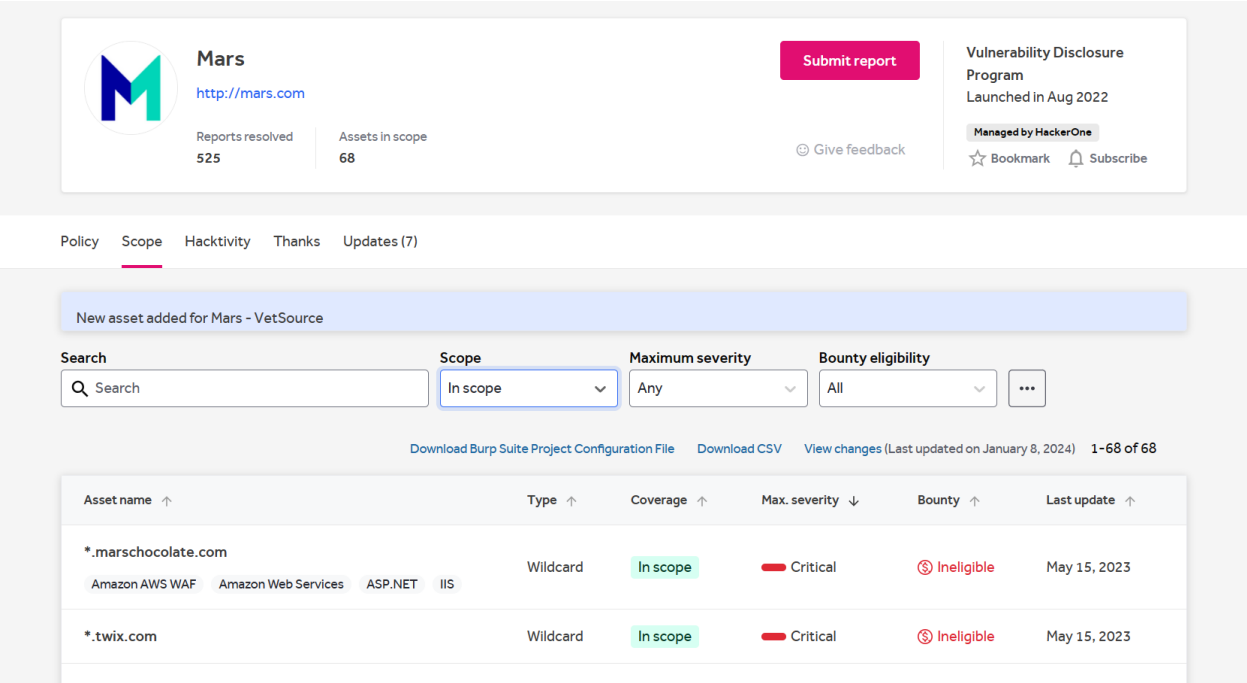
Establish a Development Environment: Install a web server in a local or isolated development environment. Make a simple HTML file with scripts in it to host your form. Put an old version of jQuery into your HTML document. Add an obsolete version of the jQuery Validation Plugin as well.

Enter test data into your form fields in a variety of ways that might be able to take advantage of plugin vulnerabilities.
Send in the Form: Keep an eye out for any faulty handling that permits script execution or HTML rendering while you see how the plugin handles input.

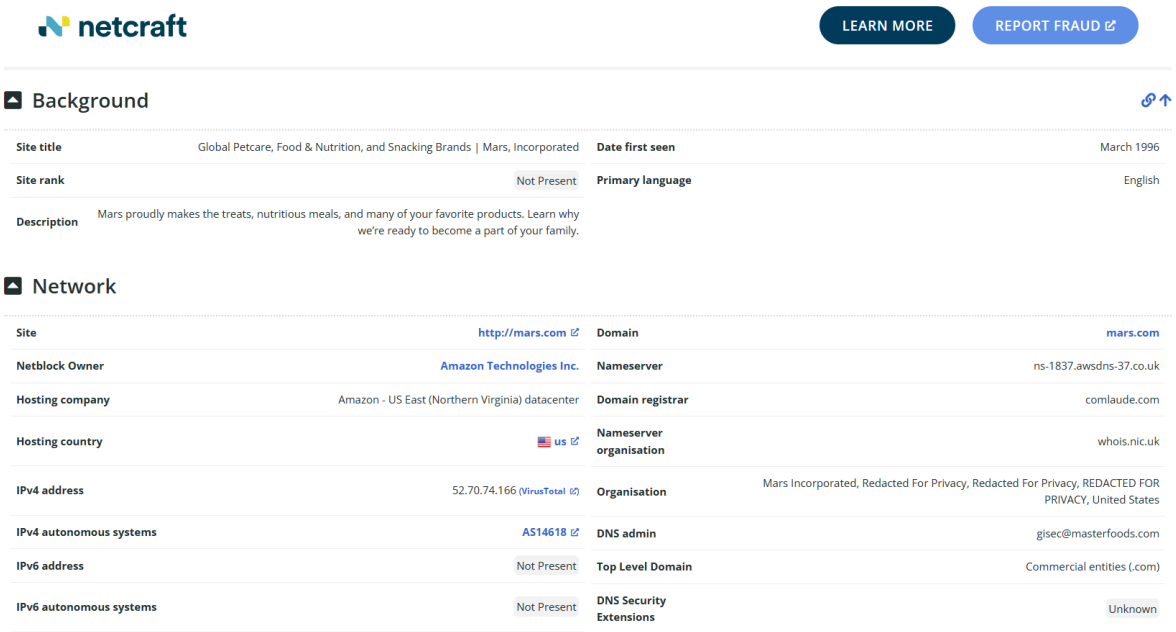
Refresh Libraries: Update to the most recent versions of the jQuery Validation Plugin and jQuery. Put in place extra security measures, such server-side validation or content security restrictions.

Restore Environment: Make sure that all harmful scripts have been removed from your test environment.

# Proof of concept (if applicable)

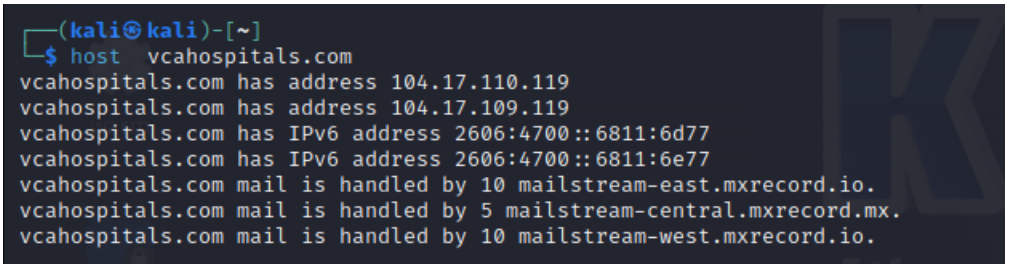


Finding information by using "**netcraft**"

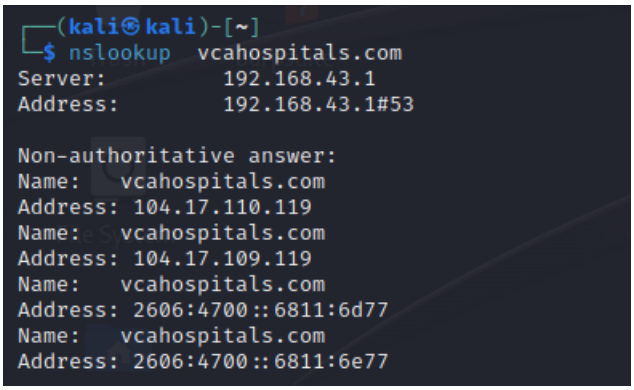



**host** and **nslookup** tools are programs that collect data from the Domain Name System (DNS) using command lines. The DNS assists in converting human-readable domain names into IP addresses that computers use to identify one another on a network.

## Host




## Nslookup



By using “**sublist3r**” like tool we can find subdomains of the main,


By using “**whatweb”** can find which type of technology used in the website.

```
┌──(kali㉿kali)-[~]
└─$ whatweb https://vcahospitals.com/
https://vcahospitals.com/ [200 OK] ASP_NET, Cookies[ASP.NET_SessionId,SC_ANALYTICS_GLOBAL_COOKIE,shell#lang], Country[UNITED STATES][US], Frame, HTML5, HTTPServer[cloudflare], HttpOnly[ASP.NET_SessionId,SC_ANALYTICS_GLOBAL_COOKIE], IP[
104.17.110.119], Open-Graph-Protocol[website], Script[application/json,text/javascript], SiteCore, Strict-Transport-Security[max-age=31536000;includeSubDomains], Title[VCA Animal Hospitals: World-Class Veterinary Care], UncommonHeaders
[cf-ray,cf-cache-status,referrer-policy,x-content-type-options,content-security-policy]
```

## Vulnerability scanning using Nmap tool

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap 104.17.109.119
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-24 03:07 EDT
Nmap scan report for 104.17.109.119
Host is up (0.015s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE
53/tcp   open  domain
80/tcp   open  http
443/tcp  open  https
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 5.56 seconds
```

Vulnerability scanning using **nikto,**

```
└─$ nikto -h https://vcahospitals.com/
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Multiple IPs found: 104.17.110.119, 104.17.109.119, 2606:4700::6811:6d77, 2606:4700::6811:6e77
+ Target IP:          104.17.110.119
+ Target Hostname:    vcahospitals.com
+ Target Port:        443
---------------------------------------------------------------------------
+ SSL Info:        Subject:  /C=US/ST=California/L=Los Angeles/O=VCA, INC./CN=*.vcahospitals.com
                   Ciphers:  TLS_AES_256_GCM_SHA384
                   Issuer:   /C=US/O=DigiCert Inc/CN=DigiCert Global G2 TLS RSA SHA256 2020 CA1
+ Start Time:         2024-04-25 03:36:39 (GMT-4)
---------------------------------------------------------------------------
+ Server: cloudflare
+ /: Cookie shell#lang created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /3cBdn5O8.config: Uncommon header 'x-azure-ref' found, with contents: 20240425T073643Z-r1869b9b46cb28pnfx60nrnch0000000008y0000000016fnf.
+ /3cBdn5O8.tmp: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnera
bilities/missing-content-type-header/
+ /index.php?: Uncommon header 'request-context' found, with contents: appId=cid-v1:d1202b7a-902c-461b-b226-39431ed7365b.
+ /scripts/: Cookie sxa_site created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /robots.txt: Entry '/know-your-pet/*/*/' is returned a non-forbidden or redirect HTTP code (400). See: https://portswigger.net/kb/issues/00600600_robots-txt-file
+ /robots.txt: contains 7 entries which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/Glossary/Robots.txt
+ /index.aspx: Uncommon header 'srv-name' found, with contents: VCA-PRD-CD2.
+ /: The Content-Encoding header is set to "deflate" which may mean that the server is vulnerable to the BREACH attack. See: http://breachattack.com/
+ Server is using a wildcard certificate: *.vcahospitals.com. See: https://en.wikipedia.org/wiki/Wildcard_certificate
+ /sitemap.xml: This gives a nice listing of the site content.
+ /apache/: Cookie LastVisitedHospitalId created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /login.php?pachtofile=http://blog.cirt.net/rfiinc.txt: Uncommon header 'cf-chl-out' found, with contents: 13geHcgVDmgq/LR525/sIZKZ+EAKHztjmqBiEz3Vd73yJrhsDoGM+4B8XXR6F3FLlo+tI7sJ5jGN3EvoObURFMoBZ+EThewDhrXEprxrKafMMGbbxgbEKBQc7Y4fsUR
hOMfdSBM647q/uWYQVHbuSA==$PmPPe8mNuK99IRJNhwhdPw==.
+ /login.php?pachtofile=http://blog.cirt.net/rfiinc.txt: Uncommon header 'origin-agent-cluster' found, with contents: ?1.
+ /login.php?pachtofile=http://blog.cirt.net/rfiinc.txt: Uncommon header 'accept-ch' found, with contents: Sec-CH-UA-Bitness, Sec-CH-UA-Arch, Sec-CH-UA-Full-Version, Sec-CH-UA-Mobile, Sec-CH-UA-Model, Sec-CH-UA-Platform-Version, Sec-CH
-UA-Full-Version-List, Sec-CH-UA-Platform, Sec-CH-UA, UA-Bitness, UA-Arch, UA-Full-Version, UA-Mobile, UA-Model, UA-Platform-Version, UA-Platform, UA.
+ /login.php?pachtofile=http://blog.cirt.net/rfiinc.txt: Uncommon header 'cf-mitigated' found, with contents: challenge.
+ /login.php?pachtofile=http://blog.cirt.net/rfiinc.txt: Uncommon header 'cross-origin-embedder-policy' found, with contents: require-corp.
+ /etc/cloudservices: Uncommon header 'cache' found, with contents: Miss.
+ /etc/cloudservices: Uncommon header 'redirtype' found, with contents: BulkRedirect:facc7388-da9c-49e7-93e7-64a6f91dad99.
+ /?sc_mode=edit: Cookie vca#sc_mode created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /?sc_mode=edit: Cookie website#sc_mode created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /?sc_mode=edit: Cookie coveoanalytics#sc_mode created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /en/setup: Cookie vca_corporate#lang created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /.well-known/assetlinks.json: Google Asset Links Specification file may contain server info. See: RFC-5785 https://github.com/google/digitalassetlinks/blob/master/well-known/details.md
+ /package.json: Node.js package file found. It may contain sensitive information.
+ /.well-known/apple-app-site-association: Apple Universal Links.
+ /.well-known/assetlinks.json: Android App Links.
+ /cdn-cgi/trace: Retrieved access-control-allow-origin header: *.
+ /cdn-cgi/trace: Cloudflare trace CGI found, which may leak some system information.
+ 7974 requests: 0 error(s) and 30 item(s) reported on remote host
+ End Time:         2024-04-25 05:38:26 (GMT-4) (6827 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

# OWASP ZAP Report

## Alert counts by site and risk

This table shows, for each site for which one or more alerts were raised, the number of alerts raised at each risk level.

Alerts with a confidence level of "False Positive" have been excluded from these counts.

(The numbers in brackets are the number of alerts raised for the site at or above that risk level.)

| | | Risk | | | |
|---|---|---|---|---|---|
| | | High (= High) | Medium (>= Medium) | Low (>= Low) | Informational (>= Informational) |
| Site | https://vcahospitals.com | 0 (0) | 8 (8) | 9 (17) | 6 (23) |

**Vulnerable JS Library**

| | |
|---|---|
| **Source** | raised by a passive scanner (Vulnerable JS Library (Powered by Retire.js)) |
| **CWE ID** | 829 |
| **Reference** | ▪ https://github.com/jquery-validation/jquery-validation/blob/master/changelog.md#1194--2022-05-19<br>▪ https://github.com/jquery-validation/jquery-validation/commit/5bbd80d27fc6b607d2f7f106c89522051a9fb0dd<br>▪ https://github.com/advisories/GHSA-ffmh-x56j-9rc3<br>▪ https://github.com/jquery-validation/jquery-validation/blob/master/changelog.md#1200--2023-10-10<br>▪ https://github.com/jquery-validation/jquery-validation/blob/master/changelog.md#1193--2021-01-09 |

## Proposed mitigation or fix

In order to preserve the security and integrity of your online applications, you must take a proactive and organized approach to mitigating risks related to the use of out-of-date or vulnerable JavaScript libraries. Start by routinely checking your libraries for known vulnerabilities in your dependencies using tools like npm audit or yarn audit. These tools can be used to pinpoint individual vulnerabilities and frequently offer recommendations or automated updates to more secure versions.

After vulnerabilities have been found, order updates according to their seriousness and potential effects on your program. By making changes to the dependency files for your project and using your package manager to run update commands, you can update the vulnerable libraries to the most recent versions. Examining change logs and documentation is crucial in order to spot broken changes that can impact the functionality of your program.

Testing is a crucial next step; make sure thorough testing is carried out in a staging

or development environment to confirm that upgrades don't interfere with already-existing functionality. To identify problems early and expedite upgrades, use automated testing and continuous integration/continuous deployment (CI/CD) procedures.

# 4.PII (Personally Identifiable Information) Disclosure

### What is PII vulnerability?

Data that may be used alone or in conjunction with other data to identify, contact, or locate a single person, or to identify an individual in context, is known as Personally Identifiable Information, or PII. When this private data is not sufficiently secured, there is a risk of identity theft, illegal access, and data breaches. This is known as a personally identifiable information vulnerability. These vulnerabilities may arise from a number of sources, such as poor security rules, phishing attacks, weak system access controls, or inadequate data encryption. A PII vulnerability can have serious repercussions for both the organizations in charge of protecting the data and the individuals whose information has been compromised.

Financial losses, legal repercussions, harm to one's reputation, and regulatory fines may result from this, particularly when it comes to regulations like GDPR and HIPAA that place stringent limits on how personal data is handled. Implementing strong security measures, carrying out routine security assessments, educating staff members on data protection best practices, and creating incident response plans to quickly minimize any effects are all necessary to address PII risks.

