Sri Lanka Institute of Information Technology

IE2062-Web Security

IT Number	Name
IT22581402	C.D.Aluthge

Information gathering and reconnaissance phase

a. Subdomain enumeration
   i. Recon-ng

b. Getting alive subdomains
   i. Nslookup

c. DNS enumeration
   i. Dnsrecon

d. Public devices enumeration
   i. Censys

e. Find WAF (web application firewall) protection.
   i. Wafwoof

f. Find open ports.
   i. Nmap

g. Exploitation
   i. sqlmap

vulnerability analysis phase

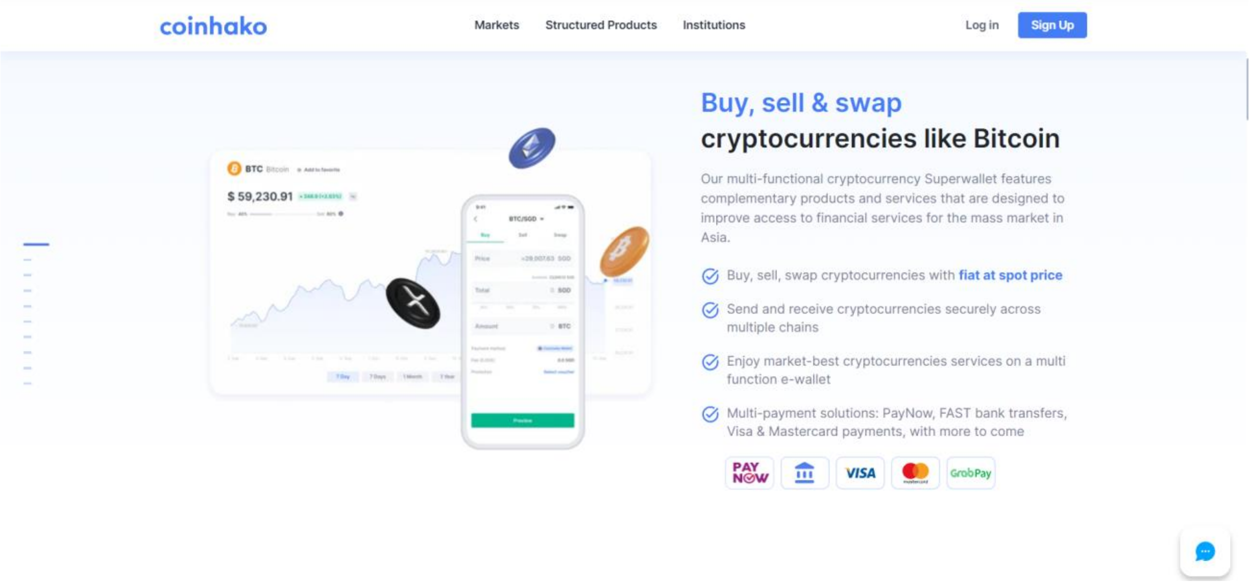
1. Target domain: https://www.coinhako.com

    a. Weak Ciphers Enabled (Confirmed)
    b. [Possible] Phishing by Navigating Browser Tabs
    c. Web Application Firewall Detected
    d. Unexpected Redirect Response Body (Too Large)
    e. Forbidden Resource
    f. Missing X-XSS-Protection Header

Conclusion

Scope:

Coinhako is a renowned crypto exchange to buy cryptocurrencies in Singapore, and Asia. Trade crypto, manage your crypto wallet and view cryptocurrency.

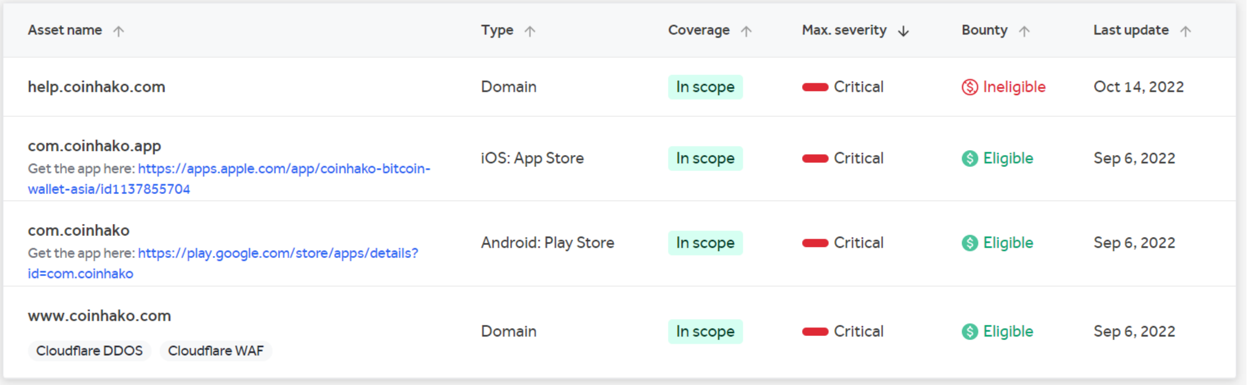


Buy, sell & swap cryptocurrencies like Bitcoin

Our multi-functional cryptocurrency Superwallet features complementary products and services that are designed to improve access to financial services for the mass market in Asia.

- Buy, sell, swap cryptocurrencies with **fiat at spot price**
- Send and receive cryptocurrencies securely across multiple chains
- Enjoy market-best cryptocurrencies services on a multi function e-wallet
- Multi-payment solutions: PayNow, FAST bank transfers, Visa & Mastercard payments, with more to come

In Scope:



Asset name	Type	Coverage	Max. severity	Bounty	Last update
help.coinhako.com	Domain	In scope	Critical	Ineligible	Oct 14, 2022
com.coinhako.app Get the app here: https://apps.apple.com/app/coinhako-bitcoin-wallet-asia/id1137855704	iOS: App Store	In scope	Critical	Eligible	Sep 6, 2022
com.coinhako Get the app here: https://play.google.com/store/apps/details?id=com.coinhako	Android: Play Store	In scope	Critical	Eligible	Sep 6, 2022
www.coinhako.com Cloudflare DDOS Cloudflare WAF	Domain	In scope	Critical	Eligible	Sep 6, 2022

OWASP top 10 vulnerabilities

- Broken Access Control
- Cryptographic Failures
- Injection

- Insecure Design
- Security Misconfiguration
- Vulnerable and Outdated Components
- Identification and Authentication Failures
- Software and Data Integrity Failures
- Security Logging and Monitoring Failures
- Server-Side Request Forgery

Broken Access Control

When users are unable to adequately impose limitations on what they can access or do within a system, this is known as broken access control. This vulnerability makes it possible for unauthorized individuals to access private data or carry out tasks that shouldn't be permitted. Insecure references to objects, improperly configured permissions, and a lack of authentication checks are the causes of it. Because it can result in illegal data exposure, manipulation, or system penetration, it's a major problem.

Cryptographic Failures

Cryptographic collapses are the result of poor algorithms, incorrect implementations, or improper management of encryption keys, which compromise the security of information encrypted via encryption techniques.

Undermining security safeguards and perhaps leading in data breaches, this can lead to unauthorized access to or exposure of sensitive data.

Injection

Injection: it is a type of security vulnerability in which untrusted data is sent to an interpreter as part of a command or query. Most often, attackers use this to inject malicious code into input fields, for example, login forms or search boxes. The interpreter then executes this code, which can give the attacker unauthorized access to data, alter the application's behavior, or gain full control over the system. There are many types of injections, such as SQL injection, where the attackers manipulate database queries, or cross-site scripting, where the malicious script is injected into a web page that other users view.

Insecure Design

Insecure design means the security flaws within the fundamental system or application design born out of insufficient attention to security at the design time. Consequently, systems have vulnerabilities that enable unauthorized system access or breaches. Examples are poor user authentication or no network segmentation. To fix insecure design, significant changes to the architecture are needed to develop the designs securely.

Security Misconfiguration

Imagine you accidentally leave your front door closed but not locked. That's what a security misconfiguration is like. Essentially, it's not properly securing your software systems, servers, or web applications. This can include default settings, shipped insecurely, or unnecessary features. Unauthorized users might target such vulnerabilities to access your info or disrupt the operation.

Therefore, one should monitor and update security measures continuously to avoid misconfiguration.

Vulnerable and Outdated Components

Vulnerable components are software elements that are known to contain security flaws that could be exploited by hackers. However, software components that have not received an upgrade in a long time are known as outdated components, and they may not include important security updates and bug fixes. In order to maintain a secure system environment, both sorts of components present security concerns and should be routinely updated and monitored.