**Affected components for PII:**

1. Consider how sensitive the personally identifiable information is that has been shared. Consider the many forms of information that may be at stake, such as bank details, SSNs, healthcare records, and passwords. More severe consequences and potential harm to impacted persons may result from the disclosure of highly sensitive PII.

2. Evaluate the impact this event has had on the public's opinion of the business and its dependability. Consider the fallout, the publicity, and the potential decline in sales should news get out. If the company's reputation suffers, it could permanently harm its brand and future business opportunities.

3. Think about the financial effects of instances of PII disclosure. Consider the financial outlay that will be required for a response, an inquiry, a remedy, a lawsuit, potential legal action, and damages for the injured parties. Think about the likely decline in customers, revenue, and job opportunities.

4. Evaluate the effectiveness of the company's response to the breach of personally identifiable information as well as its countermeasures. Assess the speed with which issues are located, addressed, and reported to the appropriate parties. Consider the actions taken to mitigate the damage, such as enhancing privacy policies, bolstering security, and offering credit monitoring services.

**Steps to reproduce:**

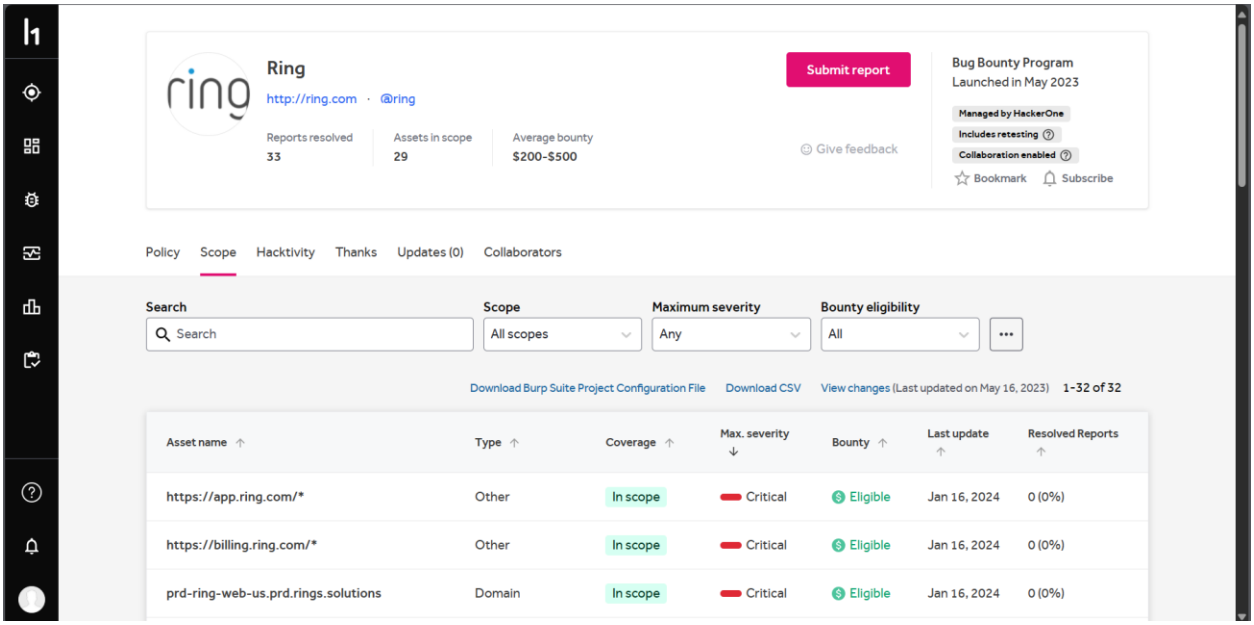
Setup of the Environment: Specify the surroundings where the vulnerability can be replicated. This covers hardware specifications, software versions, network setups, and any other necessary circumstances.

Requirements for Access: Mention any roles, permissions, or user accounts that are needed in order to see the vulnerability.

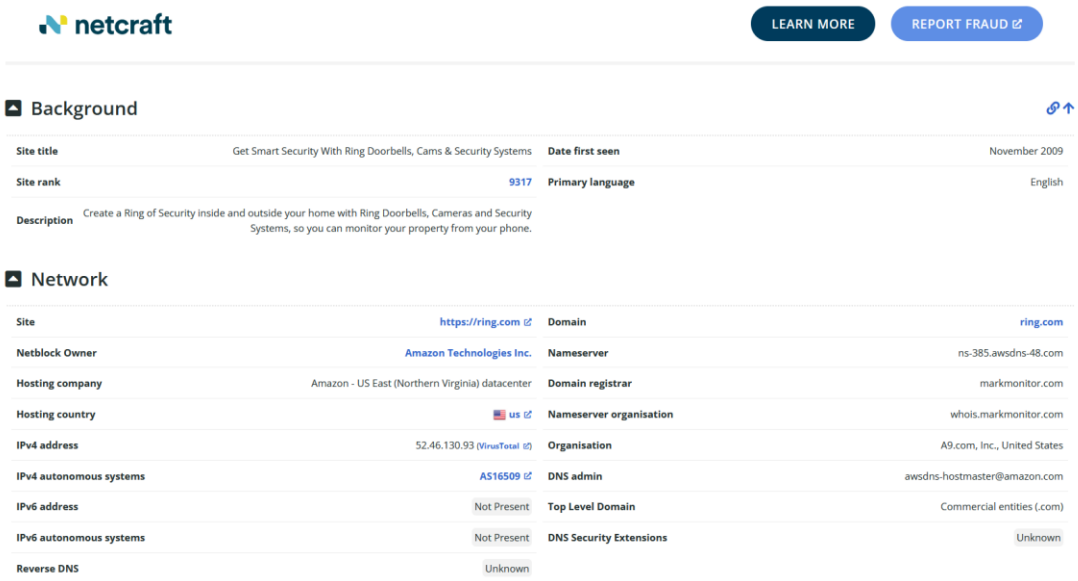
System State: Describe any prerequisite configurations or states that the system needs to be in before the test begins.

Required Tools and Materials: Enumerate all the tools and other materials needed to replicate the vulnerability, including scripts, software for network traffic interceptions, and test data inputs.

## Proof of concepts:

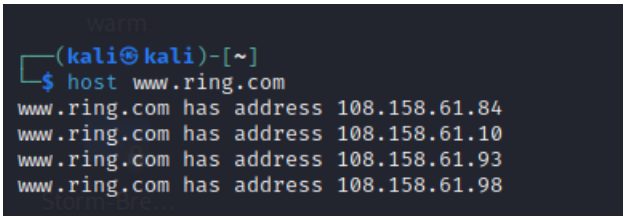


Information gathering,




**host** and **nslookup** tools are programs that collect data from the Domain Name System (DNS) using command lines. The DNS assists in converting human-readable domain names into IP addresses that computers use to identify one another on a network.

## Host

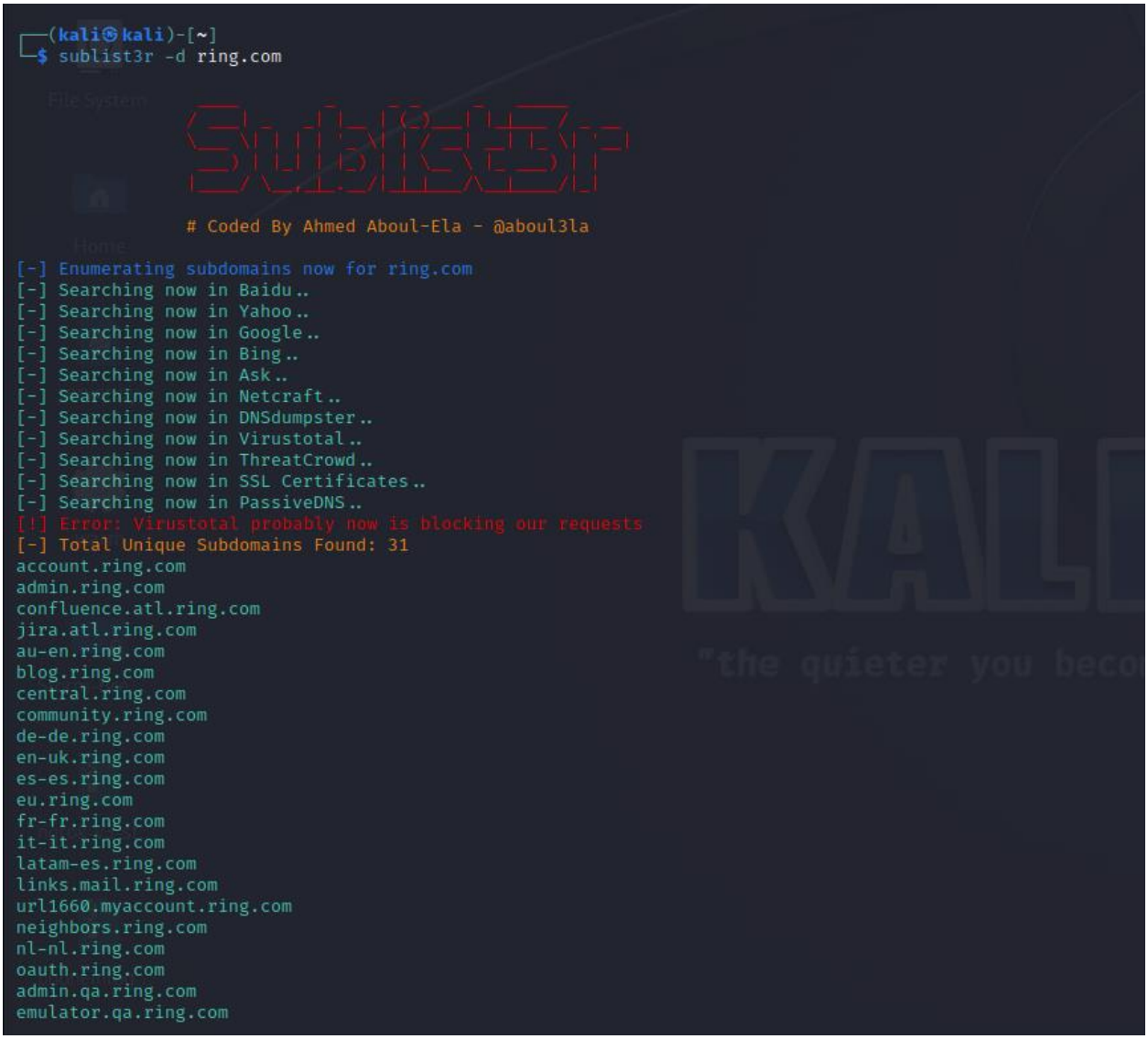


## Nslookup


┌──(kali㉿kali)-[~]
└─$ nslookup www.ring.com
Server:         192.168.43.1
Address:        192.168.43.1#53

Non-authoritative answer:
Name:   www.ring.com
Address: 108.158.61.98
Name:   www.ring.com
Address: 108.158.61.93
Name:   www.ring.com
Address: 108.158.61.10
Name:   www.ring.com
Address: 108.158.61.84

By using "**sublist3r**" like tool we can find subdomains of the main,




## Whatweb

Use **nmap** and check the open ports,

```
┌──(kali㉿kali)-[~]
└─$ nmap -Pn 108.158.61.98

Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-25 12:48 EDT
Nmap scan report for server-108-158-61-98.bom78.r.cloudfront.net (108.158.61.98)
Host is up (0.037s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE
53/tcp  open  domain
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 17.87 seconds
```

## OWASP ZAP Report

### Alert counts by site and risk

This table shows, for each site for which one or more alerts were raised, the number of alerts raised at each risk level.

Alerts with a confidence level of "False Positive" have been excluded from these counts.

(The numbers in brackets are the number of alerts raised for the site at or above that risk level.)

| | | Risk | | | |
|---|---|---|---|---|---|
| | | High (= High) | Medium (>= Medium) | Low (>= Low) | Informational (>= Informational) |
| Site | https://ring.com | 1 (1) | 6 (7) | 7 (14) | 10 (24) |

### PII Disclosure

| | |
|---|---|
| Source | raised by a passive scanner (PII Disclosure) |
| CWE ID | 359 |
| WASC ID | 13 |

## Alerts

**Risk=High, Confidence=High (1)**

https://ring.com (1)

PII Disclosure (1)

▶ GET https://ring.com/

**Proposed mitigation or fix:**

1.Data encryption

At Rest: Use robust encryption techniques, such AES (Advanced Encryption Standard), to protect sensitive personally identifiable information kept on servers, databases, or any other type of storage device.

In Transit: To avoid data interception, use encryption protocols like TLS (Transport Layer Security) for data being transferred over networks.

2.Access control

Least Privilege Principle: Make sure that PII access is solely based on what is required for people to carry out their job duties.

Implement role-based access control (RBAC) to control access to personally identifiable information (PII) according to an individual's job in the company.

Authorization and Authentication: Use multi-factor authentication (MFA) to reinforce authentication procedures. Make sure the permission processes are strong and subject to frequent audits.

3.Data minimization

Limit Collection: Just gather PII that is absolutely required for the goals that your company has in mind.

Data Retention Policies: To guarantee that PII is not held for longer than necessary and is disposed of securely, establish and implement data retention policies.

# 5.Cross Domain Misconfiguration

**What is cross domain misconfiguration?**

When a website or online application incorrectly configures the mechanisms that regulate how resources are shared across various domains, a cross-domain misconfiguration vulnerability occurs. This usually entails mishandling JSONP (JSON with Padding) responses or mistakes in the configuration of Cross-Origin Resource Sharing (CORS) regulations. A security feature called CORS permits or prohibits requests for resources on a webpage to come from domains other than the one that provided the original resource. Strict CORS settings that only permit trusted domains to interact with the resources are ideal. However, any external domain can access the resources if they are configured too loosely, for instance, by setting the Access-Control-Allow-Origin header to "*".

This leaves the application vulnerable to a number of threats, such as session hijacking, data theft, and other exploits linked to cross-site scripting (XSS). To ensure that only trustworthy domains are able to send cross-domain requests to the server, secure configuration necessitates exact definition of which domains are permitted access to the resources. Errors in these policies' setups might thus seriously jeopardize a web application's security by permitting accidental cross-domain interactions.

**Affected Components:**

1.web servers: Incoming requests and outgoing responses are handled by web servers. Misconfigurations pertaining to CORS or other cross-domain policies within web server settings have a direct effect on the way resources are distributed among several domains.

2.Web browsers: In order to limit the ways in which scripts imported from one origin can interact with resources from another, browsers implement the Same-Origin Policy (SOP) and CORS. The browser may unintentionally allow scripts from one domain to access data from another due to a misconfiguration.

3.APIs: It is possible that APIs designed to be accessible through a variety of clients, such as mobile apps and browsers, be set incorrectly to accept queries from

any origin. Sensitive data leakage and illegal access to API endpoints may result from this.

4.Web application frameworks: A lot of web development frameworks have pre-configured options for managing requests from different origins. Developers may fail to adjust these parameters according to the needs of their particular applications, which could result in security flaws.

**Impact Assessment:**

1.Data breach: The loss of sensitive data through unauthorized access may have the worst consequences. Attackers may obtain confidential company information, financial information, or personal information if they are able to take advantage of cross-domain misconfigurations. This could result in privacy violations and serious financial or reputational harm.

2.Session hijacking: Attackers may be able to act on behalf of users by taking advantage of their cookies or authentication tokens if CORS settings are not setup correctly. This may result in the execution of unauthorized operations, which may modify user data or carry out harmful transactions.

3.Account takeover: Attackers may be able to take control of user accounts by taking advantage of cross-domain misconfigurations, which can be especially harmful if the accounts have administrative privileges or access to important areas of the program.

4.Loss of trust: Customers' confidence in a service or product can be damaged by any security issue, especially if it leads to a data breach. This may cause a business to lose clients and have a bad effect on its reputation and place in the market.

**Steps to reproduce:**

One way to test and find a cross-domain misconfiguration vulnerability is to simulate calls to a target website or API from unapproved domains and see how these requests are processed. Testers alter the Origin header in their queries using programs like Postman or a web proxy like Burp Suite to make it appear as though they are coming from a different domain. It is important to verify if the server's answer contains headers such as Access-Control-Allow-Origin and if it is configured to accept all origins (*) or only a subset of them. There is a misconfiguration if it permits everything or reacts to unauthorized origins with sensitive rights. Additionally, to see if the browser imposes the anticipated limitations, testers may use JavaScript to generate a basic HTML page and conduct real cross-origin queries.

The website exhibits a vulnerability if it permits these cross-domain queries without the necessary authorization. This procedure is essential for detecting possible security breaches that can result in other exploits or illegal access to data. By disclosing these results, companies can strengthen their security protocols and safeguard confidential data.