Identification and Authentication Failures

When systems are unable to accurately confirm user identities or authenticate their access credentials, identification and authentication failures take place. Whereas authentication failures are caused by an inability to verify user identities, identification failures are the consequence of incorrect user recognition. In order to identify and fix system vulnerabilities, strong authentication procedures and frequent security assessments are essential. These failures might result in unauthorized access and security breaches.

Software and Data Integrity Failures

Software code or data accuracy, consistency, and reliability are jeopardized in software and data integrity issues. Data integrity problems are caused by mistakes, unauthorized access, or cyberattacks, whereas programming errors, malicious code, or unauthorized alterations are the causes of software integrity problems. To reduce risks and guarantee system security and dependability, preventive measures include backups, data encryption, access controls, and regular updates.

Security Logging and Monitoring Failures

Failed security logging and monitoring systems result in a delayed detection of security incidents and higher risks since they are unable to reliably capture and evaluate security-related events. Monitoring failures are caused by insufficient tools or response protocols, whereas logging failures are caused by misconfigurations or deactivated logging systems. Organizations should establish strong response protocols, use comprehensive logging practices, deploy efficient monitoring tools like SIEM tools, train security teams, and continuously improve logging and monitoring configurations in light of best practices and emerging threats in order to address these failures. By taking these steps, cybersecurity defenses are strengthened and the effects of security events are lessened.

Server-Side Request Forgery

A security flaw known as server-side request forgery (SSRF) allows attackers to take control of a susceptible server and send unwanted requests, usually to external or internal systems. Unauthorized access, data loss, and more exploitation may result from this. Input validation, whitelisting authorized resources, network segmentation, and access controls are all part of prevention. Frequent security testing improves overall system security by assisting in the identification and mitigation of SSRF threats.

Information gathering and reconnaissance phase.

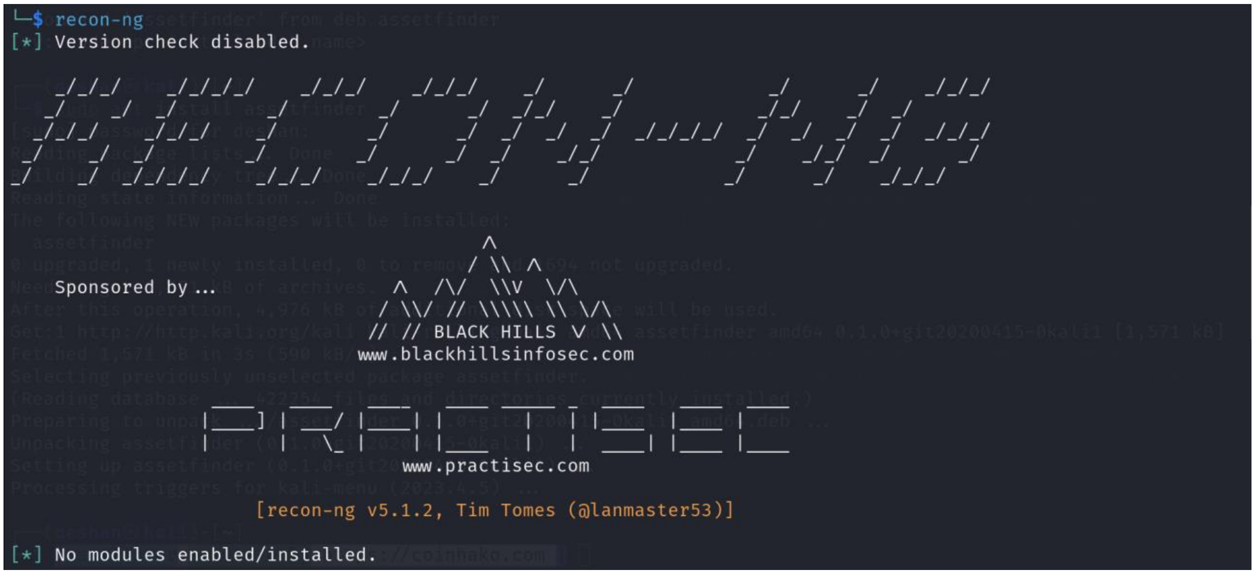
The objective is to gather as much pertinent data as you can about the intended system or organization. Understanding the target's infrastructure, seeing potential vulnerabilities, and organizing additional security testing operations are all made easier with the aid of this information.

Domain: https://coinhako.com

- **Recon-ng**

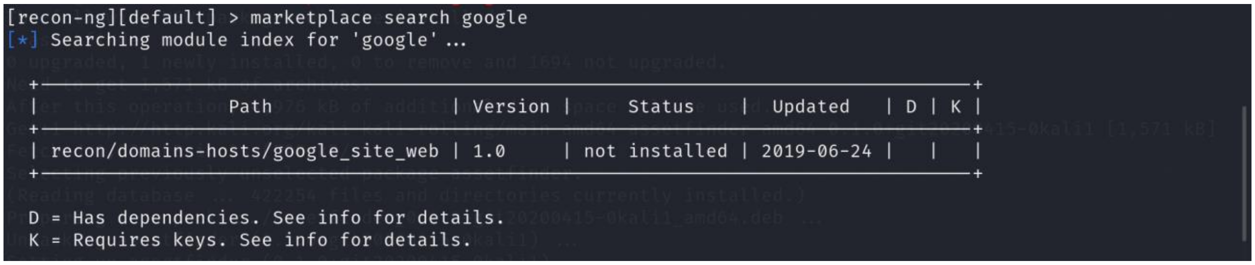
Recon-ng is an open-source reconnaissance tool with Python foundation that is used for penetration testing and security evaluations. It collects data from domains, IPs, emails, and social media platforms using a modular framework with different modules. The advantages of recon-ng are its flexibility for scripting and automation, integration with many data sources, and data enrichment capabilities. It is used by security experts to gather and evaluate intelligence, spot possible weaknesses, and improve overall security posture during the first reconnaissance phase. Its vibrant community guarantees continuous upgrades and support, which makes it an invaluable addition to cybersecurity toolkits.

Proof of concept:



```
└─$ recon-ng
[*] Version check disabled.

    _/_/_/    _/_/_/_/    _/_/_/    _/_/_/    _/      _/            _/      _/    _/_/_/
   _/    _/  _/        _/        _/      _/  _/_/    _/            _/_/    _/  _/
  _/_/_/    _/_/_/    _/        _/      _/  _/  _/  _/  _/_/_/_/  _/  _/  _/  _/  _/_/_/
 _/    _/  _/        _/        _/      _/  _/    _/_/            _/    _/_/  _/      _/
_/    _/  _/_/_/_/    _/_/_/    _/_/_/    _/      _/            _/      _/    _/_/_/

                                          /\
                                         / \\ /\
    Sponsored by...               /\  /\/  \\V  \/\
                                 / \\/ // \\\\\ \\ \/\
                                // // BLACK HILLS \/ \\
                               www.blackhillsinfosec.com

                  ____   ____   ____   ____ _____ _  ____   ____  ____
                 |____] | ___/ |____| |       |   | |____  |____ |
                 |      |   \_ |    | |____   |   |  ____| |____ |____
                                   www.practisec.com

                      [recon-ng v5.1.2, Tim Tomes (@lanmaster53)]

[*] No modules enabled/installed.
```

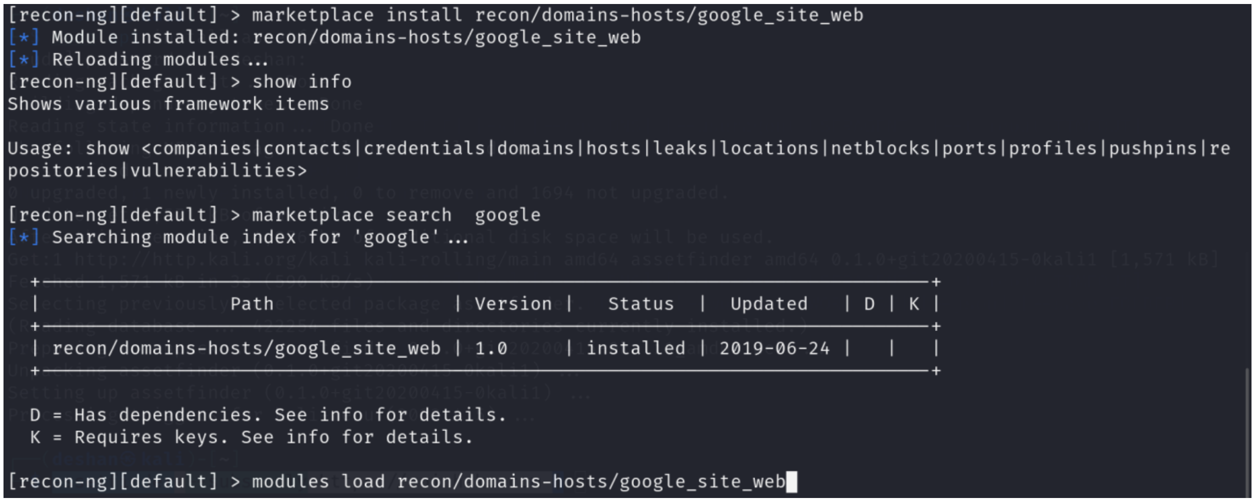
To get google website give this command.



```
[recon-ng][default] > marketplace search google
[*] Searching module index for 'google'...

  +----------------------------------------------------------------------------------+
  |              Path              | Version |     Status    |  Updated   | D | K |
  +----------------------------------------------------------------------------------+
  | recon/domains-hosts/google_site_web | 1.0     | not installed | 2019-06-24 |   |   |
  +----------------------------------------------------------------------------------+

  D = Has dependencies. See info for details.
  K = Requires keys. See info for details.
```