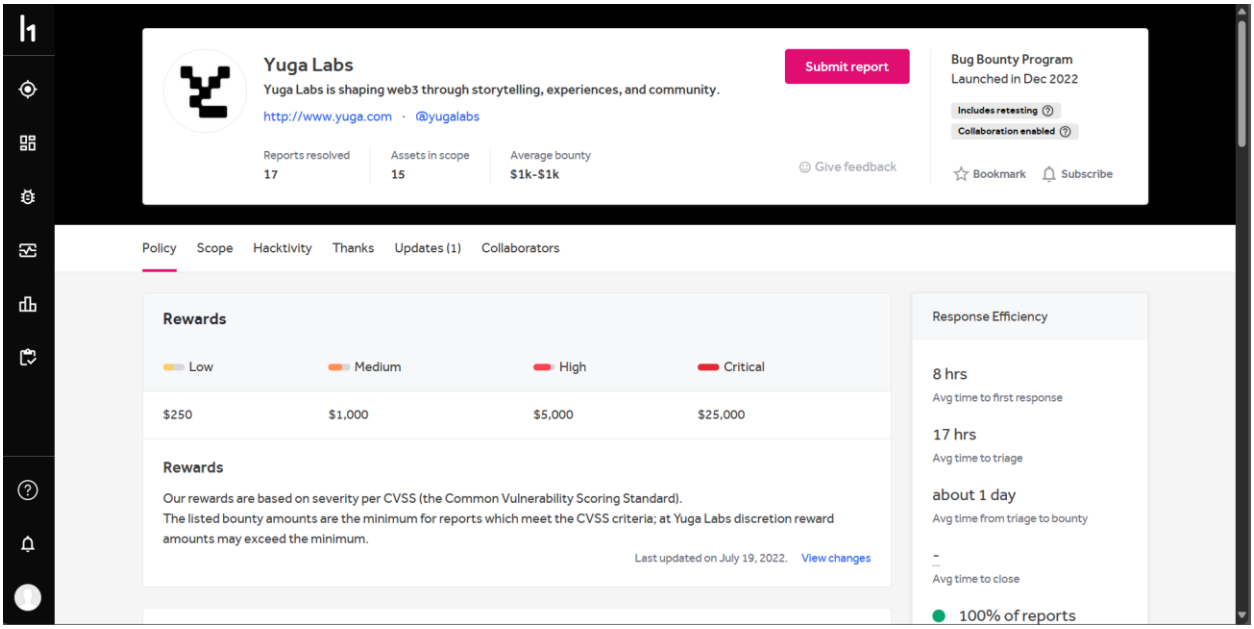
**Proof of concepts:**

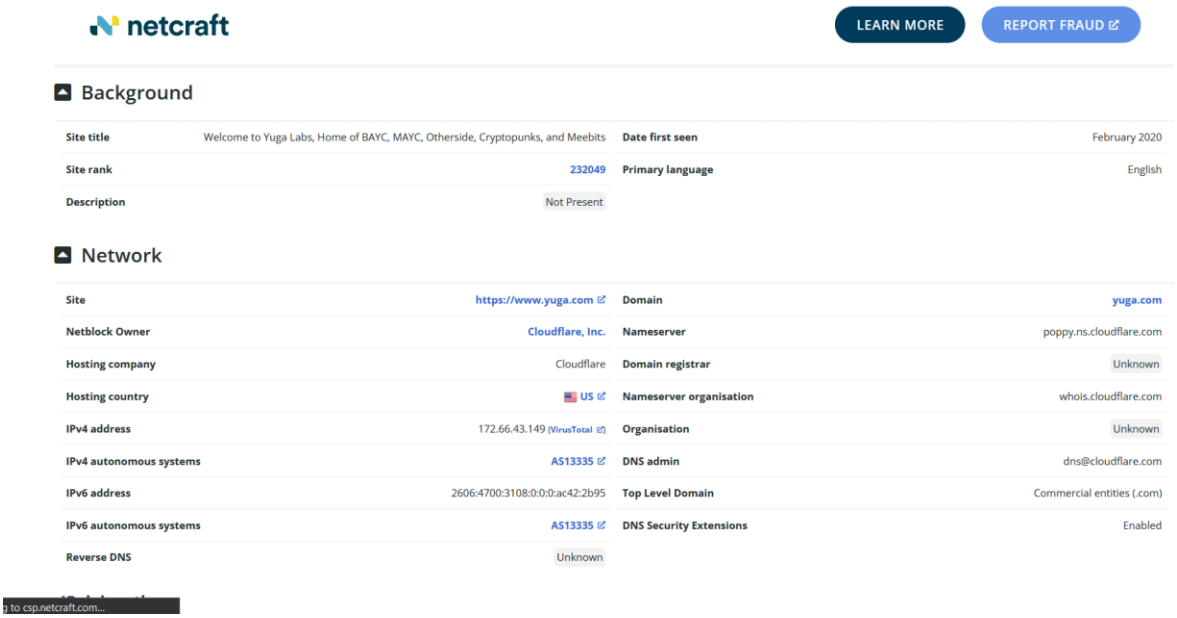
**Information gathering,**

Let's examine the various choices available to us for obtaining not just technical specs but also other important information about https://www.yuga.com/. There are a lot of different ways we might approach finishing this project. There are several possible approaches to doing this task, and we are able to utilize them all. Given that Netcraft is currently our only available resource, let's enter our domain name and see what kind of data we can obtain. We'll start by entering our domain name and then look into it more using the unique information that Netcraft has provided.

Now that we have no other choice except to use Netcraft, let's examine our network to see what information we can gather. We can make the most of our time together if we follow through on this. Since Netcraft is currently our sole available tool, let's see what we can figure out from it. This will enable us to spend our limited resources more wisely. As Netcraft is the only tool available to us at the

moment, let's examine the information it can provide. For the time being, Netcraft is all we have, so let's see what information we can extract from it.



**host** and **nslookup** tools are programs that collect data from the Domain Name System (DNS) using command lines. The DNS assists in converting human-readable domain names into IP addresses that computers use to identify one another on a network.

## Host

```
┌──(kali㉿kali)-[~]
└─$ host www.yuga.com
www.yuga.com has address 172.66.43.149
www.yuga.com has address 172.66.40.107
www.yuga.com has IPv6 address 2606:4700:3108::ac42:286b
www.yuga.com has IPv6 address 2606:4700:3108::ac42:2b95
```

## Nslookup

```
┌──(kali㉿kali)-[~]
└─$ nslookup www.yuga.com
Server:         192.168.43.1
Address:        192.168.43.1#53

Non-authoritative answer:
Name:   www.yuga.com
Address: 172.66.40.107
Name:   www.yuga.com
Address: 172.66.43.149
Name:   www.yuga.com
Address: 2606:4700:3108::ac42:2b95
Name:   www.yuga.com
Address: 2606:4700:3108::ac42:286b
```

## Whatweb


By using "**whatweb"** can find which type of technology used in the websit.

By using “**sublist3r**” like tool we can find subdomains of the main,

**Sublist3r**

# OWASP ZAP Report

## Alert counts by site and risk

This table shows, for each site for which one or more alerts were raised, the number of alerts raised at each risk level.

Alerts with a confidence level of "False Positive" have been excluded from these counts.

(The numbers in brackets are the number of alerts raised for the site at or above that risk level.)

| | | Risk | | | |
|---|---|---|---|---|---|
| | | High (= High) | Medium (>= Medium) | Low (>= Low) | Informational (>= Informational) |
| Site | https://www.yuga.com | 0 (0) | 3 (3) | 4 (7) | 7 (14) |

## Cross-Domain Misconfiguration

| | |
|---|---|
| **Source** | raised by a passive scanner (Cross-Domain Misconfiguration) |
| **CWE ID** | 264 |
| **WASC ID** | 14 |
| **Reference** | ▪ https://vulncat.fortify.com/en/detail?id=desc.config.dotnet.html5_overly_permissive_cors_policy |

**Proposed mitigation or fix:**

1.Only allow trusted sites: Though it may seem apparent, only trustworthy websites should have their origins listed in the Access-Control-Allow-Origin header. Specifically, it is best to avoid dynamically reflecting origins from cross-origin queries without validation as it is easily exploited.

2. Proper configuration of cross-origin requests: The origin should be correctly indicated in the Access-Control-Allow-Origin header if a web resource includes sensitive data. [1]

# 6.Content Security Policy (CSP) Header Not Set

**What is Content Security Policy (CSP) Header Not Set vulnerability?**

A security problem known as Content Security Policy (CSP) Header Not Set happens when a website lacks the CSP header, which specifies which content sources are permitted on the page. By limiting the sources of scripts, styles, pictures, and other resources, a CSP header can stop clickjacking, cross-site scripting, and other assaults. Either an HTML meta element or a server-side command can be used to set a CSP header. A website such as www.securityheaders.io can be used to determine whether your website contains a CSP header. [2] [3]

**What is Content Security Policy (CSP):**

A security standard called Content Security Policy was created to stop XSS assaults, but it also handles several other security risks. For CSP to function, web administrators must first develop a policy that identifies the reliable sources for various resources, including scripts, pictures, stylesheets, and more. Through the process of building a whitelist of sources, CSP lowers the likelihood that malicious material will be loaded or executed on the webpage.

**Affected components:**

1.Web browser**:** Because CSP is implemented at the browser level, the client's web browser is the main component that is impacted. Browsers are susceptible to attacks like XSS and content injections because they are unable to enforce limits on content sources in the absence of a suitable CSP.

2.Web Servers: The CSP headers must be served by web servers in addition to the web content. The entire program is susceptible if a web server is not set up to send the CSP headers. Certain configurations are required for popular servers like Apache, Nginx, and IIS in order to include CSP headers in HTTP responses.

3.APIs: Applications that use frontend scripts to communicate with external APIs or services must be taken into consideration by CSP in order to prevent blocking and to guarantee that no dangerous or unapproved sources are permitted.

4.Development and deployment: Implementing CSP can also be impacted by tools used in the development and deployment pipeline that affect the setting and management of headers. For example, throughout the deployment process, automated build tools must be configured to contain the appropriate header information.

**Impact assessment:**

An attack on the system's security that results from arbitrary code being run. a damaged reputation in addition to a betrayed trust. noncompliance with regulatory norms in breach. After carrying out an impact assessment, a company can adopt suitable security measures and prioritize repair actions, thereby gaining a better knowledge of the potential risks and consequences linked to a route traversal vulnerability.

**Steps to reproduce:**

1.Identify a target website: Choose a web application without a CSP header implemented. This can be confirmed by using command-line tools like curl or by examining the response headers using browser developer tools .

2.Check for input fields: Look for areas of the online application, including search boxes, comment sections, login forms, etc., where user input is received and shown on the website.

3.Confirm lack of CSP: Verify again that there isn't a CSP header present that could prevent inline scripts or scripts loaded from unapproved sources from running. By looking through the HTTP response's headers in the Network tab of the browser's developer tools, you may verify this.

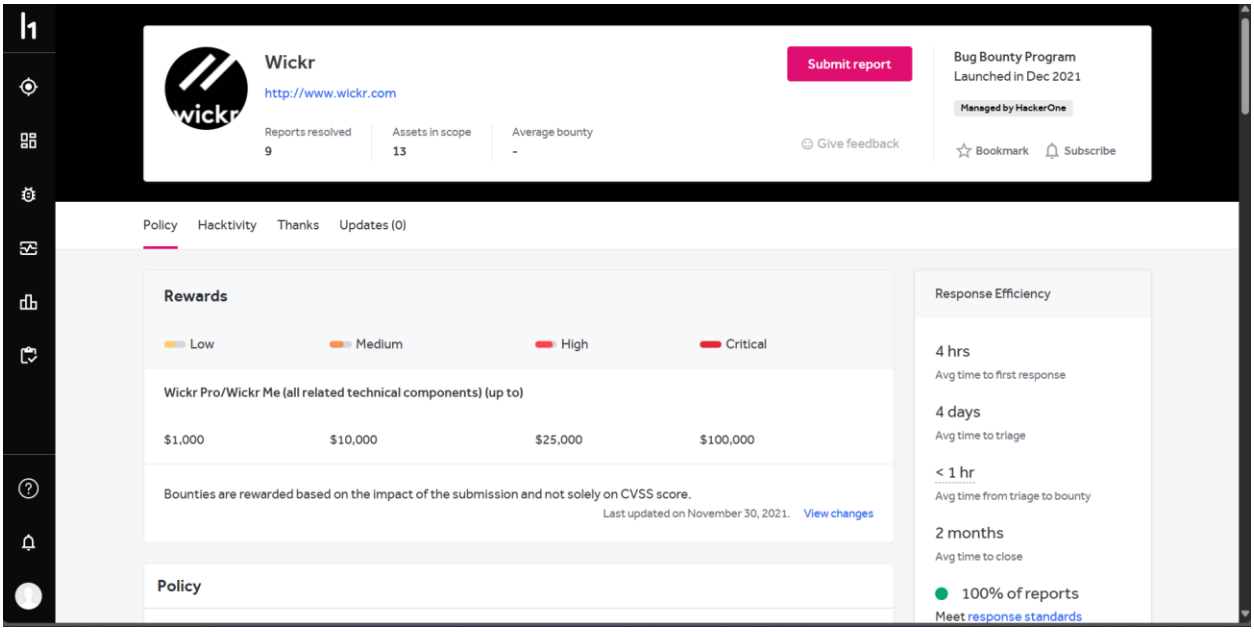
**Proof of concepts:**

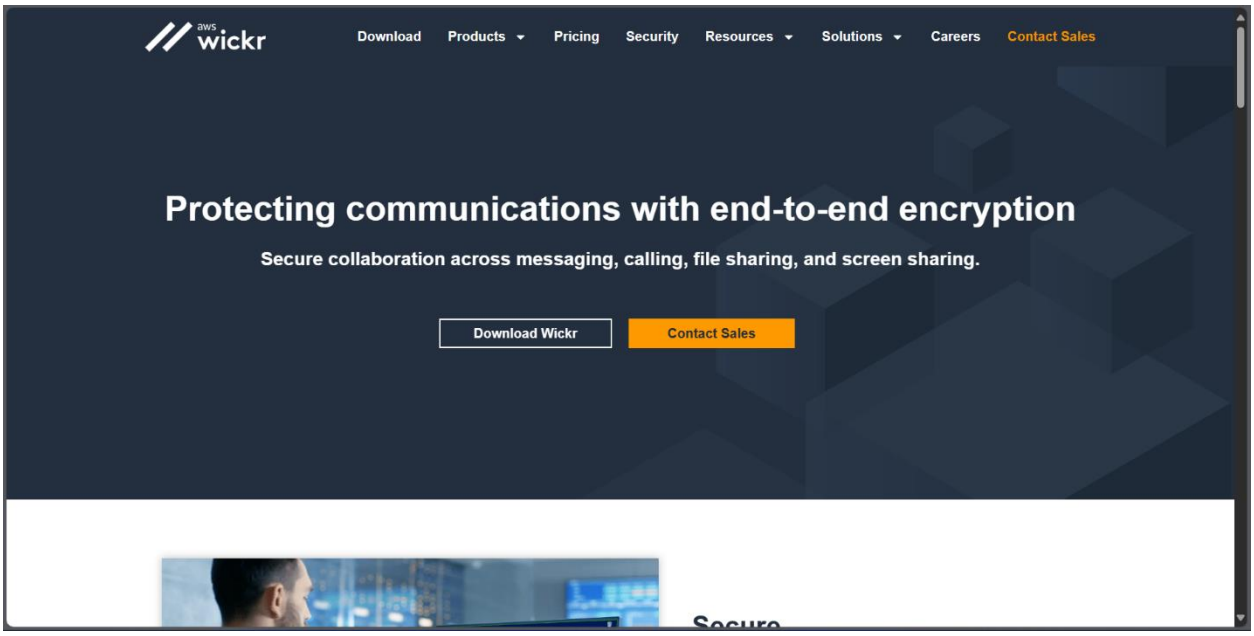
**Information gathering,**

Let's look at the options we have for getting not just the technical details but also other crucial details regarding https://www.yuga.com/. There are numerous approaches we may take to completing this assignment. We are able to make use of all the many methods that could be used to do this assignment. Since Netcraft is the only resource we have right now, let's enter our domain name and see what information we can find. We'll enter our domain name first, and then we'll use the special data that Netcraft has supplied to further investigate it.