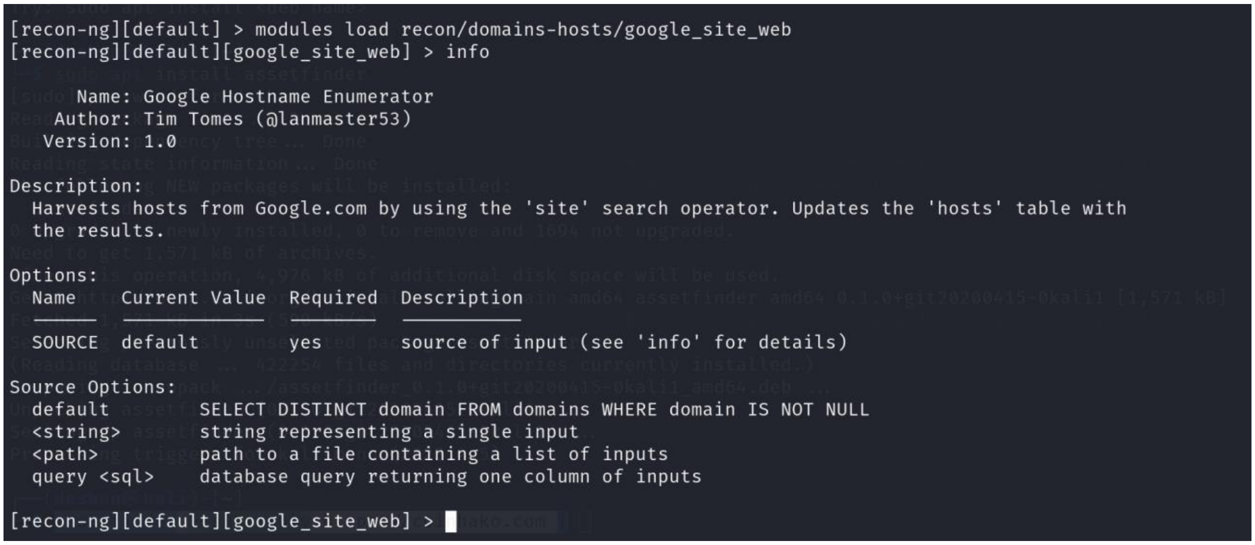
You can see it’s not installed yet. We must download installation path.

After installing path using show info to see its download or not.



```
[recon-ng][default] > marketplace install recon/domains-hosts/google_site_web
[*] Module installed: recon/domains-hosts/google_site_web
[*] Reloading modules ...
[recon-ng][default] > show info
Shows various framework items

Usage: show <companies|contacts|credentials|domains|hosts|leaks|locations|netblocks|ports|profiles|pushpins|re
positories|vulnerabilities>

[recon-ng][default] > marketplace search  google
[*] Searching module index for 'google' ...

  +-----------------------------------------------------------------------------------------+
  |                Path                | Version |   Status  |  Updated   | D | K |
  +-----------------------------------------------------------------------------------------+
  | recon/domains-hosts/google_site_web | 1.0     | installed | 2019-06-24 |   |   |
  +-----------------------------------------------------------------------------------------+

  D = Has dependencies. See info for details.
  K = Requires keys. See info for details.

[recon-ng][default] > modules load recon/domains-hosts/google_site_web
```

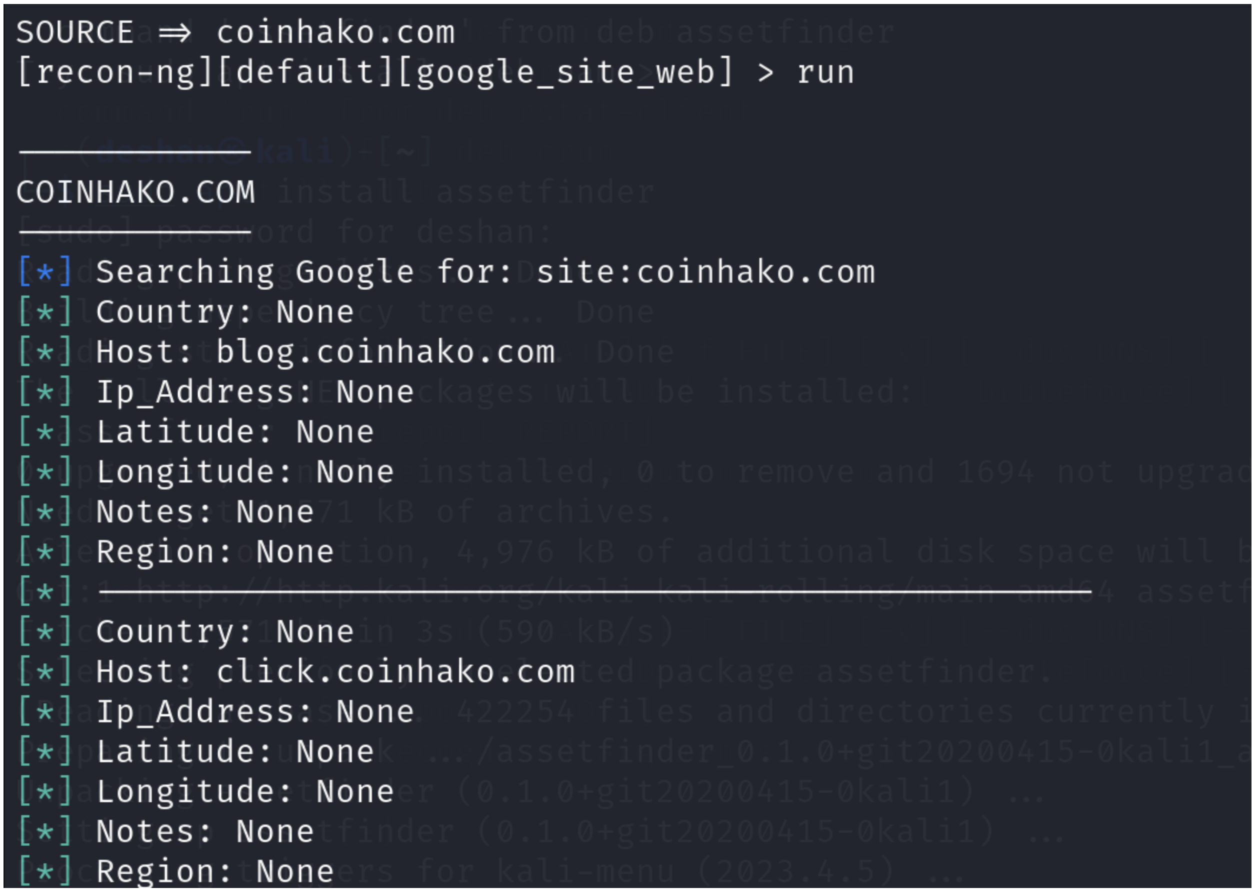
Load the installed module path and use info see options.



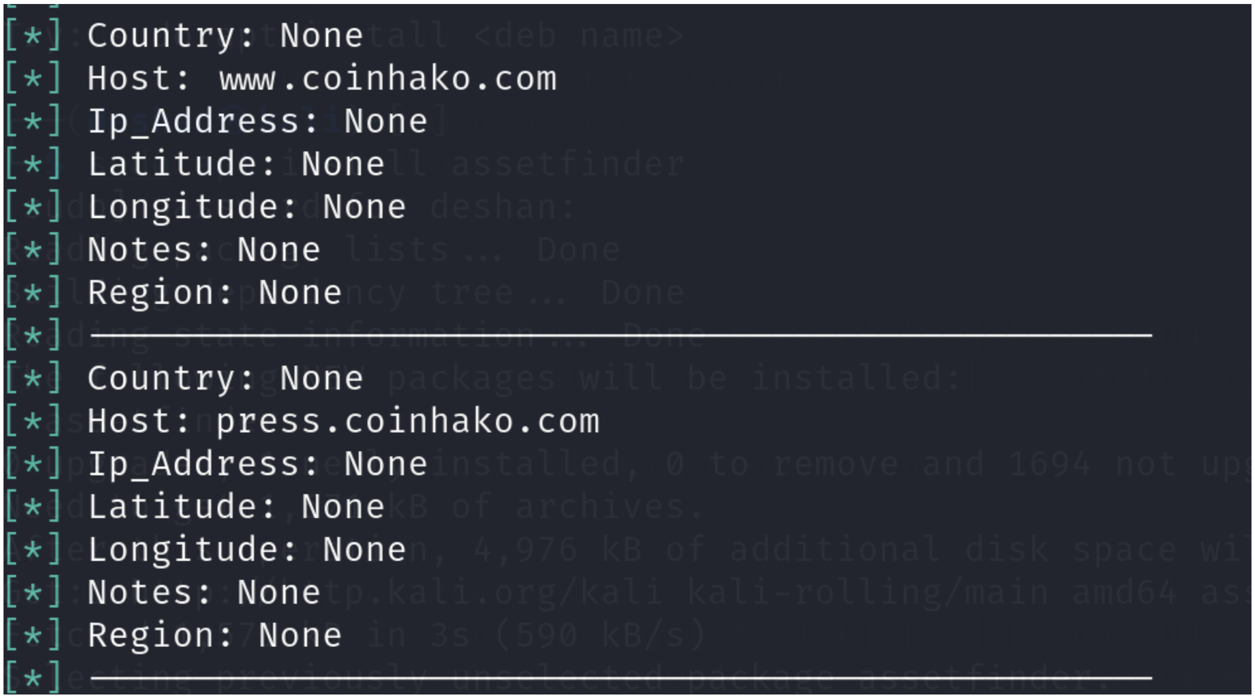
```
[recon-ng][default] > modules load recon/domains-hosts/google_site_web
[recon-ng][default][google_site_web] > info

      Name: Google Hostname Enumerator
    Author: Tim Tomes (@lanmaster53)
   Version: 1.0

Description:
  Harvests hosts from Google.com by using the 'site' search operator. Updates the 'hosts' table with
  the results.

Options:
  Name    Current Value  Required  Description
  ------  -------------  --------  -----------
  SOURCE  default        yes       source of input (see 'info' for details)

Source Options:
  default        SELECT DISTINCT domain FROM domains WHERE domain IS NOT NULL
  <string>       string representing a single input
  <path>         path to a file containing a list of inputs
  query <sql>    database query returning one column of inputs

[recon-ng][default][google_site_web] >
```

Go to options and set source to our targeted domain www.coinhako.com

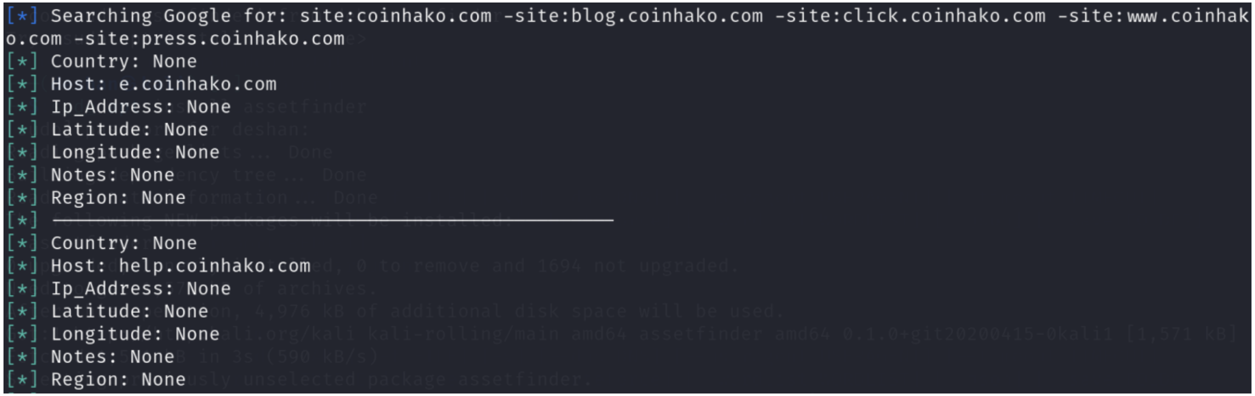
and run it.



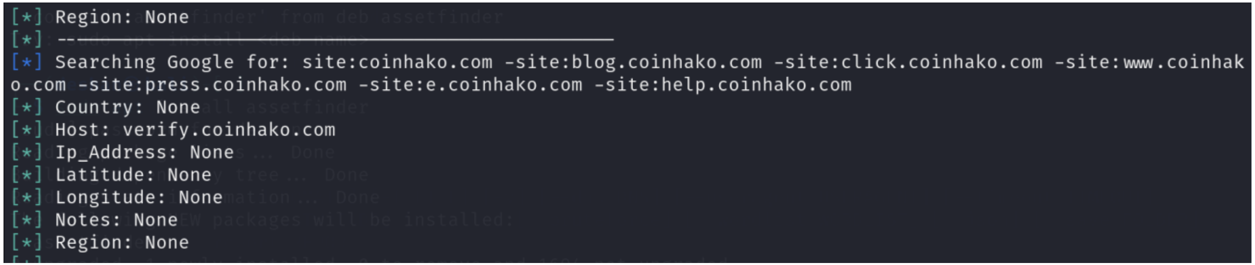
```
SOURCE ⇒ coinhako.com
[recon-ng][default][google_site_web] > run

────────────
COINHAKO.COM
────────────
[*] Searching Google for: site:coinhako.com
[*] Country: None
[*] Host: blog.coinhako.com
[*] Ip_Address: None
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ──────────────────────────────────────────────────
[*] Country: None
[*] Host: click.coinhako.com
[*] Ip_Address: None
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
```



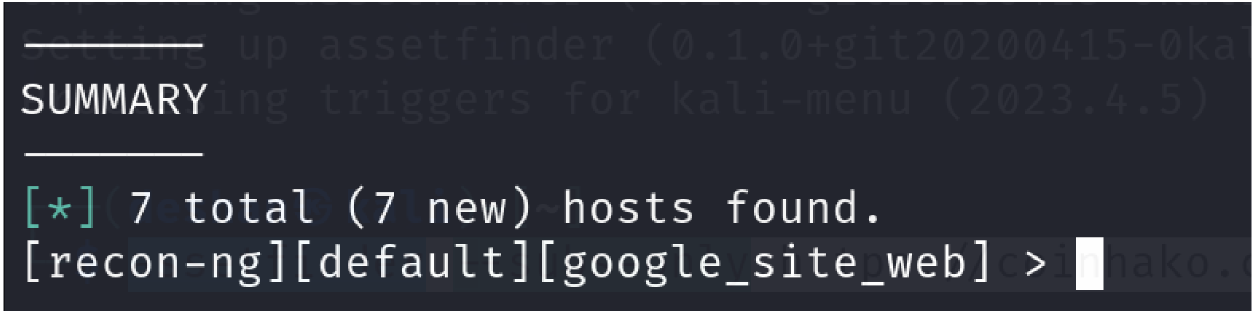
```
[*] Country: None
[*] Host: www.coinhako.com
[*] Ip_Address: None
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] -------------------------------------------------
[*] Country: None
[*] Host: press.coinhako.com
[*] Ip_Address: None
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] -------------------------------------------------
```



```
[*] Searching Google for: site:coinhako.com -site:blog.coinhako.com -site:click.coinhako.com -site:www.coinhak
o.com -site:press.coinhako.com
[*] Country: None
[*] Host: e.coinhako.com
[*] Ip_Address: None
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] -------------------------------------------------
[*] Country: None
[*] Host: help.coinhako.com
[*] Ip_Address: None
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
```



```
[*] Region: None
[*] -------------------------------------------------
[*] Searching Google for: site:coinhako.com -site:blog.coinhako.com -site:click.coinhako.com -site:www.coinhak
o.com -site:press.coinhako.com -site:e.coinhako.com -site:help.coinhako.com
[*] Country: None
[*] Host: verify.coinhako.com
[*] Ip_Address: None
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
```