Since we are left with no option but to use Netcraft, let's look at our network and see what data we can get. If we follow through on this, we can maximize our time together. Given that Netcraft is presently our only tool of choice, let's see what we

can learn from it. This will allow us to make better use of the few resources we have. Considering that Netcraft is currently our only tool, let's look at the data it can offer. We have nothing but Netcraft at this point, so let's see what we can glean from that.



netcraft

LEARN MORE REPORT FRAUD

Share:

## Background

| | | | |
|---|---|---|---|
| **Site title** | AWS Wickr \| Protecting Communications with End-to-End Encryption | **Date first seen** | December 2004 |
| **Site rank** | 312392 | **Primary language** | English |
| **Description** | Wickr is a single end-to-end encrypted service that provides a full suite of collaboration capabilities on any device. | | |

## Network

| | | | |
|---|---|---|---|
| **Site** | https://www.wickr.com | **Domain** | wickr.com |
| **Netblock Owner** | Amazon.com, Inc. | **Nameserver** | ns-1258.awsdns-29.org |
| **Hosting company** | Amazon | **Domain registrar** | markmonitor.com |
| **Hosting country** | US | **Nameserver organisation** | whois.pir.org |
| **IPv4 address** | 13.224.68.14 (VirusTotal) | **Organisation** | Amazon Technologies, Inc., P.O. Box 8102, Reno, 89507, United States |
| **IPv4 autonomous systems** | AS16509 | **DNS admin** | awsdns-hostmaster@amazon.com |
| **IPv6 address** | 2600:9000:21ca:4600:1d:523f:580:93a1 | **Top Level Domain** | Commercial entities (.com) |

to csp.netcraft.com...

netcraft

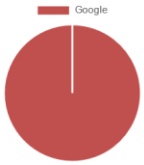
LEARN MORE REPORT FRAUD

## Web Trackers

Web Trackers are third-party resources loaded onto a webpage. Trackable resources include social sharing widgets, javascript files, and images. These trackers can be used to monitor individual user behaviour across the web. Data derived from these trackers are primarily used for advertising or analytics purposes.

**2 known trackers were identified.**

**Companies**

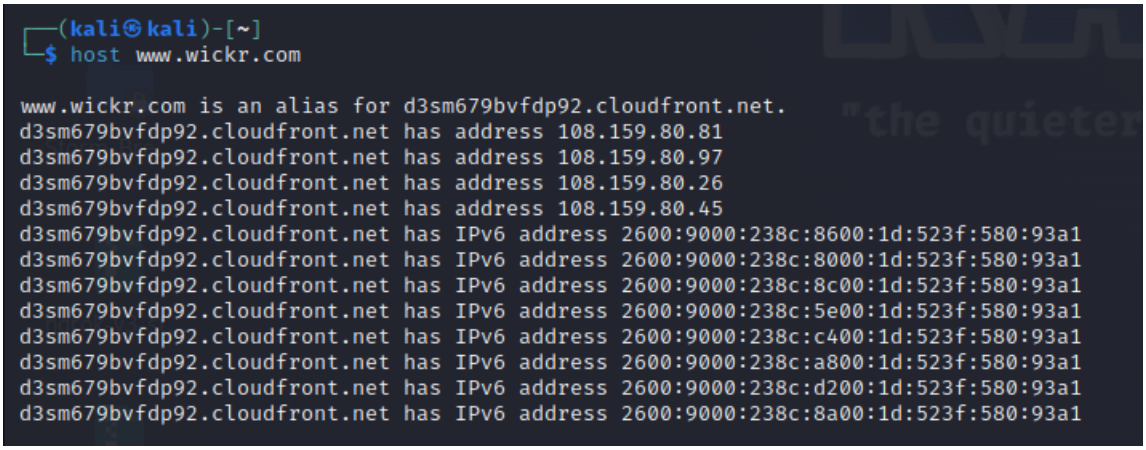



**Categories**


| Company | Primary Category | Tracker | Popular Sites with this Tracker |
|---|---|---|---|
| Google | Analytics | Googletagmanager | www.coingecko.com, www.virustotal.com, www.speedtest.net |
| | CDN | Googlecdn | www.of.moncompteformation.gouv.fr, www.inspq.qc.ca, www.nexusmods.com |

**host** and **nslookup** tools are programs that collect data from the Domain Name System (DNS) using command lines. The DNS assists in converting human-readable domain names into IP addresses that computers use to identify one another on a network.

**Host**



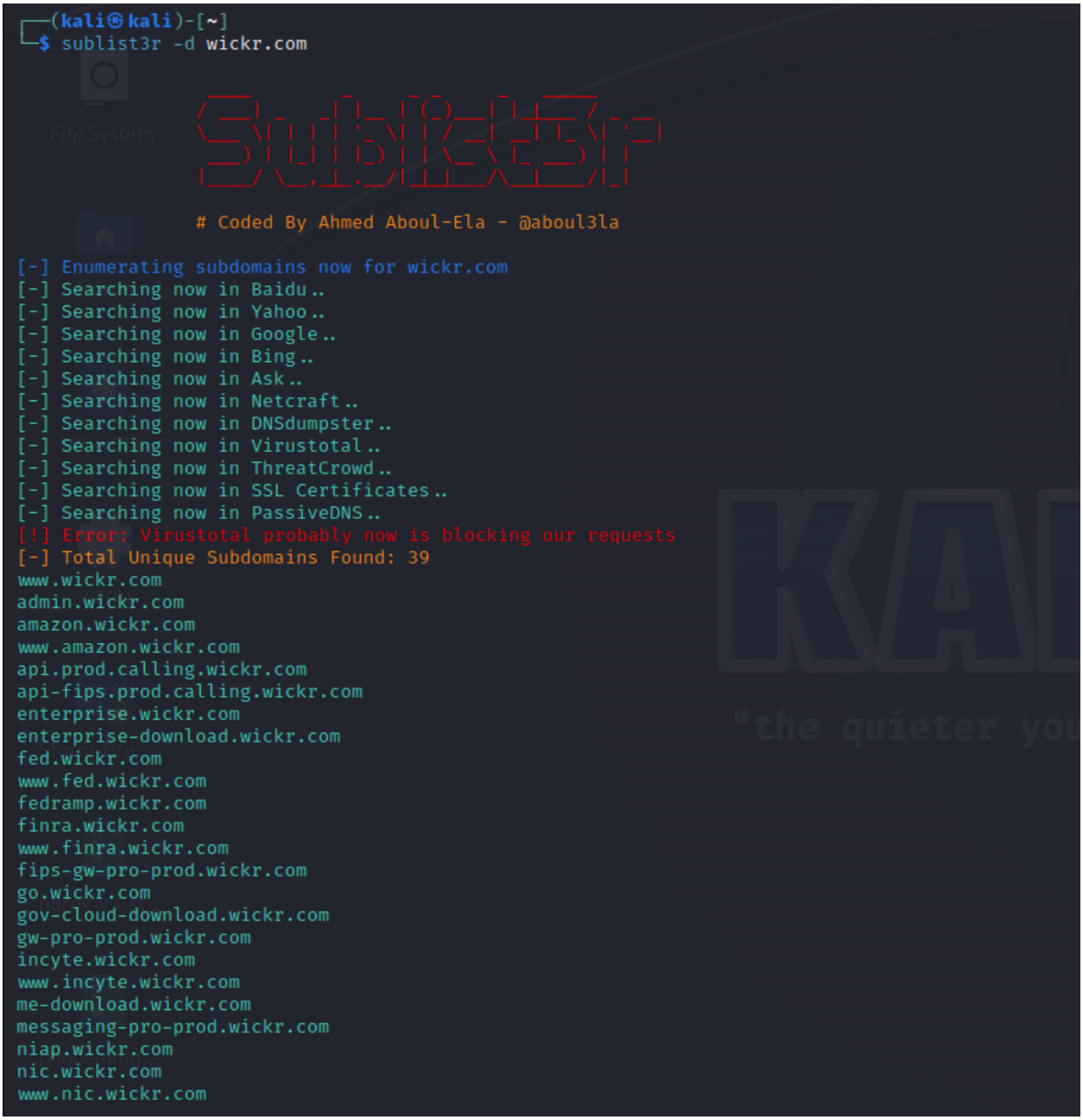
**Nslookup**

```
┌──(kali㉿kali)-[~]
└─$ nslookup www.wickr.com
Server:         192.168.43.1
Address:        192.168.43.1#53

Non-authoritative answer:
www.wickr.com   canonical name = d3sm679bvfdp92.cloudfront.net.
Name:   d3sm679bvfdp92.cloudfront.net
Address: 108.159.80.97
Name:   d3sm679bvfdp92.cloudfront.net
Address: 108.159.80.81
Name:   d3sm679bvfdp92.cloudfront.net
Address: 108.159.80.45
Name:   d3sm679bvfdp92.cloudfront.net
Address: 108.159.80.26
Name:   d3sm679bvfdp92.cloudfront.net
Address: 2600:9000:238c:ac00:1d:523f:580:93a1
Name:   d3sm679bvfdp92.cloudfront.net
Address: 2600:9000:238c:9c00:1d:523f:580:93a1
Name:   d3sm679bvfdp92.cloudfront.net
Address: 2600:9000:238c:2400:1d:523f:580:93a1
Name:   d3sm679bvfdp92.cloudfront.net
Address: 2600:9000:238c:4800:1d:523f:580:93a1
Name:   d3sm679bvfdp92.cloudfront.net
Address: 2600:9000:238c:c00:1d:523f:580:93a1
Name:   d3sm679bvfdp92.cloudfront.net
Address: 2600:9000:238c:4c00:1d:523f:580:93a1
Name:   d3sm679bvfdp92.cloudfront.net
Address: 2600:9000:238c:4400:1d:523f:580:93a1
Name:   d3sm679bvfdp92.cloudfront.net
Address: 2600:9000:238c:b200:1d:523f:580:93a1
```

Use nmap and check the open ports,

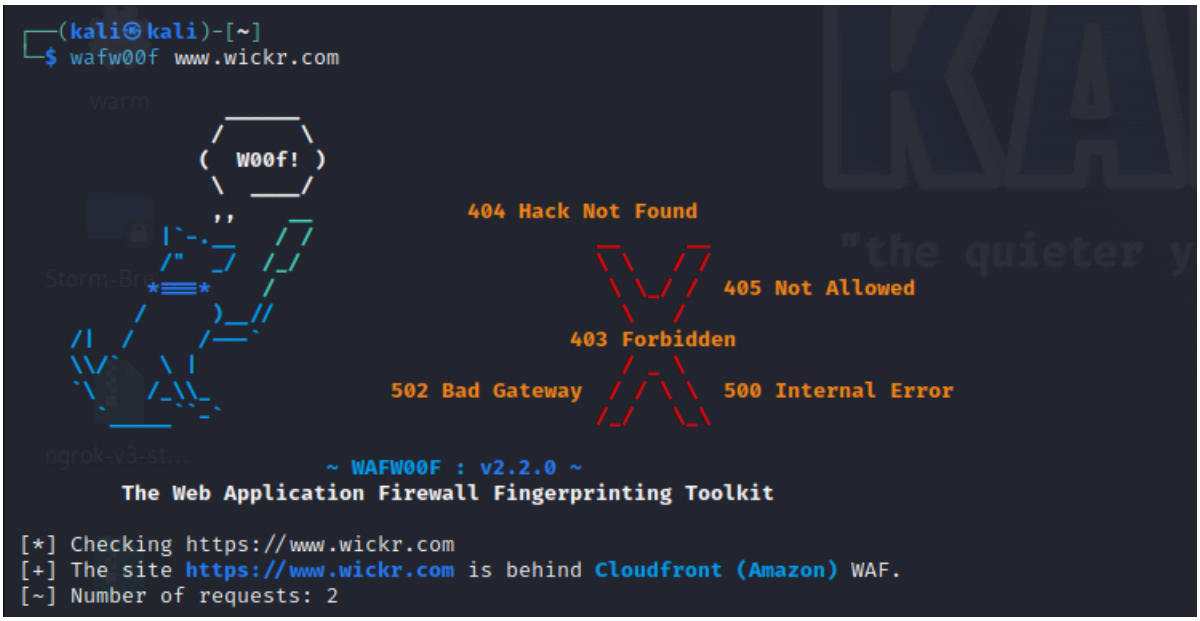
```
┌──(kali㉿kali)-[~]
└─$ nmap -Pn 108.159.80.97

Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-26 13:16 EDT
Nmap scan report for server-108-159-80-97.bom78.r.cloudfront.net (108.159.80.97)
Host is up (0.036s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE
53/tcp  open  domain
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 13.59 seconds
```

By using "**sublist3r**" like tool we can find subdomains of the main,

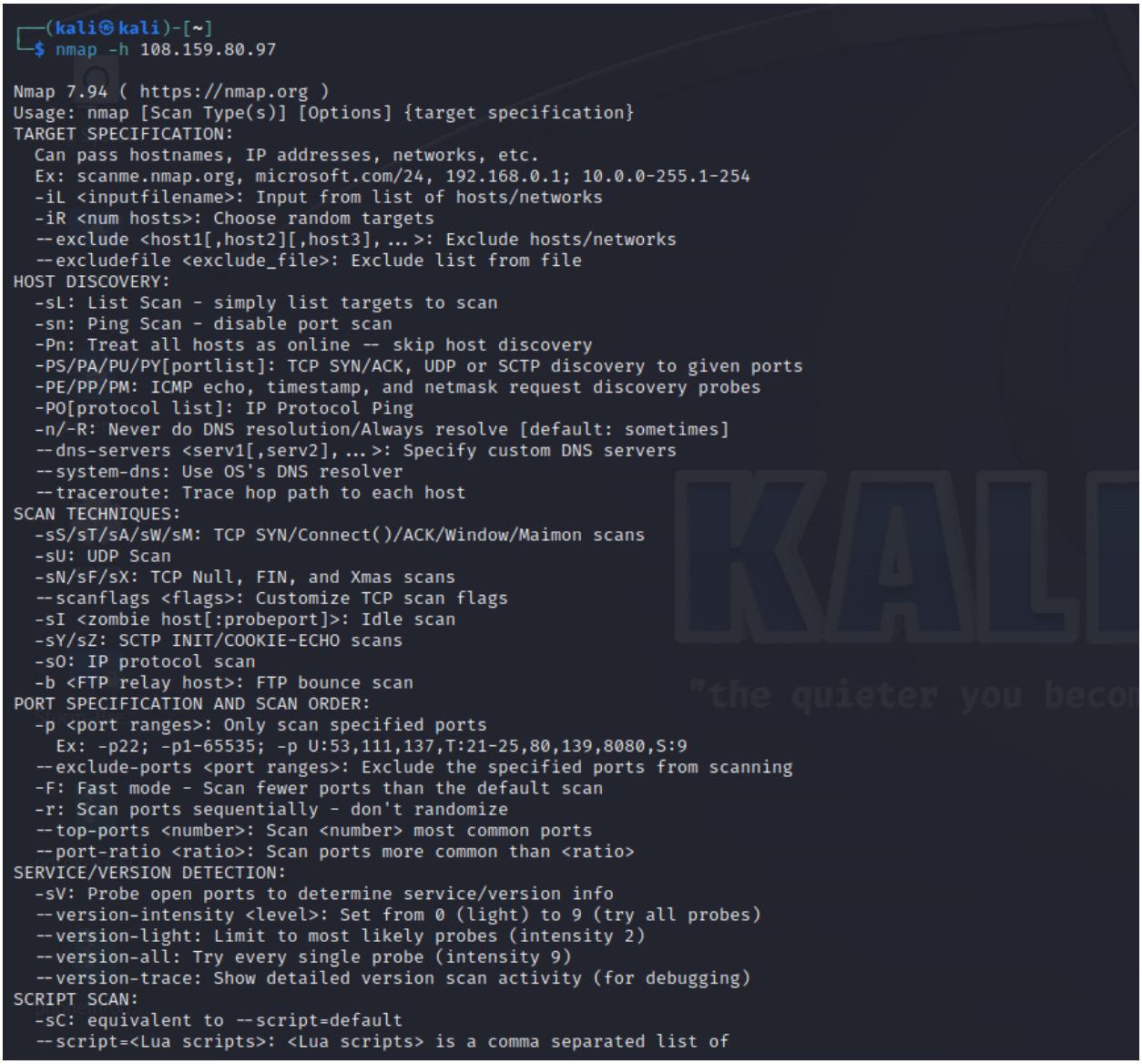


**Waf00f tool:**

Knowing if a web server is protected by a WAF and identifying the type of WAF in use are essential steps in a security assessment because they affect the methods and resources a tester can utilize to further examine the web server for vulnerabilities.



## Vulnerability scan using nmap,



# OWASP ZAP Report

## Alert counts by site and risk

This table shows, for each site for which one or more alerts were raised, the number of alerts raised at each risk level.

Alerts with a confidence level of "False Positive" have been excluded from these counts.

(The numbers in brackets are the number of alerts raised for the site at or above that risk level.)

| | | Risk | | | |
|---|---|---|---|---|---|
| | | High (= High) | Medium (>= Medium) | Low (>= Low) | Informational (>= Informational) |
| Site | https://www.wickr.com | 1 (1) | 2 (3) | 4 (7) | 4 (11) |

### Risk=Medium, Confidence=High (1)

https://www.wickr.com (1)

Content Security Policy (CSP) Header Not Set (1)

▶ GET https://www.wickr.com/

### Content Security Policy (CSP) Header Not Set

| | |
|---|---|
| **Source** | raised by a passive scanner (Content Security Policy (CSP) Header Not Set) |
| **CWE ID** | 693 |
| **WASC ID** | 15 |
| **Reference** | ▪ https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy<br>▪ https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html<br>▪ https://www.w3.org/TR/CSP/<br>▪ https://w3c.github.io/webappsec-csp/<br>▪ https://web.dev/articles/csp<br>▪ https://caniuse.com/#feat=contentsecuritypolicy<br>▪ https://content-security-policy.com/ |

**Proposed mitigation or fix**

You must set up your web server to return the Content-Security-Policy HTTP Header and provide values to control the resources that the browser is permitted to load for your page in order to resolve Content Security Policy (CSP) Header Not Set.

# 7.Absence of anti CSRF token

### What is CSRF?

One kind of security flaw is called Cross-Site Request Forgery (CSRF), and it attacks the faith a web application has in a user's browser. When a person is authenticated to a web service, it enables an attacker to trick them into doing things they do not want to. CSRF takes advantage of the fact that every time a web browser makes a request to a domain, it automatically sends cookies related to that domain, including session cookies that serve as the user's authentication.

In essence, this attack takes control of a user's web session and uses it to carry out unwanted tasks. It can be lessened in a number of ways, one of which is the use of anti-CSRF tokens, which are erratic, secret values created by the server and necessary for each state-changing request that the client submits. These tokens guarantee that the request is being sent voluntarily by the user and isn't being controlled by a malevolent website. Other tactics include ensuring that requests come from reliable sources by examining the Referer and Origin HTTP headers and limiting the way cookies are sent with cross-site requests by using the SameSite cookie attribute. In order to prevent cross-site scripting vulnerabilities (CSRFs), developers must be aware of these risks and take appropriate precautions when designing their web apps.

**Affected components:**

**1.Session management mechanism:** Sessions are used by web applications to recognize and control user state during various interactions. Because each request includes a session cookie that verifies the request's legitimacy from the client, CSRF attacks take use of these sessions.

**2.Forms and URLs:** CSRF attacks can exploit any form or URL that causes a state change (such as a transaction or database update) in the web application. This comprises, among other things, forms for updating user profiles, changing passwords, and completing financial transactions.

**3.Authentication Cookies:** Since CSRF attacks rely on the improper usage of the user's authenticated session, these are essential. These cookies are automatically included in any request made to the server by the browser of an authenticated user, giving the impression that the requests are valid.

**4.End users:** In the end, CSRF attacks have an impact on the application's end users. In the absence of appropriate security measures, their accounts may be misused to carry out unauthorized tasks, which could result in account takeover, data theft, or financial loss.

**Impact assessment:**

Conducting an impact assessment on Cross-Site Request Forgery (CSRF) vulnerabilities is imperative in order to comprehend the possible harm that could be incurred by a business and its users in the event that these vulnerabilities are leveraged. Analyzing the impact entails looking at the functionality and data handled by the impacted program, as well as the possible repercussions of illegal acts.

**1.Identify risks:** Determine which crucial data (such as user information or financial information) and operations (such as password changes or transaction processing) may be impacted in the event of a cross-site request forgery (CSRF) attack.

**2.Estiamate the damage:** Consider the potential fallout from an assault, including monetary losses, private information being revealed, reputational harm to the business, and legal issues in the event that compliance guidelines are broken.

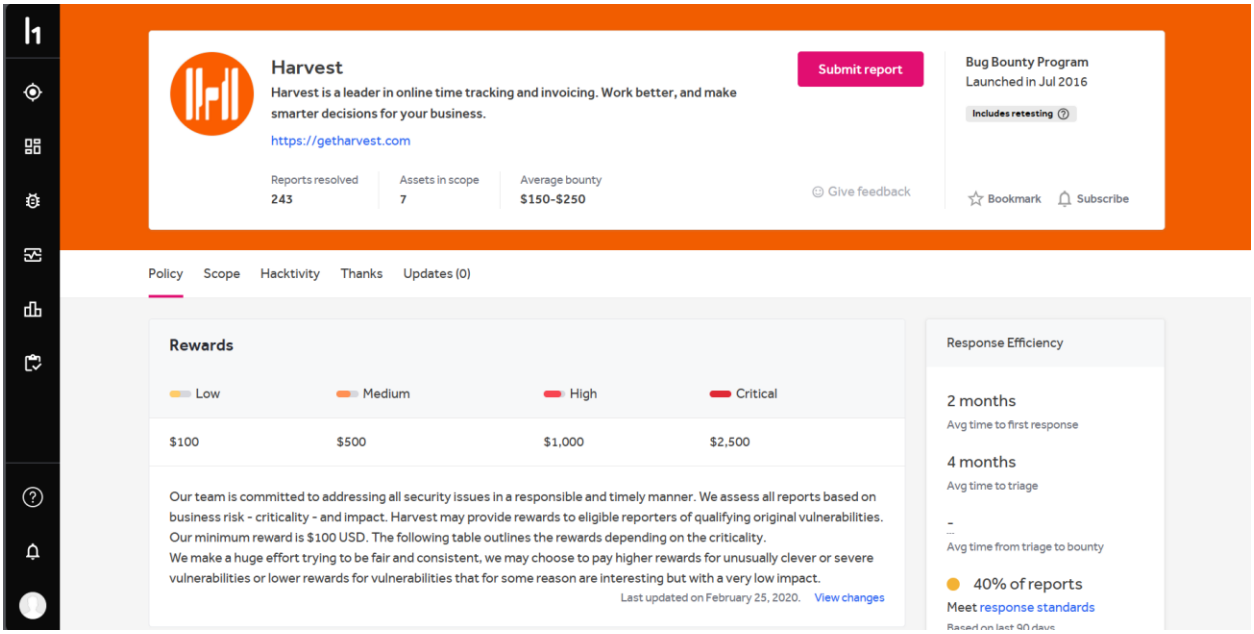
**3.Plan to prevent:** Choose the best defenses against CSRF attacks, such as appropriate use of cookies and security tokens, and how to react in the event that an attack occurs.

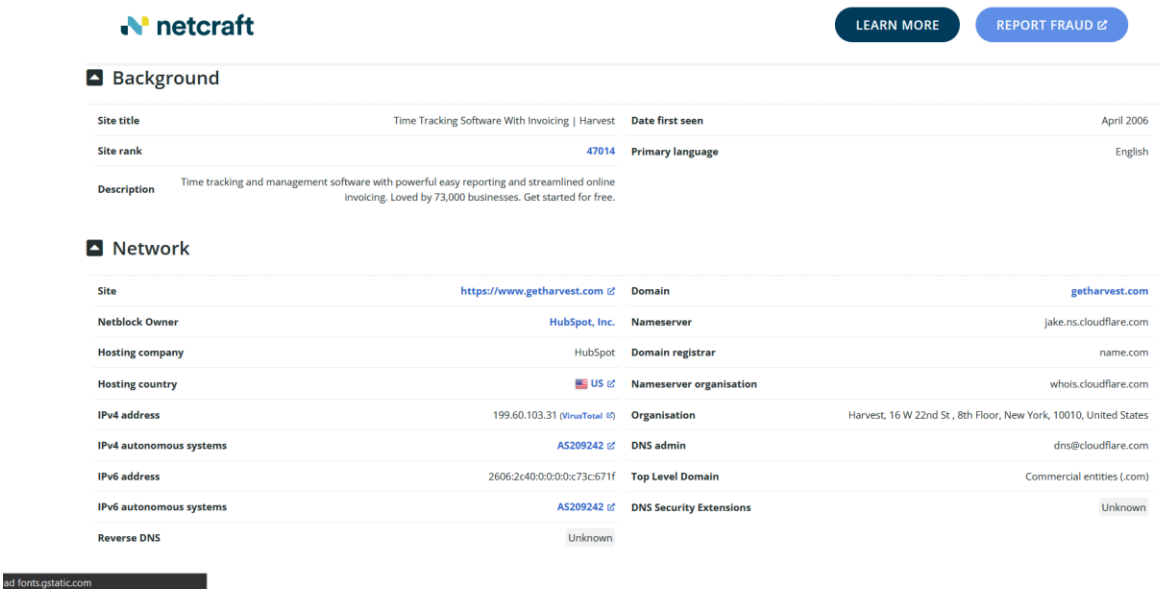
**Steps to reproduce:**

For testing reasons, you first need to identify a crucial step in a web service, like changing account information or completing a transaction, in order to replicate a Cross-Site Request Forgery (CSRF) attack. After that, you watch how the program manages session authentication, paying close attention to any actions that don't require any further security checks and instead rely only on the user's session cookies. Next, you craft a malicious webpage with an auto-submit form that uses the victim's authenticated session to carry out the intended activity. This form is designed to be submitted to the application upon a visitor's arrival to your malicious website.

You open your malicious page in a separate browser tab and log into the target application to test the vulnerability. A cross-site request for input vulnerability (CSRF) is confirmed when an action is processed by the application without requiring additional user engagement or verification. Through testing, web applications' security flaws are found and fixed, and it is ensured that sensitive actions need more than just an authenticated session to be approved.

**Proof of concepts:**






Programs called **host** and **nslookup** tools use command lines to retrieve data from the Domain Name System (DNS). The DNS helps translate IP addresses, which computers use to identify one another on a network, into human-readable domain names.

## Host

```
┌──(kali㉿kali)-[~]
└─$ host getharvest.co
getharvest.co is an alias for pr-suspensions.go.co.
pr-suspensions.go.co is an alias for pr-suspensions-neuweb-biz.expedrion.biz.
pr-suspensions-neuweb-biz.expedrion.biz is an alias for Registry-Web-Suspension-1912215664.us-east-1.elb.amazonaws.com.
Registry-Web-Suspension-1912215664.us-east-1.elb.amazonaws.com has address 52.205.79.79
Registry-Web-Suspension-1912215664.us-east-1.elb.amazonaws.com has address 52.200.89.194
```

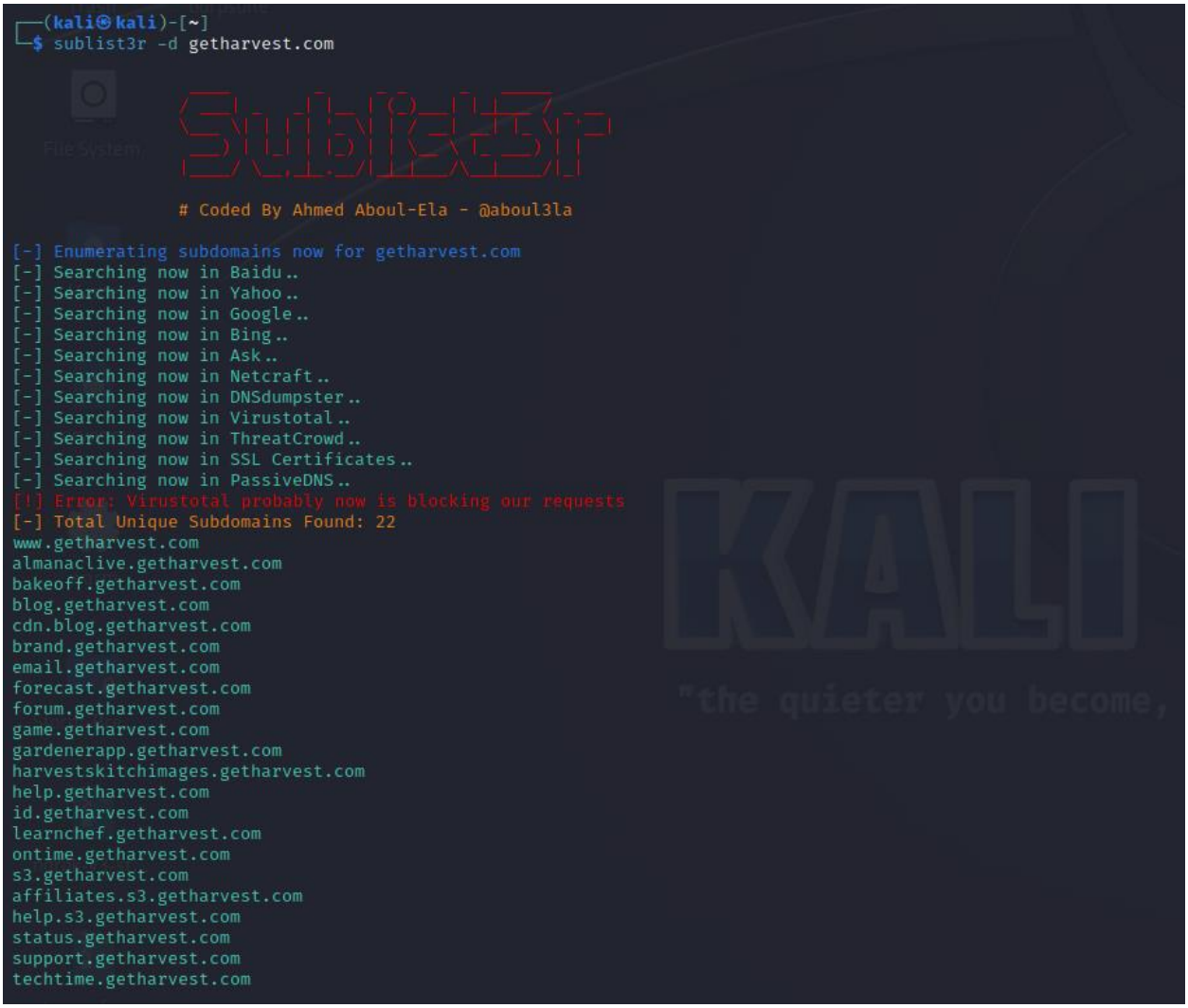
## Nslookup

```
┌──(kali㉿kali)-[~]
└─$ nslookup www.getharvest.com
Server:         192.168.43.1
Address:        192.168.43.1#53

Non-authoritative answer:
www.getharvest.com      canonical name = 19495563.group13.sites.hubspot.net.
19495563.group13.sites.hubspot.net      canonical name = group13.sites.hscoscdn10.net.
Name:   group13.sites.hscoscdn10.net
Address: 199.60.103.31
Name:   group13.sites.hscoscdn10.net
Address: 199.60.103.225
Name:   group13.sites.hscoscdn10.net
Address: 2606:2c40::c73c:67e1
Name:   group13.sites.hscoscdn10.net
Address: 2606:2c40::c73c:671f
```

## Whatweb

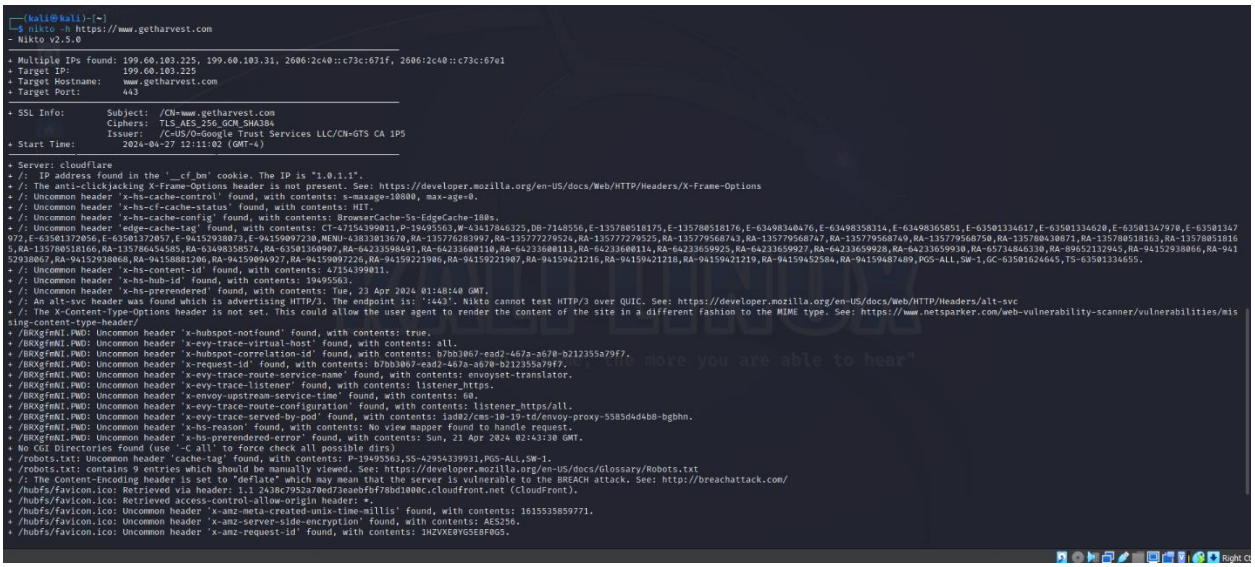
## Sublist3r

To locate subdomains within a specific domain, we utilize Sublist3r. It's a program that finds active and perhaps helpful subdomains, assisting penetration testers and security researchers in mapping the domain architecture.



## Nikto

Using Nikto checking the vulnerability:



# OWASP ZAP Report

## Alert counts by site and risk

This table shows, for each site for which one or more alerts were raised, the number of alerts raised at each risk level.

Alerts with a confidence level of "False Positive" have been excluded from these counts.