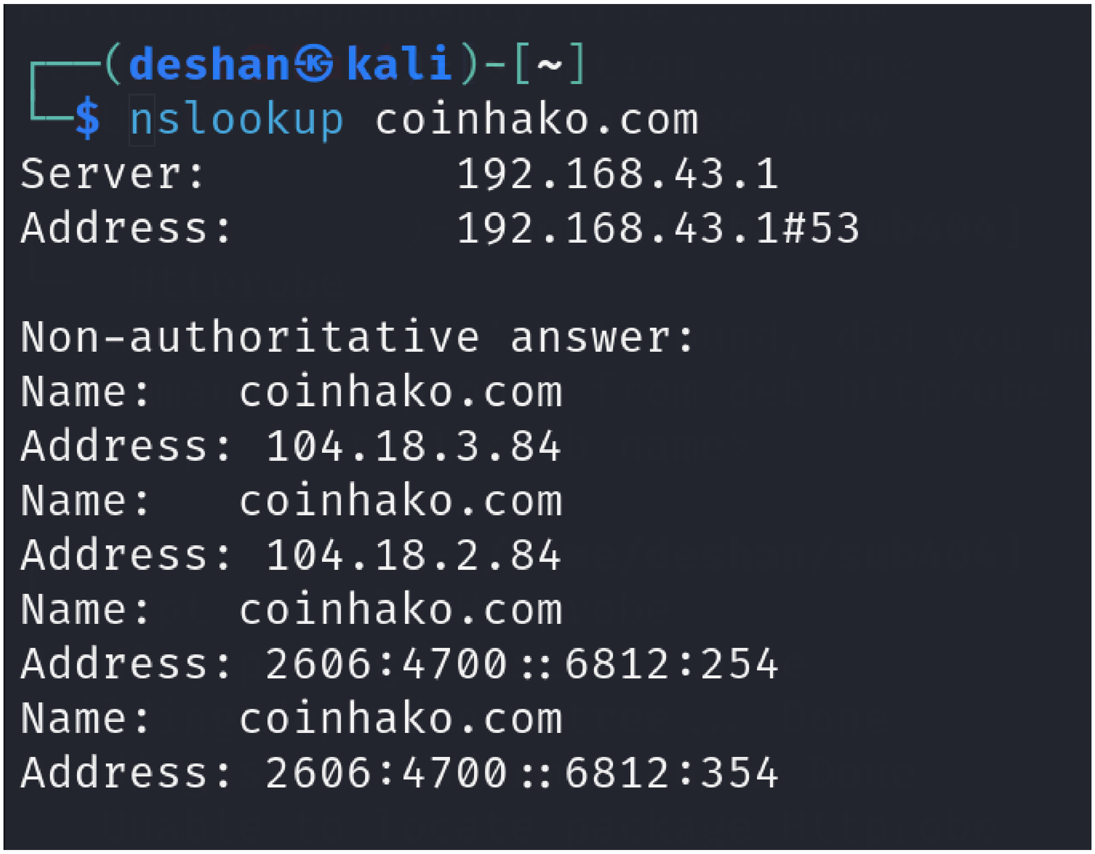
```
-------
SUMMARY
-------
[*] 7 total (7 new) hosts found.
[recon-ng][default][google_site_web] >
```

Getting alive subdomains

- Nslookup

A command-line utility called `nslookup` is used to query DNS (Domain Name System) servers and get DNS-related data. In addition to performing reverse DNS lookups and retrieving other DNS records like A, AAAA, MX, NS, PTR, and TXT records, it resolves domain names to IP addresses. It is flexible for diagnosing DNS problems, validating DNS configurations, and testing DNS servers because it may be used in both interactive and non-interactive modes. Network administrators and cybersecurity experts frequently utilize `nslookup`, which runs on several platforms, for DNS-related activities and diagnostics.

Proof of concept:



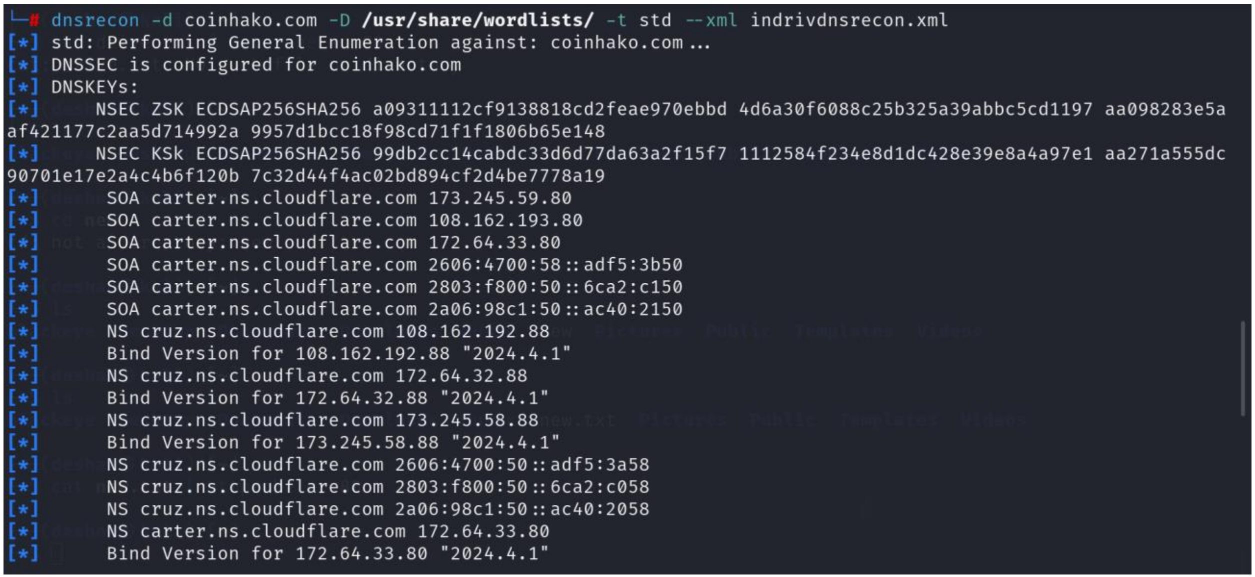
```
┌──(deshan㉿kali)-[~]
└─$ nslookup coinhako.com
Server:         192.168.43.1
Address:        192.168.43.1#53

Non-authoritative answer:
Name:   coinhako.com
Address: 104.18.3.84
Name:   coinhako.com
Address: 104.18.2.84
Name:   coinhako.com
Address: 2606:4700::6812:254
Name:   coinhako.com
Address: 2606:4700::6812:354
```

DNS enumeration

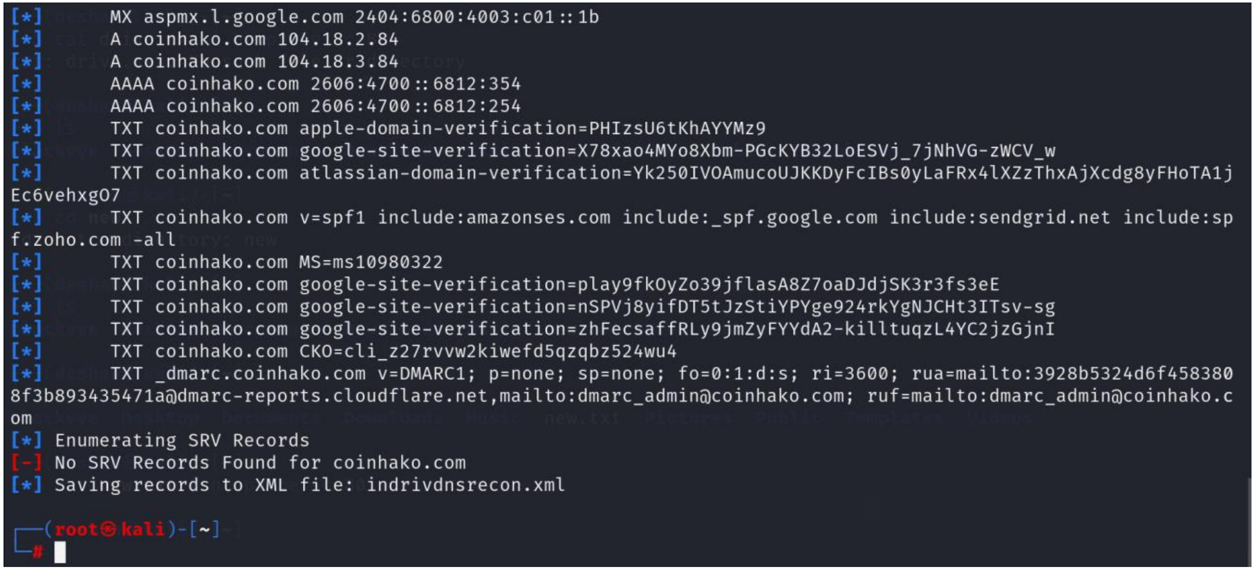
- **DNSrecon**

For security evaluations and penetration tests, DNSRecon is a powerful open-source tool for DNS reconnaissance that finds and counts DNS information. It is particularly good at finding subdomains, enumerating DNS records (A, AAAA, MX, NS, SOA, TXT), transferring DNS zones, and it can handle wordlist-based and brute-force attacks. With its automation, customization, and integration features, it's a useful tool for figuring out possible attack points, comprehending target infrastructure, and raising security awareness in general. DNSRecon is still a dependable option for DNS reconnaissance operations because of its community support and frequent updates.

Proof of concept:



```
└─# dnsrecon -d coinhako.com -D /usr/share/wordlists/ -t std --xml indrivdnsrecon.xml
[*] std: Performing General Enumeration against: coinhako.com ...
[*] DNSSEC is configured for coinhako.com
[*] DNSKEYs:
[*]     NSEC ZSK ECDSAP256SHA256 a09311112cf9138818cd2feae970ebbd 4d6a30f6088c25b325a39abbc5cd1197 aa098283e5a
af421177c2aa5d714992a 9957d1bcc18f98cd71f1f1806b65e148
[*]     NSEC KSk ECDSAP256SHA256 99db2cc14cabdc33d6d77da63a2f15f7 1112584f234e8d1dc428e39e8a4a97e1 aa271a555dc
90701e17e2a4c4b6f120b 7c32d44f4ac02bd894cf2d4be7778a19
[*]      SOA carter.ns.cloudflare.com 173.245.59.80
[*]      SOA carter.ns.cloudflare.com 108.162.193.80
[*]      SOA carter.ns.cloudflare.com 172.64.33.80
[*]      SOA carter.ns.cloudflare.com 2606:4700:58::adf5:3b50
[*]      SOA carter.ns.cloudflare.com 2803:f800:50::6ca2:c150
[*]      SOA carter.ns.cloudflare.com 2a06:98c1:50::ac40:2150
[*]      NS cruz.ns.cloudflare.com 108.162.192.88
[*]      Bind Version for 108.162.192.88 "2024.4.1"
[*]      NS cruz.ns.cloudflare.com 172.64.32.88
[*]      Bind Version for 172.64.32.88 "2024.4.1"
[*]      NS cruz.ns.cloudflare.com 173.245.58.88
[*]      Bind Version for 173.245.58.88 "2024.4.1"
[*]      NS cruz.ns.cloudflare.com 2606:4700:50::adf5:3a58
[*]      NS cruz.ns.cloudflare.com 2803:f800:50::6ca2:c058
[*]      NS cruz.ns.cloudflare.com 2a06:98c1:50::ac40:2058
[*]      NS carter.ns.cloudflare.com 172.64.33.80
[*]      Bind Version for 172.64.33.80 "2024.4.1"
```



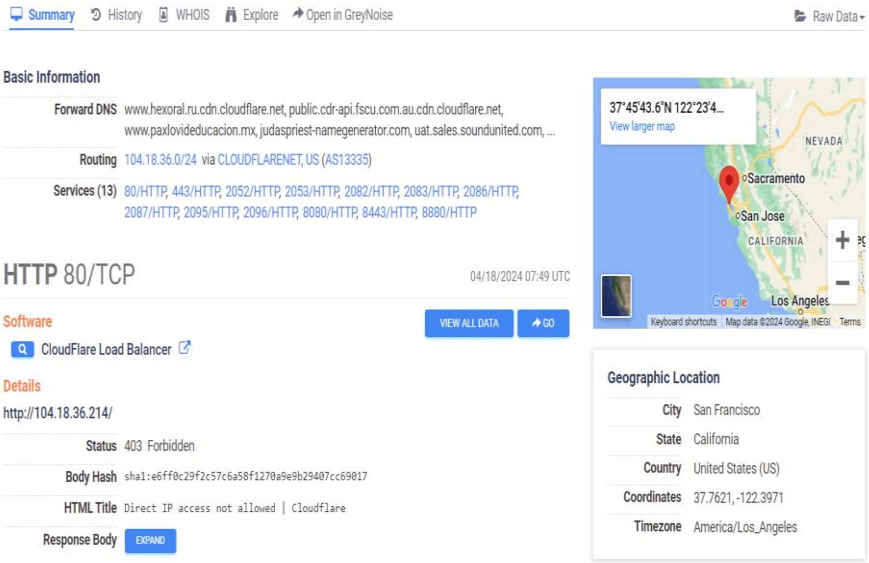
```
[*]      MX aspmx.l.google.com 2404:6800:4003:c01::1b
[*]      A coinhako.com 104.18.2.84
[*]      A coinhako.com 104.18.3.84
[*]      AAAA coinhako.com 2606:4700::6812:354
[*]      AAAA coinhako.com 2606:4700::6812:254
[*]      TXT coinhako.com apple-domain-verification=PHIzsU6tKhAYYMz9
[*]      TXT coinhako.com google-site-verification=X78xao4MYo8Xbm-PGcKYB32LoESVj_7jNhVG-zWCV_w
[*]      TXT coinhako.com atlassian-domain-verification=Yk250IVOAmucoUJKKDyFcIBs0yLaFRx4lXZzThxAjXcdg8yFHoTA1j
Ec6vehxgO7
[*]      TXT coinhako.com v=spf1 include:amazonses.com include:_spf.google.com include:sendgrid.net include:sp
f.zoho.com -all
[*]      TXT coinhako.com MS=ms10980322
[*]      TXT coinhako.com google-site-verification=play9fkOyZo39jflasA8Z7oaDJdjSK3r3fs3eE
[*]      TXT coinhako.com google-site-verification=nSPVj8yifDT5tJzStiYPYge924rkYgNJCHt3ITsv-sg
[*]      TXT coinhako.com google-site-verification=zhFecsaffRLy9jmZyFYYdA2-killtuqzL4YC2jzGjnI
[*]      TXT coinhako.com CKO=cli_z27rvvw2kiwefd5qzqbz524wu4
[*]      TXT _dmarc.coinhako.com v=DMARC1; p=none; sp=none; fo=0:1:d:s; ri=3600; rua=mailto:3928b5324d6f458380
8f3b893435471a@dmarc-reports.cloudflare.net,mailto:dmarc_admin@coinhako.com; ruf=mailto:dmarc_admin@coinhako.c
om
[*] Enumerating SRV Records
[-] No SRV Records Found for coinhako.com
[*] Saving records to XML file: indrivdnsrecon.xml

┌──(root㉿kali)-[~]
└─# █
```

Public devices enumeration

- ## Censys

Censys is a potent internet scanning tool that scans and monitors devices, networks, and services online all the time. It performs thorough scans, keeps track of SSL/TLS certificates, keeps an eye on attack surfaces, finds vulnerabilities, and offers threat information. Large volumes of data may be searched and analyzed, processes can be automated with the help of the API, and Censys can be integrated into security workflows. In the connected digital world of today, Censys is useful for asset discovery, vulnerability management, compliance monitoring, and proactive risk reduction.

Proof of concept:



Basic Information

Forward DNS	www.hexoral.ru.cdn.cloudflare.net, public.cdr-api.fscu.com.au.cdn.cloudflare.net, www.paxlovideducacion.mx, judaspriest-namegenerator.com, uat.sales.soundunited.com, ...
Routing	104.18.36.0/24 via CLOUDFLARENET, US (AS13335)
Services (13)	80/HTTP, 443/HTTP, 2052/HTTP, 2053/HTTP, 2082/HTTP, 2083/HTTP, 2086/HTTP, 2087/HTTP, 2095/HTTP, 2096/HTTP, 8080/HTTP, 8443/HTTP, 8880/HTTP