(The numbers in brackets are the number of alerts raised for the site at or above that risk level.)

| | | Risk | | | |
|---|---|---|---|---|---|
| | | High (= High) | Medium (>= Medium) | Low (>= Low) | Informational (>= Informational) |
| Site | https://www.getharvest.com | 1 (1) | 6 (7) | 9 (16) | 9 (25) |

## Absence of Anti-CSRF Tokens

| | |
|---|---|
| **Source** | raised by a passive scanner (Absence of Anti-CSRF Tokens) |
| **CWE ID** | 352 |
| **WASC ID** | 9 |
| **Reference** | ▪ https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html<br>▪ https://cwe.mitre.org/data/definitions/352.html |

**Risk=Medium, Confidence=Low (1)**

**https://www.getharvest.com (1)**

**Absence of Anti-CSRF Tokens (1)**

▶ GET https://www.getharvest.com/

## Proposed mitigation or fix

**1.Use secret cookies:** Recall that every request will include cookies—even the ones that are hidden. Whether the end-user was duped into making the request or not, all authentication tokens will be submitted. Moreover, the application container uses session IDs just to link the request to a particular session object. The end-user's intention to submit the request is not confirmed by the session identifier.

**2.URL rewriting:** Since the attacker is unable to predict the victim's session ID, this could be considered a helpful CSRF prevention approach. Nonetheless, the URL exposes the user's session ID. It is not advisable to address one security vulnerability by creating a new one.

**3.HTTPS:** HTTPS is insufficient on its own to prevent CSRF attacks. Nonetheless, HTTPS need to be regarded as a requirement for the credibility of any precautionary actions. [4]

# 8.DOS (Denial of service)

**What is DOS?**

A malicious attempt to stop a targeted server, service, or network from operating normally by flooding the target or its surrounding infrastructure with excessive amounts of Internet traffic is known as a denial of service (DoS) attack. DoS attacks function by sending a large number of fictitious requests to a target, usually with the intention of consuming up all of the server's resources or bandwidth, all from a single Internet-connected device (one network connection). Due to this overload, the server is unable to process valid requests, which may cause it to crash or drastically slow down.

Such attacks can have a variety of effects, from slight irritation to serious interruptions at the corporate level and data breaches. DoS attacks are often used by hackers to intimidate, disrupt, or protest against businesses or organizations. Despite their crude and unsophisticated nature, they have the ability to take down websites and render enterprises inoperable.

**Affected components for DOS:**

A Denial of Service (DoS) attack has the potential to impact many network or system components. Since web servers are directly accessible from the internet, they are frequently the primary target. These servers host websites and other services. Malicious traffic can overwhelm the network infrastructure itself, which includes firewalls, load balancers, and routers. This can cause other systems and users who use the same network resources to experience a degradation in service in addition to the direct target. Moreover, the cascade of traffic may cause databases and application servers connected to the main servers to become unavailable or overburdened. The user experience is severely compromised, which goes beyond the immediate hardware and software and could result in a loss of customer trust as well as monetary losses.

**Impact Assessment:**

**1.Infastructure impact:** DoS attacks have the potential to overwhelm your servers and network with excessive traffic, slowing or taking them offline. This has an impact on the functionality of any network-dependent services as well as the availability of your website.

**2.Security and compliance:** Other potential security flaws could be disregarded during a denial-of-service assault, raising the possibility of more problems. Additionally, if the attack compromises data, it may result in legal issues, particularly if data protection laws are broken.

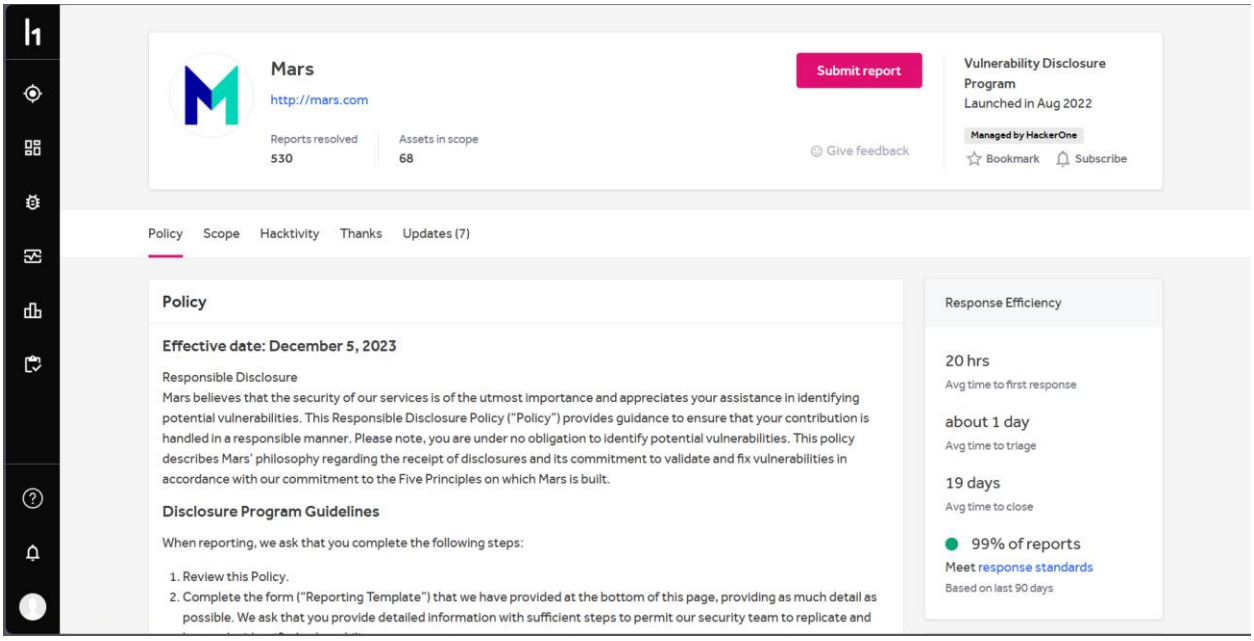
**3.Reputaion impact:** Denial  severe attacks might harm the reputation of your company by making potential clients reconsider utilizing your services because they may view you to be unreliable or to have security vulnerabilities.

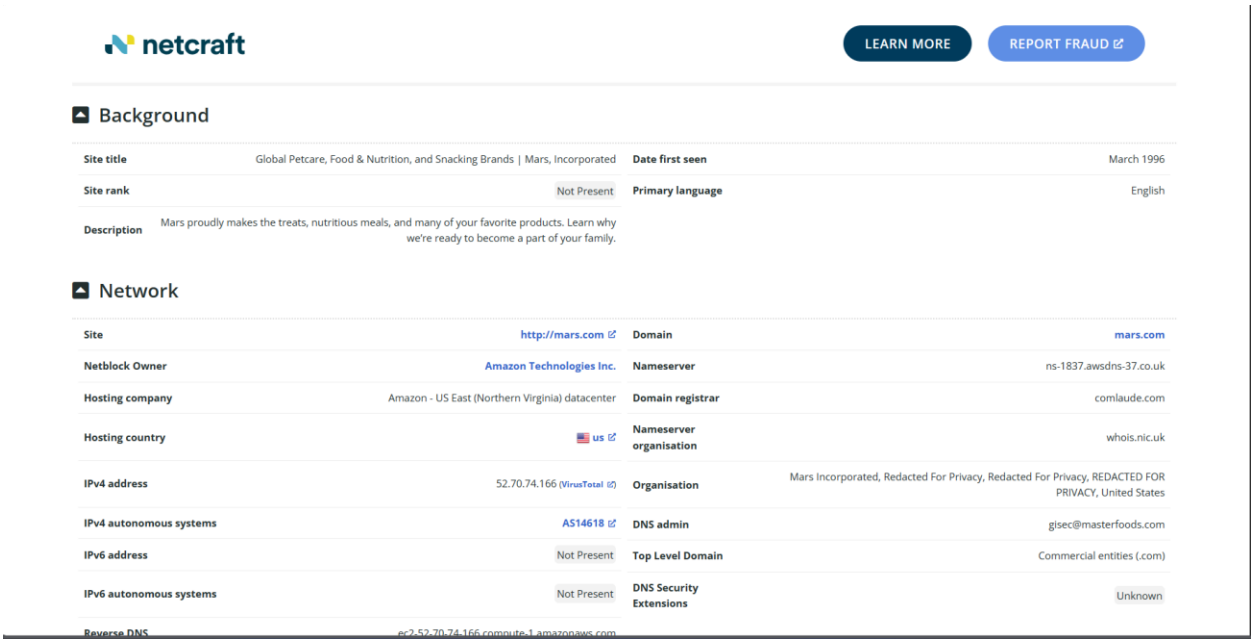
**Steps to reproduce:**

Simulating a denial-of-service attack on a computer is not a good idea because it could seriously harm the system and prevent legitimate users from using its services. However, if you wish to determine how a machine was impacted by a DoS attack in the past, there are a few things you may do. Determine the type of attack and its duration by examining server logs and additional data. Examine the logs and further data. With this information, you can determine how the assault impacted the computer and the services it provided. Examine the performance of the server. To determine whether the attack is still having an impact on the server, check its speed. Examine the system for indications of persistent issues such as high CPU consumption, delayed response times, or other issues.

Verify that the server's services are operating properly by checking them. Prior to restoring the server to regular operation, confirm that the denial-of-service attack did not disrupt any services. Put countermeasures in place: Take action, if you haven't already, to reduce the likelihood that denial-of-service attacks may occur. To increase security, this can entail installing firewalls or other security measures, making hardware or software changes, or collaborating with a third-party service.

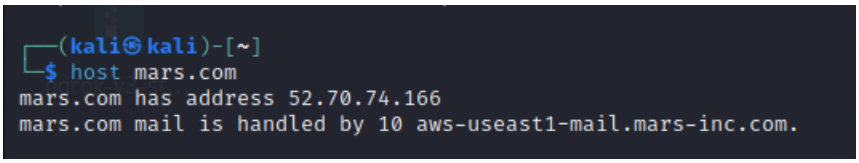
# Proof of concepts:



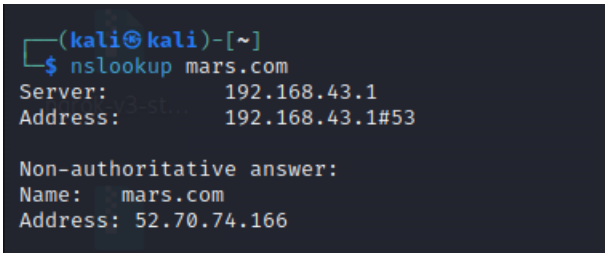



Programs called **host** and **nslookup** tools use command lines to retrieve data from the Domain Name System (DNS). The DNS helps translate IP addresses, which computers use to identify one another on a network, into human-readable domain names.

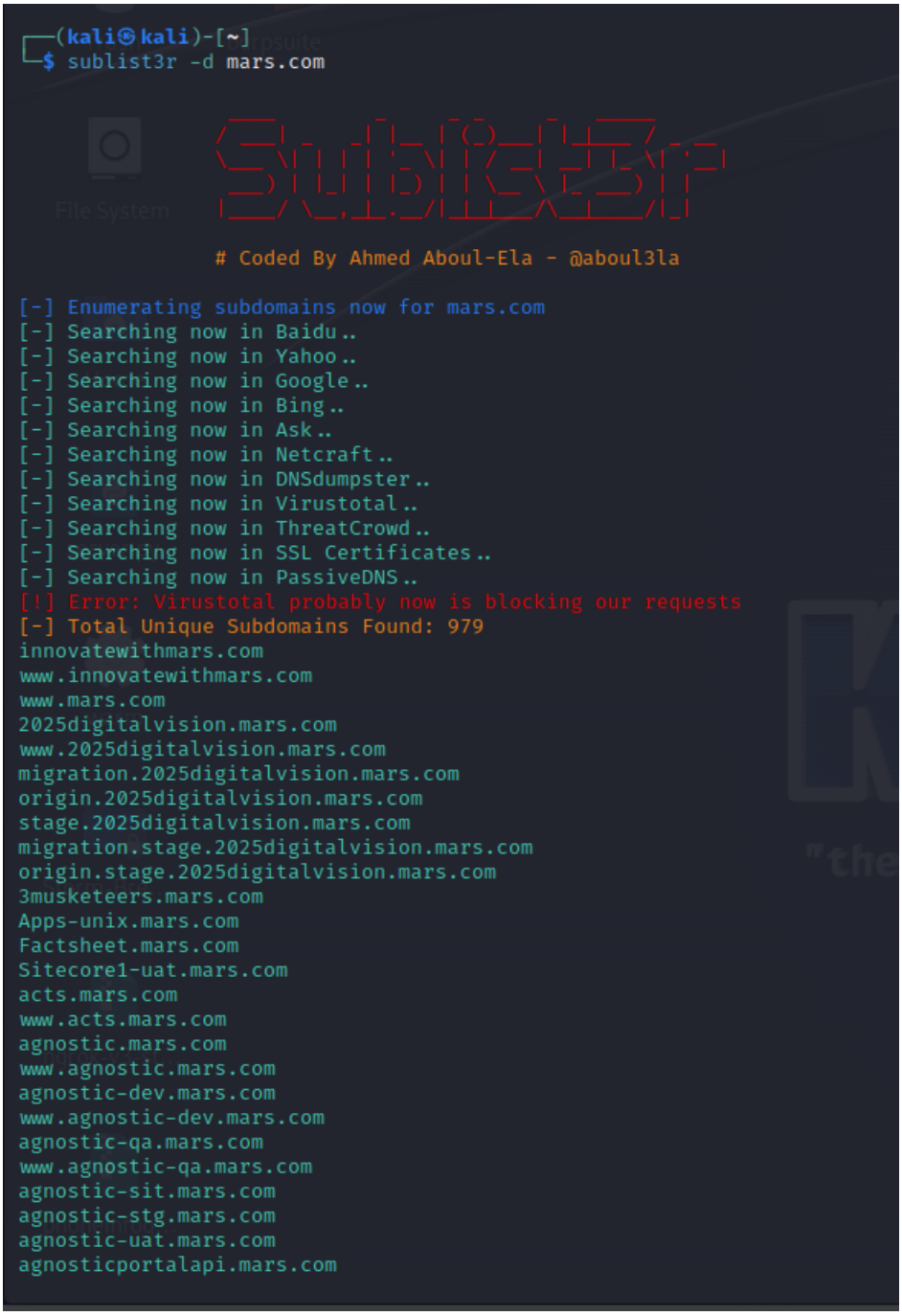
## Host




## Nslookup



## Sublist3r


┌──(kali㉿kali)-[~]
└─$ sublist3r -d mars.com
# Coded By Ahmed Aboul-Ela - @aboul3la
[-] Enumerating subdomains now for mars.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[!] Error: Virustotal probably now is blocking our requests
[-] Total Unique Subdomains Found: 979
innovatewithmars.com
www.innovatewithmars.com
www.mars.com
2025digitalvision.mars.com
www.2025digitalvision.mars.com
migration.2025digitalvision.mars.com
origin.2025digitalvision.mars.com
stage.2025digitalvision.mars.com
migration.stage.2025digitalvision.mars.com
origin.stage.2025digitalvision.mars.com
3musketeers.mars.com
Apps-unix.mars.com
Factsheet.mars.com
Sitecore1-uat.mars.com
acts.mars.com
www.acts.mars.com
agnostic.mars.com
www.agnostic.mars.com
agnostic-dev.mars.com
www.agnostic-dev.mars.com
agnostic-qa.mars.com
www.agnostic-qa.mars.com
agnostic-sit.mars.com
agnostic-stg.mars.com
agnostic-uat.mars.com
agnosticportalapi.mars.com

Checking vulnerabilities by using **nmap**

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap 52.70.74.166
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-20 11:48 EDT
Nmap scan report for ec2-52-70-74-166.compute-1.amazonaws.com (52.70.74.166)
Host is up (0.034s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT    STATE SERVICE
22/tcp  open  ssh
53/tcp  open  domain
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 13.48 seconds
```

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -script vuln 52.70.74.166
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-20 11:50 EDT
Nmap scan report for ec2-52-70-74-166.compute-1.amazonaws.com (52.70.74.166)
Host is up (0.093s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT    STATE SERVICE
22/tcp  open  ssh
53/tcp  open  domain
80/tcp  open  http
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
443/tcp open  https
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf: Couldn't find any CSRF vulnerabilities.

Nmap done: 1 IP address (1 host up) scanned in 135.45 seconds
```

**Proposed mitigation or fix**

Organizations can use a number of practical tactics to lessen Denial of Service (DoS) assaults. To start, limit the amount of traffic a server takes for a predetermined amount of time by using rate limiting to assist avoid overload. Putting web application firewalls (WAFs) in place can assist in identifying and obstructing harmful traffic patterns. Blocking specific regions can be helpful if assaults are coming from those areas. By spreading your resources across several servers or locations, you may create redundancy in your network, which guarantees that even in the event of a server failure, others can manage the load and preserve service availability. Distributing load and absorbing surges in traffic can also be achieved by using a content delivery network (CDN).