HTTP 80/TCP

04/18/2024 07:49 UTC

Software

CloudFlare Load Balancer

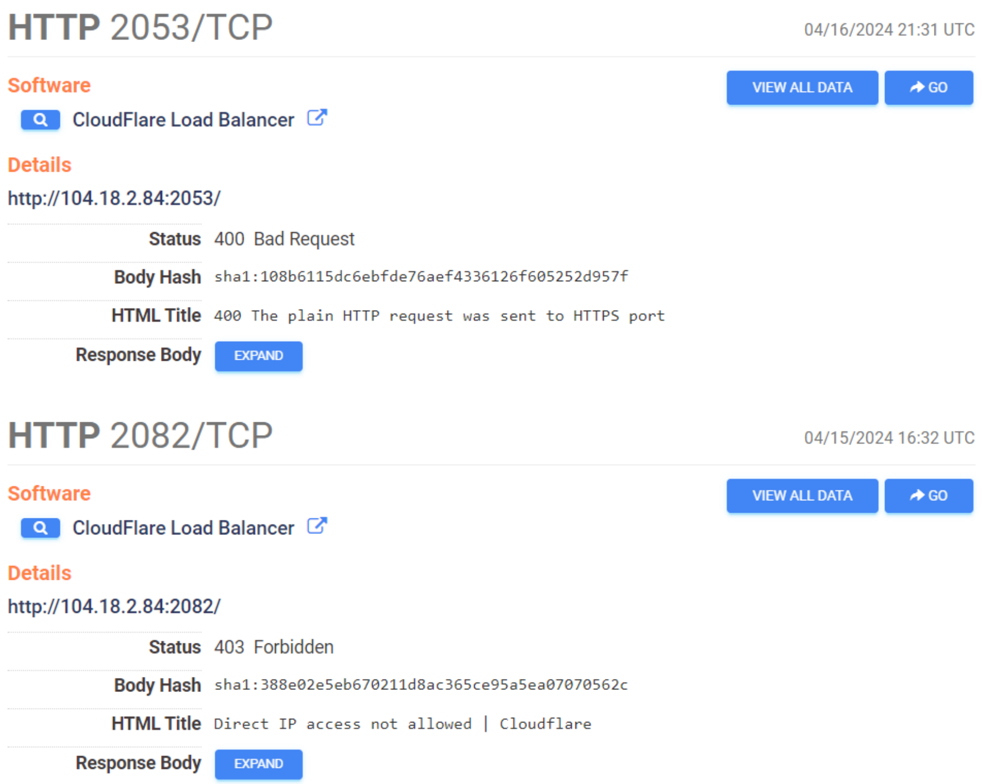
Details

http://104.18.36.214/

Status	403 Forbidden
Body Hash	sha1:e6ff0c29f2c57c6a58f1270a9e9b29407cc69017
HTML Title	Direct IP access not allowed | Cloudflare
Response Body	EXPAND

Geographic Location

City	San Francisco
State	California
Country	United States (US)
Coordinates	37.7621, -122.3971
Timezone	America/Los_Angeles



HTTP 2053/TCP

04/16/2024 21:31 UTC

Software

CloudFlare Load Balancer

Details

http://104.18.2.84:2053/

Status	400 Bad Request
Body Hash	`sha1:108b6115dc6ebfde76aef4336126f605252d957f`
HTML Title	`400 The plain HTTP request was sent to HTTPS port`
Response Body	EXPAND

HTTP 2082/TCP

04/15/2024 16:32 UTC

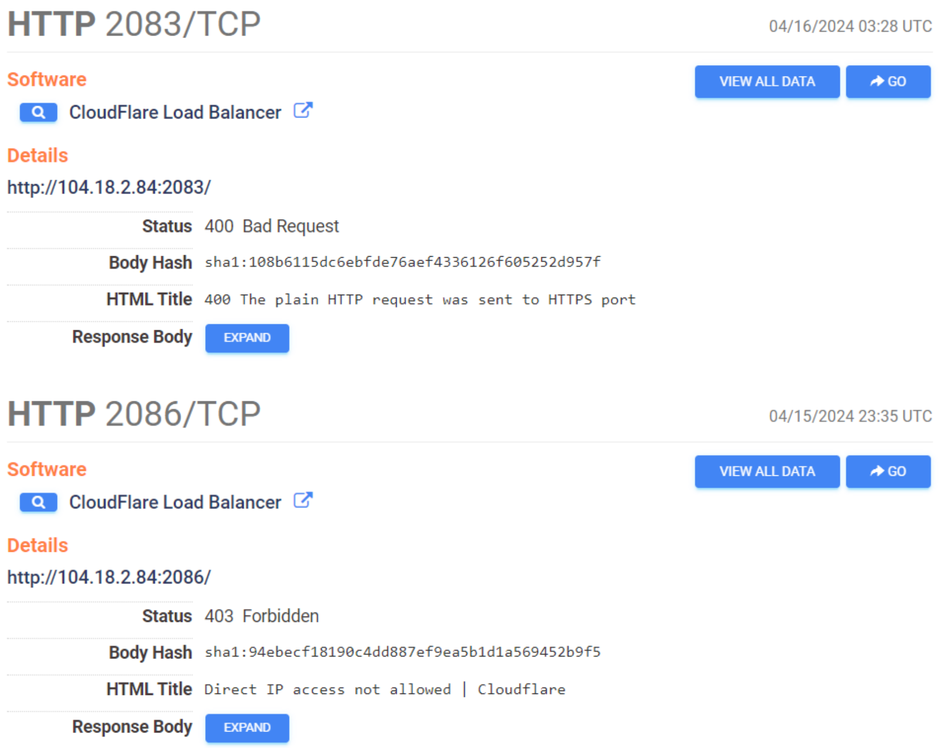
Software

CloudFlare Load Balancer

Details

http://104.18.2.84:2082/

Status	403 Forbidden
Body Hash	`sha1:388e02e5eb670211d8ac365ce95a5ea07070562c`
HTML Title	`Direct IP access not allowed | Cloudflare`
Response Body	EXPAND



HTTP 2083/TCP

04/16/2024 03:28 UTC

Software

CloudFlare Load Balancer

Details

http://104.18.2.84:2083/

Status	400 Bad Request
Body Hash	`sha1:108b6115dc6ebfde76aef4336126f605252d957f`
HTML Title	`400 The plain HTTP request was sent to HTTPS port`
Response Body	EXPAND

HTTP 2086/TCP

04/15/2024 23:35 UTC

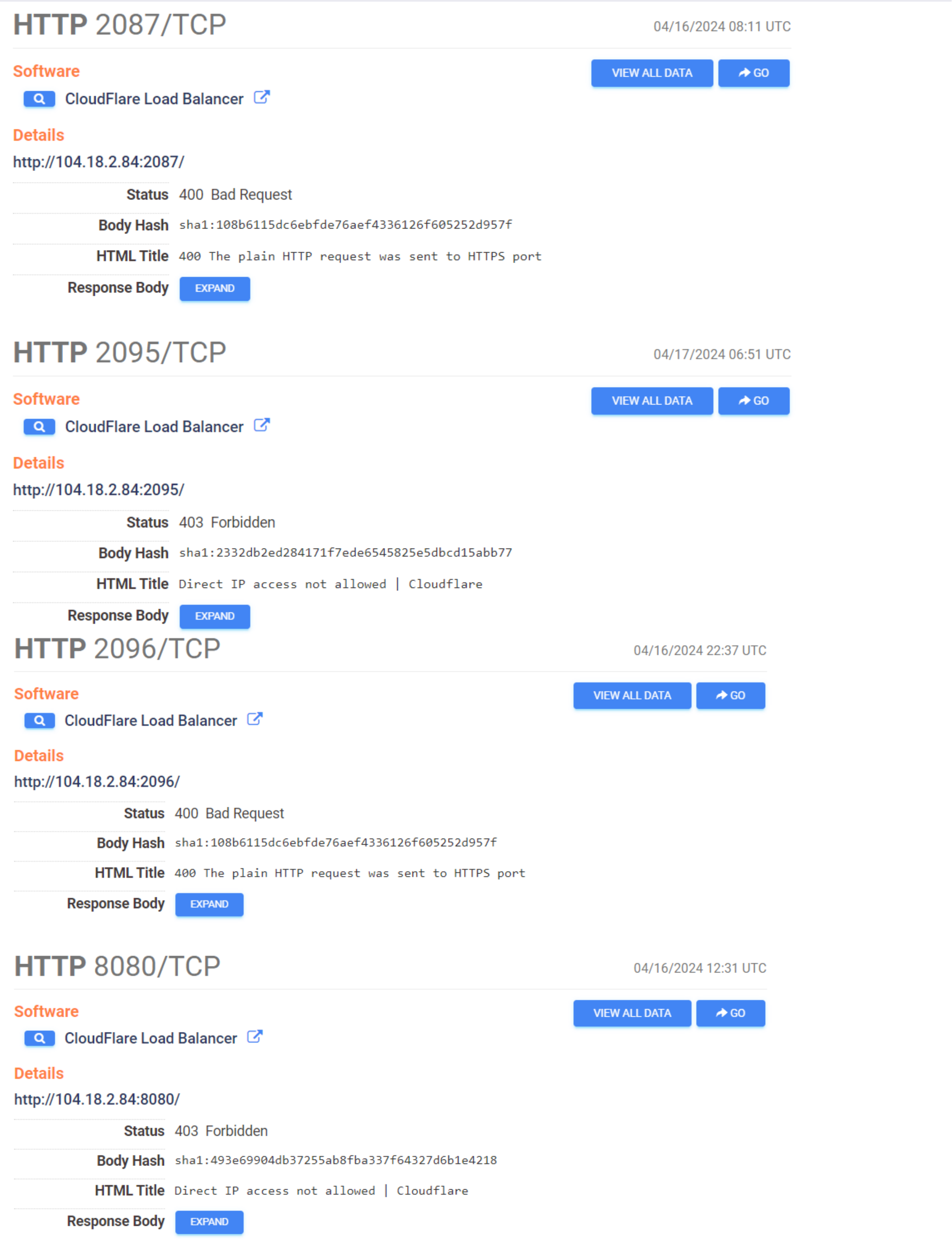
Software

CloudFlare Load Balancer

Details

http://104.18.2.84:2086/

Status	403 Forbidden
Body Hash	`sha1:94ebecf18190c4dd887ef9ea5b1d1a569452b9f5`
HTML Title	`Direct IP access not allowed | Cloudflare`
Response Body	EXPAND