Keeping an incident response plan created especially for denial-of-service (DoS) assaults will also guarantee that your team is prepared to react promptly and efficiently in the event of an attack, reducing possible harm and speeding up the restoration of services.

# 9. Strict-Transport-Security Header Not Set

### What is HSTS (HTTP Strict Transport Security)?

A web security policy tool called HTTP Strict Transport Security (HSTS) aids in defending websites from man-in-the-middle attacks such protocol downgrade assaults and cookie hijacking. Websites that declare their HSTS policy tell web browsers to only connect to them using secure HTTPS connections—instead of less secure HTTP connections. This is accomplished by the browser receiving a specific header (Strict-Transport-Security) from the server that tells it how long to remember to only use HTTPS to access the webpage. By including directives like include Subdomains, which requires HTTPS for all subdomains, and preload, which permits websites to be added to a preloaded HSTS list that browsers consult before the initial visit, the HSTS policy can be improved.

When HSTS is properly implemented, all user-browser communications with the website are encrypted, providing protection against some cyberthreats that take advantage of the less secure HTTP protocol.

**Affected components:**

A security feature that websites utilize to improve the security of connections between web browsers and servers is called HTTP Strict Transport Security (HSTS). When a website employs HTTP Strict Transport Security (HSTS), it transmits a unique response header to the browser, telling it to connect exclusively through a secure HTTPS connection for a certain amount of time, instead of using an insecure HTTP connection. This lessens the possibility of hackers intercepting or altering data transmitted between the user and the website, which is a common danger connected to insecure HTTP connections. Websites that employ HSTS make sure that all communications are secure and encrypted, shielding user information from typical dangers like man-in-the-middle attacks. By doing this, you may improve the website's general security and dependability while also protecting user information.

**Impact assessment:**

**1.Increased Vulnerability to Man-in-the-Middle (MitM) Attacks:** Websites without HSTS are vulnerable to attacks in which a hacker intercepts or modifies data being transferred between the client and the server, possibly resulting in data modification or theft.

**2.Downground Attacks:** By forcing connections to switch from secure HTTPS to less secure HTTP, attackers could allow for more exploitation, such as password theft.

**3.Damage the user trust:** People are becoming more conscious of online safety. Insufficient security protocols, particularly those that are obvious like HTTPS, may discourage people from utilizing a service.

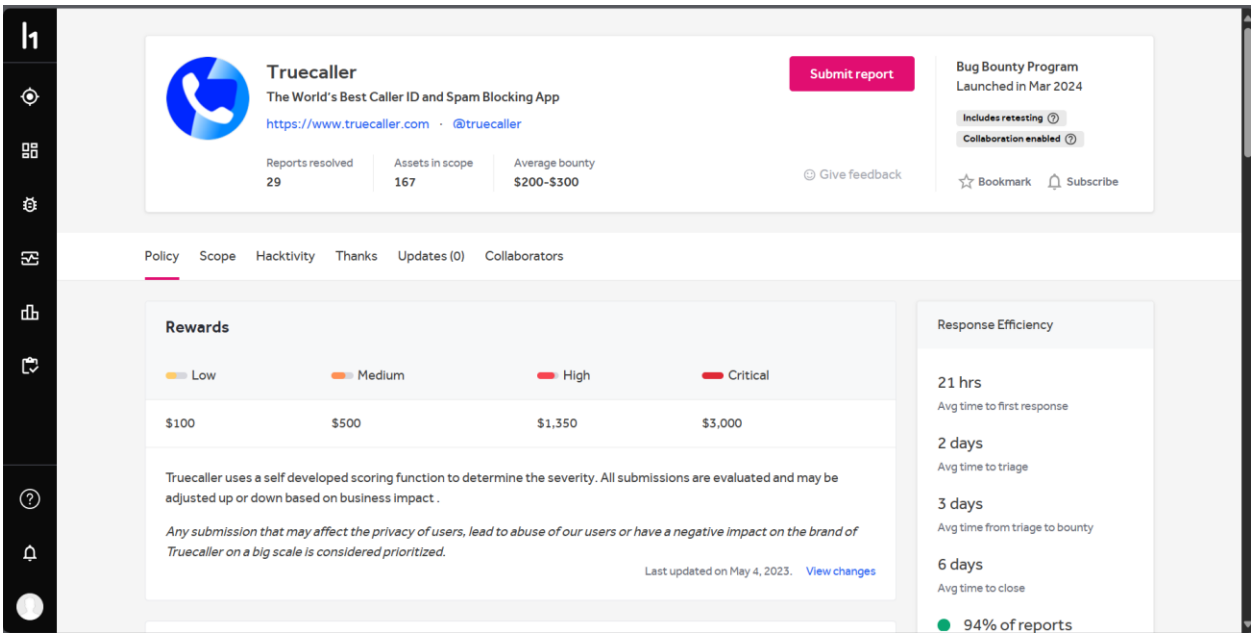
**4. Disturbance in Business Functions:** Operational disruptions can result from security breaches or data loss incidents, ranging from the loss of important data to the need to suspend operations in order to remediate a breach.

**Steps to reproduce:**

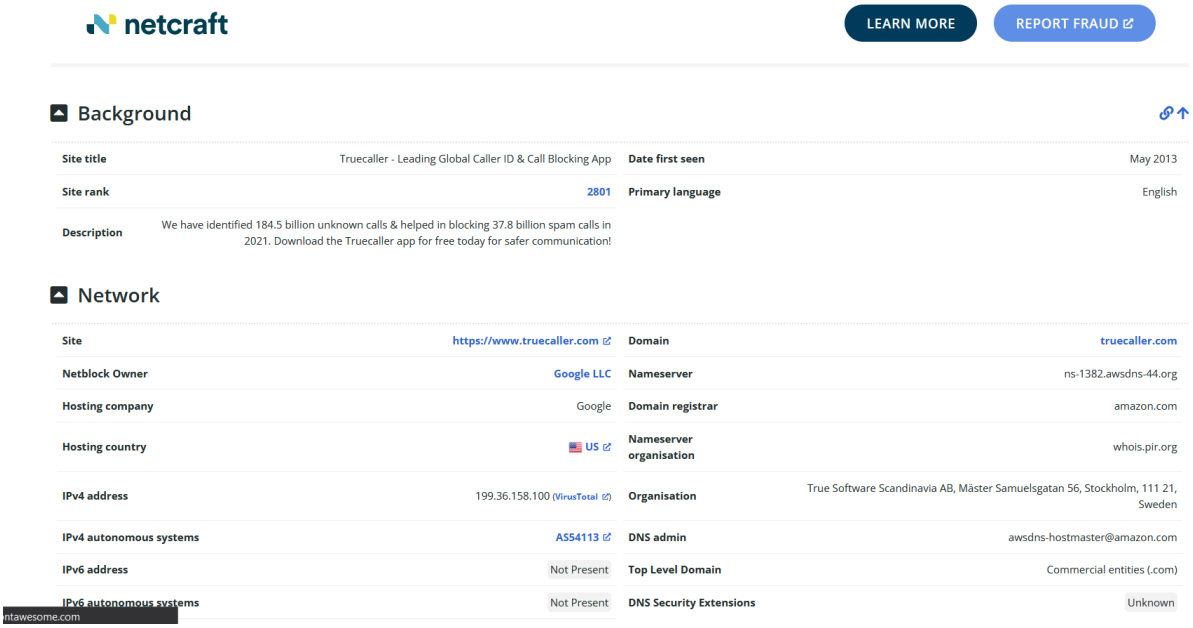
You would do a series of actions intended to illustrate the possibility of security breaches if HTTP Strict Transport Security (HSTS) is not implemented, in order to replicate a vulnerability directly linked to the lack of HSTS. Find a website that sends HTTPS replies without the Strict-Transport-Security header first. You may verify this by looking at the headers for any HTTPS response from the website using the Network tab of the browser developer tools. The next stage is to mimic or carry out an SSL stripping attack if such a site has been found. Putting the attacker in a position to intercept client-server communication is part of this.

Because HSTS isn't present, tools like ‘sslstrip’ can be used to intercept and change HTTPS traffic to HTTP. The attacker can watch unencrypted HTTP traffic while the victim interacts with what they think is a secure website, gathering private data like session tokens and login credentials. By demonstrating this vulnerability, it is made clear how dangerous it is to send data over connections that have the potential to revert from HTTPS to HTTP, and how crucial it is to put HSTS in place to ensure secure connections.

## Proof of concepts:

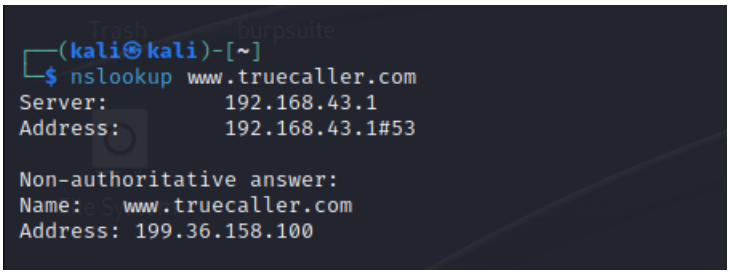


By using **netcraft** we can gather the information about this web site.

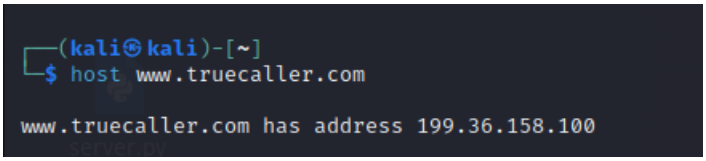



Programs called **host** and **nslookup** tools use command lines to retrieve data from the Domain Name System (DNS). The DNS helps translate IP addresses, which computers use to identify one another on a network, into human-readable domain.

## Nslookup

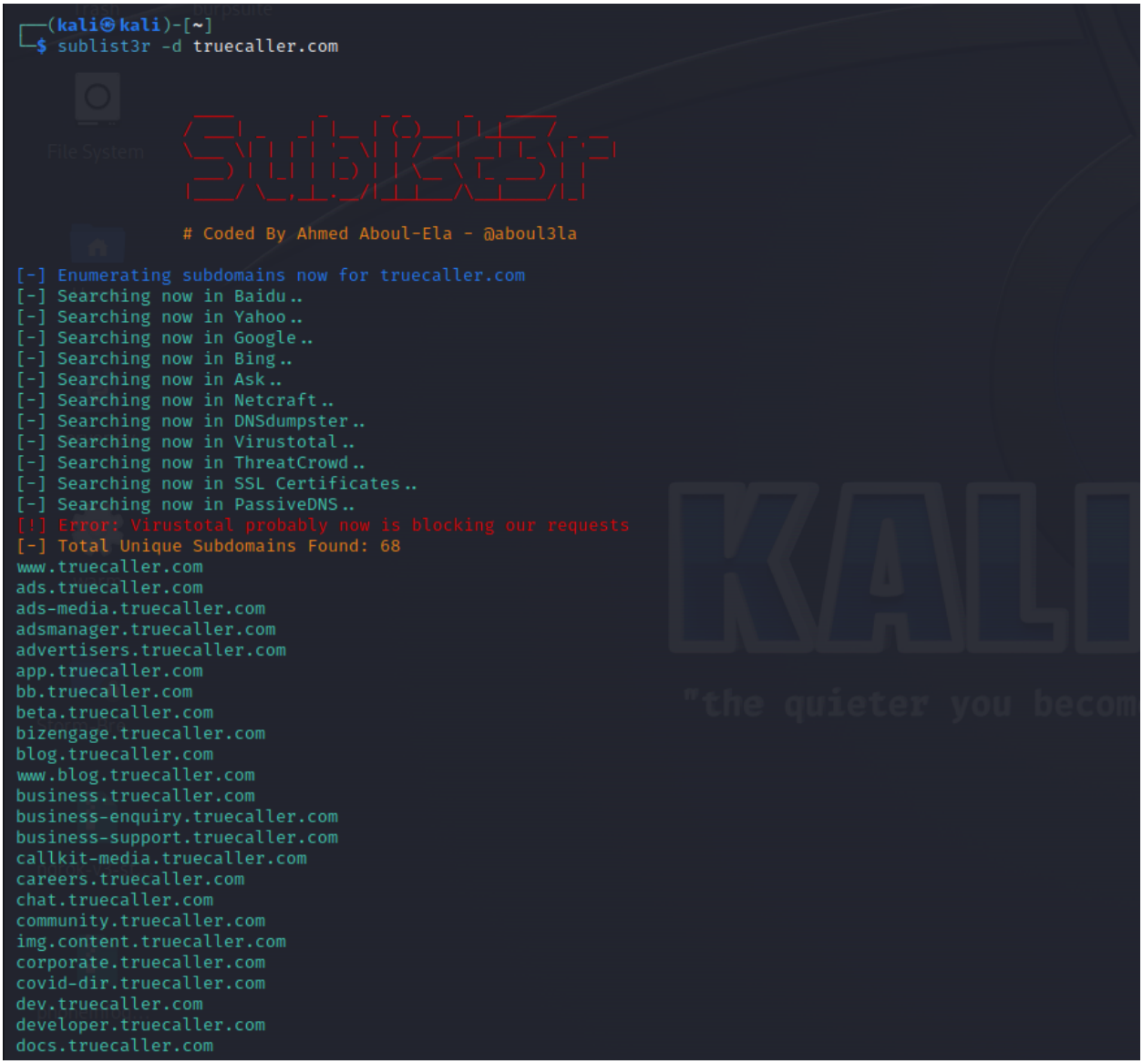



## Host



## Whatweb


## Sublist3r



Using nmap and scan the website vulnerabilities.

```
┌──(kali㉿kali)-[~]
└─$ nmap -Pn 199.36.158.100

Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-28 08:32 EDT
Nmap scan report for 199.36.158.100
Host is up (0.036s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE
53/tcp  open  domain
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 9.66 seconds
```

```
┌──(root㉿kali)-[/home/kali]
└─# nmap --script http-security-headers -p 443 truecaller.com
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-28 08:37 EDT
Nmap scan report for truecaller.com (3.108.188.61)
Host is up (0.022s latency).
Other addresses for truecaller.com (not scanned): 3.7.154.113
rDNS record for 3.108.188.61: ec2-3-108-188-61.ap-south-1.compute.amazonaws.com

PORT    STATE SERVICE
443/tcp open  https
| http-security-headers:
|   Strict_Transport_Security:
|     Header: Strict-Transport-Security: max-age=15552000; includeSubDomains
|   X_Frame_Options:
|     Header: X-Frame-Options: SAMEORIGIN
|     Description: The browser must not display this content in any frame from a page of different origin than the content itself.
|   X_XSS_Protection:
|     Header: X-XSS-Protection: 1; mode=block
|     Description: The browser will prevent the rendering of the page when XSS is detected.
|   X_Content_Type_Options:
|     Header: X-Content-Type-Options: nosniff
|     Description: Will prevent the browser from MIME-sniffing a response away from the declared content-type.
|   Cache_Control:
|_    Header: Cache-Control: s-maxage=60

Nmap done: 1 IP address (1 host up) scanned in 2.01 seconds
```

# OWASP ZAP Report

## Alert counts by site and risk

This table shows, for each site for which one or more alerts were raised, the number of alerts raised at each risk level.

Alerts with a confidence level of "False Positive" have been excluded from these counts.

(The numbers in brackets are the number of alerts raised for the site at or above that risk level.)

| | | Risk | | | |
|---|---|---|---|---|---|
| | | High (= High) | Medium (>= Medium) | Low (>= Low) | Informational (>= Informational) |
| Site | https://chat.zipzip.ai | 0 (0) | 1 (1) | 0 (1) | 0 (1) |
| | https://www.truecaller.com | 0 (0) | 3 (3) | 5 (8) | 5 (13) |

## Strict-Transport-Security Header Not Set

| | |
|---|---|
| **Source** | raised by a passive scanner (Strict-Transport-Security Header) |
| **CWE ID** | 319 |
| **WASC ID** | 15 |
| **Reference** | ▪ https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html<br>▪ https://owasp.org/www-community/Security_Headers<br>▪ https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security<br>▪ https://caniuse.com/stricttransportsecurity<br>▪ https://datatracker.ietf.org/doc/html/rfc6797 |

**Proposed mitigation or fix**

Using the HTTP Strict Transport Security (HSTS) header on all of your web servers will improve security and mitigate the "Strict-Transport-Security Header Not Set" issue. In order to accomplish this, you must first make sure that your website completely supports HTTPS and is free of mixed content concerns, which means that every element should load over HTTPS. You set up your web server to incorporate the HSTS header in its HTTP responses after setting up HTTPS successfully.

Test your website after adding the HSTS header to make sure all resources load properly via HTTPS and that there are no mistakes in the HTTPS implementation. To guarantee ongoing adherence to advised security procedures, periodically check and maintain the security settings on your server. This proactive strategy enhances your website's overall security posture and helps protect user data.

# 10.Path traversal

**What is path traversal?**

A security flaw called path traversal, sometimes referred to as directory traversal, arises when an application allows user input to be utilized to access files or directories that are kept outside of the designated restricted directories. An attacker can gain access to files and directories on a web server's file system, including configuration files, application source code, and important system files, by taking advantage of a path traversal vulnerability. This could result in the disclosure of confidential information, data manipulation, or remote code execution.

Typically, path traversal exploits entail modifying variables that point to files containing "dot-dot-slash (../)" sequences or comparable techniques. These steps are used to access files or directories that shouldn't be available through the web application by moving up in the directory structure (parent directories). An attacker could alter a query parameter, for instance, by adding "../" sequences to access parent directories if the application utilizes them to directly reference filenames inside a restricted directory.

**Affected components:**

Path traversal vulnerabilities can impact multiple elements within the ecosystem of a web application, underscoring the significance of a secure and resilient system architecture. Since files and directories are accessed and altered on the web server, it is the main component that is impacted. In the event that a path traversal attack is successful, sensitive files, configuration information, and system data may be exposed, opening the door to more intrusions.

Security protocols and application logic are also severely harmed. These attacks can be launched against applications that do a poor job of handling user inputs or file requests, particularly when sanitization and path validation methods are either

nonexistent or poorly implemented. This vulnerability frequently results from coding errors where users' input is utilized directly to access file systems without the necessary security checks.

If important files like scripts, configuration files, or databases are accessed or altered, the security infrastructure—which includes authentication methods and access control systems—may be compromised. If the attacker edits important system files or obtains sensitive data, this could result in increased privileges inside the program or possibly complete system access.

**Impact assessment:**

An attack on the system's security that results from arbitrary code being run. a damaged reputation in addition to a betrayed trust. Noncompliance with regulatory norms in breach. After carrying out an impact assessment, a company can adopt suitable security measures and prioritize repair actions, thereby gaining a better knowledge of the potential risks and consequences linked to a route traversal vulnerability.

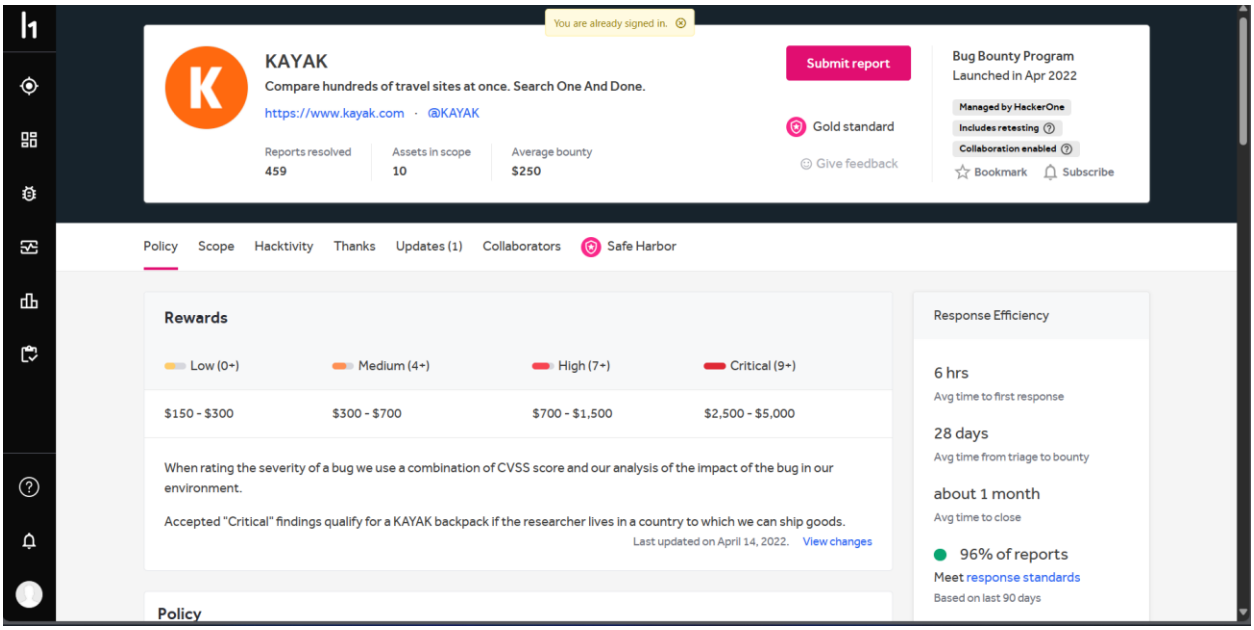
**Steps to reproduce:**

**1.Identify input vectors:** Start by listing every location within the application where files are accessed by user input. This may be accomplished by form inputs, URL parameters, or any API endpoint that performs file access functions.

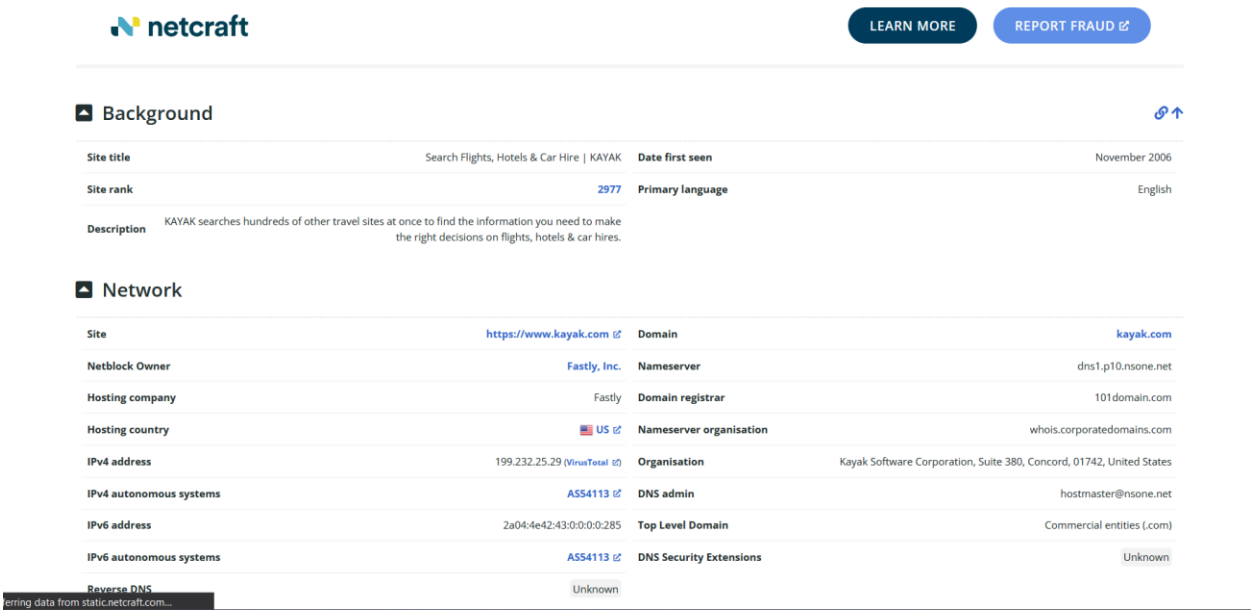
**2.Test inputs:** Enter these sequences into the file request features of the application. For instance, if a filename is specified via a parameter in a URL, change it by adding path traversal sequences before the filename.

**3.Verify the server configuration:** It's crucial to examine the server's error logs and response headers in order to learn more about the file system and server configuration, which may provide more light on weak points.

## Proof of concepts:

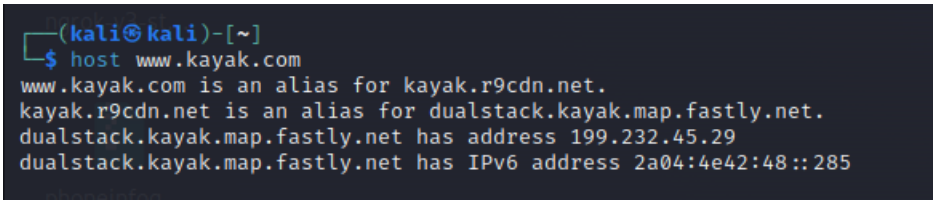


Using **netcraft** find the informations:




**host** and **nslookup** tools are programs that collect data from the Domain Name System (DNS) using command lines. The DNS assists in converting human-readable domain names into IP addresses that computers use to identify one another on a network.

## Host



## Nslookup

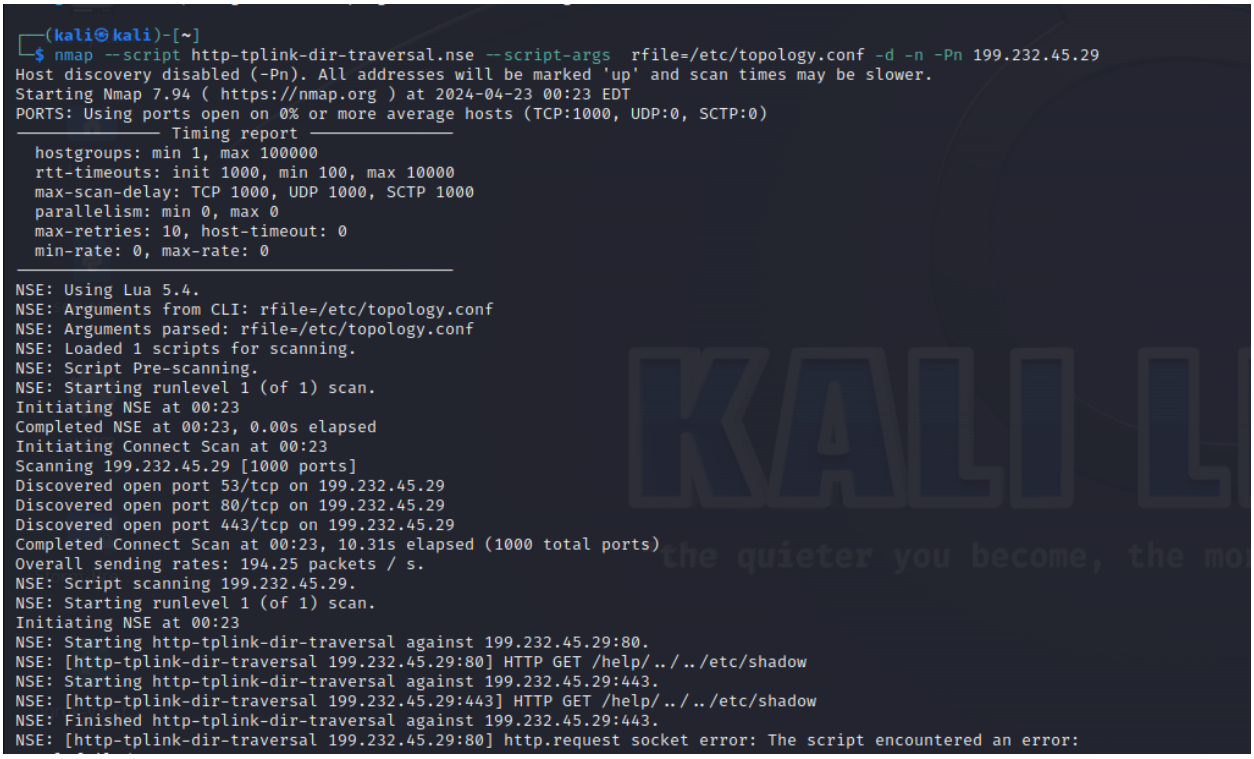
```
┌──(kali㉿kali)-[~]
└─$ nslookup www.kayak.com
Server:         192.168.43.1
Address:        192.168.43.1#53

Non-authoritative answer:
www.kayak.com   canonical name = kayak.r9cdn.net.
kayak.r9cdn.net canonical name = dualstack.kayak.map.fastly.net.
Name:   dualstack.kayak.map.fastly.net
Address: 199.232.45.29
Name:   dualstack.kayak.map.fastly.net
Address: 2a04:4e42:48::285
```

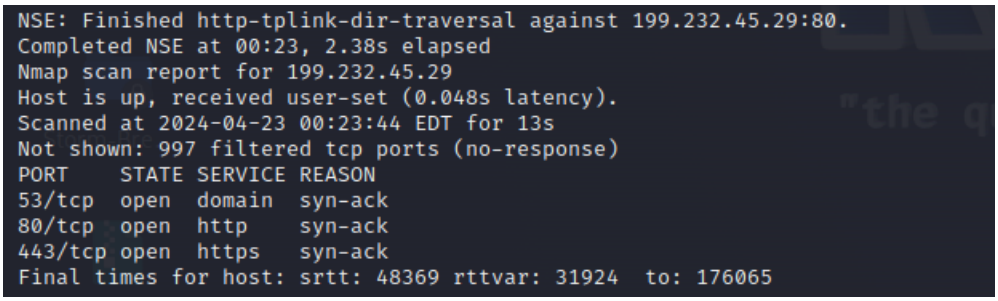
## Whatweb

## Sublist3r

Checking the sub domains under that domain.

Scan vulnerability using **nmap**,



# Owasp zap report

### Alert counts by site and risk

This table shows, for each site for which one or more alerts were raised, the number of alerts raised at each risk level.

Alerts with a confidence level of "False Positive" have been excluded from these counts.

(The numbers in brackets are the number of alerts raised for the site at or above that risk level.)

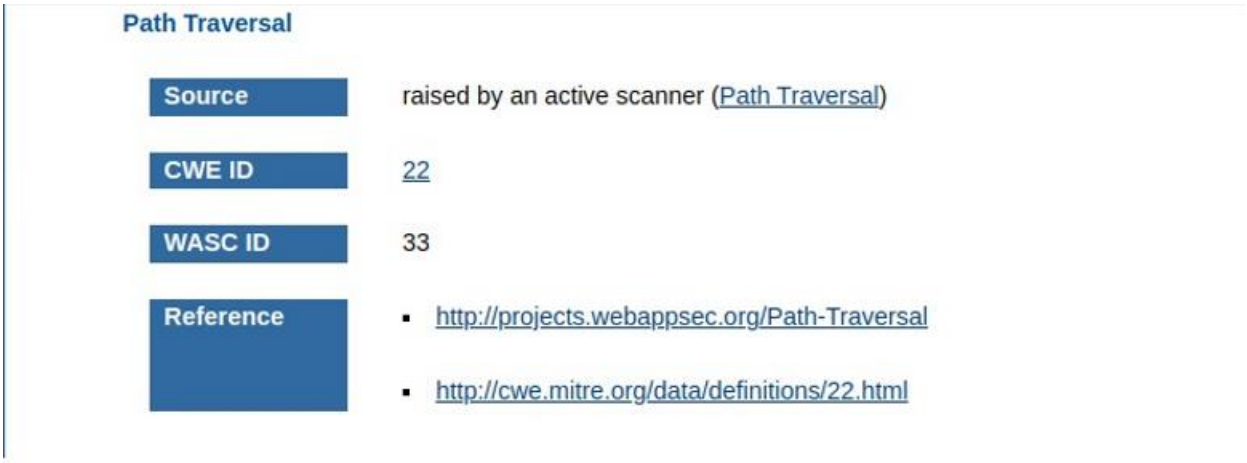
| | | Risk | | | |
|---|---|---|---|---|---|
| | | High (= High) | Medium (>= Medium) | Low (>= Low) | Informational (>= Informational) |
| Site | https://business.kayak.com | 1 (1) | 0 (1) | 0 (1) | 0 (1) |

### Alert counts by alert type

This table shows the number of alerts of each alert type, together with the alert type's risk level.

(The percentages in brackets represent each count as a percentage, rounded to one decimal place, of the total number of alerts included in this report.)

| Alert type | Risk | Count |
|---|---|---|
| Path Traversal | High | 33 (157.1%) |




**Proposed mitigation or fix**

**1.Input validation:** User input should always be verified and sanitized to make sure it doesn't contain any dangerous characters or patterns. Accept only inputs that are expected and validated. Disallow characters such as "\" or "../".

**2.Use absolute path:** Make that all user-input paths are absolute paths and that they are located in the desired directory.

**3.Use secure APIs:** Use higher-level APIs wherever you can, as they are built to manage route traversal safely and automatically.

**4.Least privilege:** Make sure the programs on the web server are running with the least number of permissions required. By doing this, the effect of a possible path traversal attack is reduced.

# 5.References

[1] wikipedia, "Cross-origin resource sharing," wikipedia, 8 4 2024. [Online]. Available: https://en.wikipedia.org/wiki/Cross-origin_resource_sharing. [Accessed 20 4 2024].

[2] wikipedia, "Content Security Policy," wikipedia, 26 1 2024. [Online]. Available: https://en.wikipedia.org/wiki/Content_Security_Policy. [Accessed 20 4 2024].

[3] imperva, "Content Security Policy (CSP)," imperva, 20 1 2023. [Online]. Available: https://www.imperva.com/learn/application-security/content-security-policy-csp-header/. [Accessed 20 4 2024].

[4] KirstenS, "owasp," owasp, [Online]. Available: https://owasp.org/www-community/attacks/csrf#. [Accessed 21 4 2024].