HTTP 2087/TCP

04/16/2024 08:11 UTC

Software

CloudFlare Load Balancer

Details

http://104.18.2.84:2087/

Status	400 Bad Request
Body Hash	`sha1:108b6115dc6ebfde76aef4336126f605252d957f`
HTML Title	`400 The plain HTTP request was sent to HTTPS port`
Response Body	EXPAND

HTTP 2095/TCP

04/17/2024 06:51 UTC

Software

CloudFlare Load Balancer

Details

http://104.18.2.84:2095/

Status	403 Forbidden
Body Hash	`sha1:2332db2ed284171f7ede6545825e5dbcd15abb77`
HTML Title	`Direct IP access not allowed | Cloudflare`
Response Body	EXPAND

HTTP 2096/TCP

04/16/2024 22:37 UTC

Software

CloudFlare Load Balancer

Details

http://104.18.2.84:2096/

Status	400 Bad Request
Body Hash	`sha1:108b6115dc6ebfde76aef4336126f605252d957f`
HTML Title	`400 The plain HTTP request was sent to HTTPS port`
Response Body	EXPAND

HTTP 8080/TCP

04/16/2024 12:31 UTC

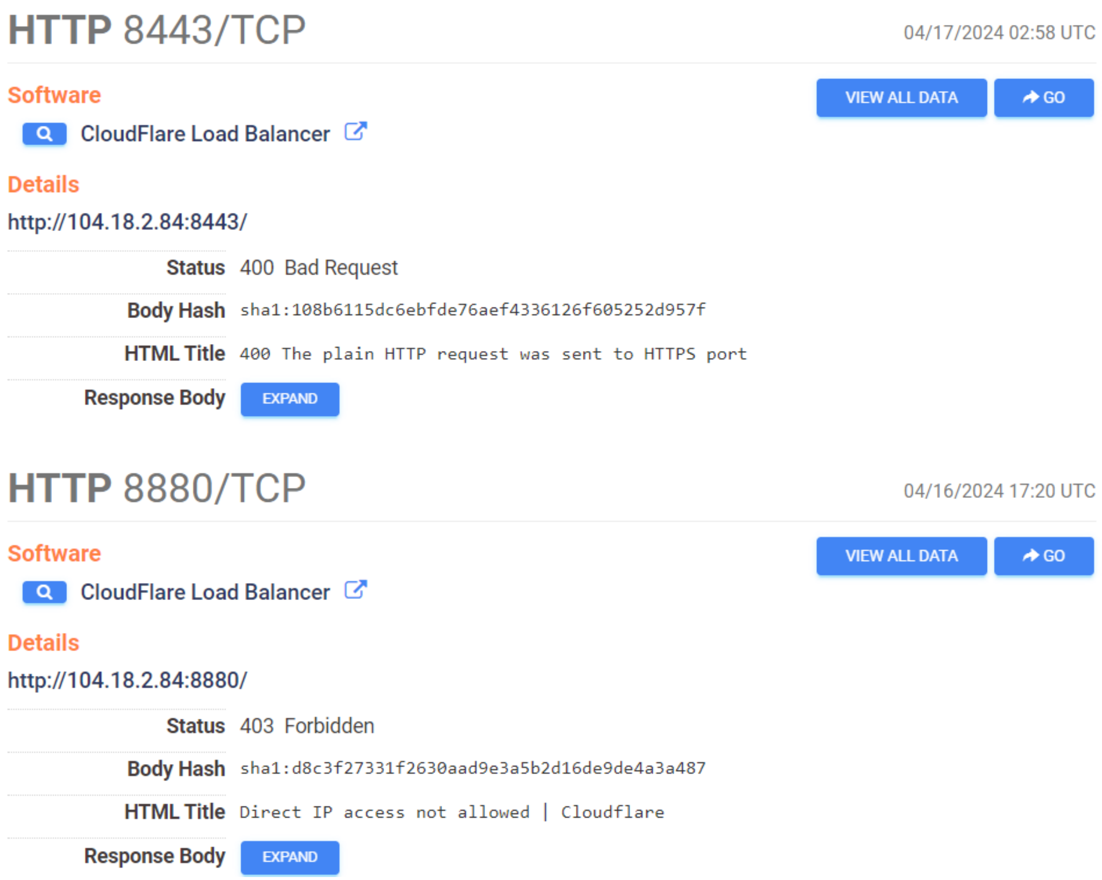
Software

CloudFlare Load Balancer

Details

http://104.18.2.84:8080/

Status	403 Forbidden
Body Hash	`sha1:493e69904db37255ab8fba337f64327d6b1e4218`
HTML Title	`Direct IP access not allowed | Cloudflare`
Response Body	EXPAND



HTTP 8443/TCP

04/17/2024 02:58 UTC

Software

CloudFlare Load Balancer

Details

http://104.18.2.84:8443/

```
Status         400 Bad Request
Body Hash      sha1:108b6115dc6ebfde76aef4336126f605252d957f
HTML Title     400 The plain HTTP request was sent to HTTPS port
Response Body
```

HTTP 8880/TCP

04/16/2024 17:20 UTC

Software

CloudFlare Load Balancer

Details

http://104.18.2.84:8880/

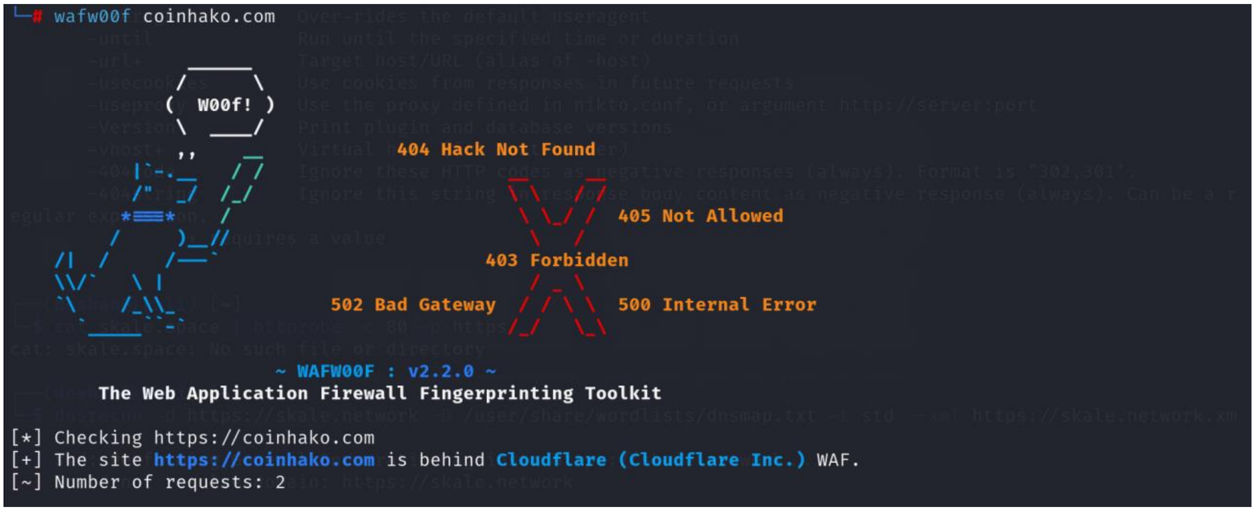
```
Status         403 Forbidden
Body Hash      sha1:d8c3f27331f2630aad9e3a5b2d16de9de4a3a487
HTML Title     Direct IP access not allowed | Cloudflare
Response Body
```

Find WAF (web application firewall) protection.

- **WafwOOf:**

Wafw00f is an open-source program that uses content patterns, HTTP responses, and header analysis to detect and fingerprint web application firewalls (WAFs). It gives testers and security experts information about the particular WAF technology or vendor being used as well as assists them in determining whether a web application is protected by a WAF. In addition to being user-friendly and seamlessly integrating with automated testing workflows, Wafw00f also keeps an updated database of WAF signatures and offers community support. It is a useful tool for security assessment and reconnaissance jobs, supporting the creation of efficient testing plans and workarounds.

Proof of concept:



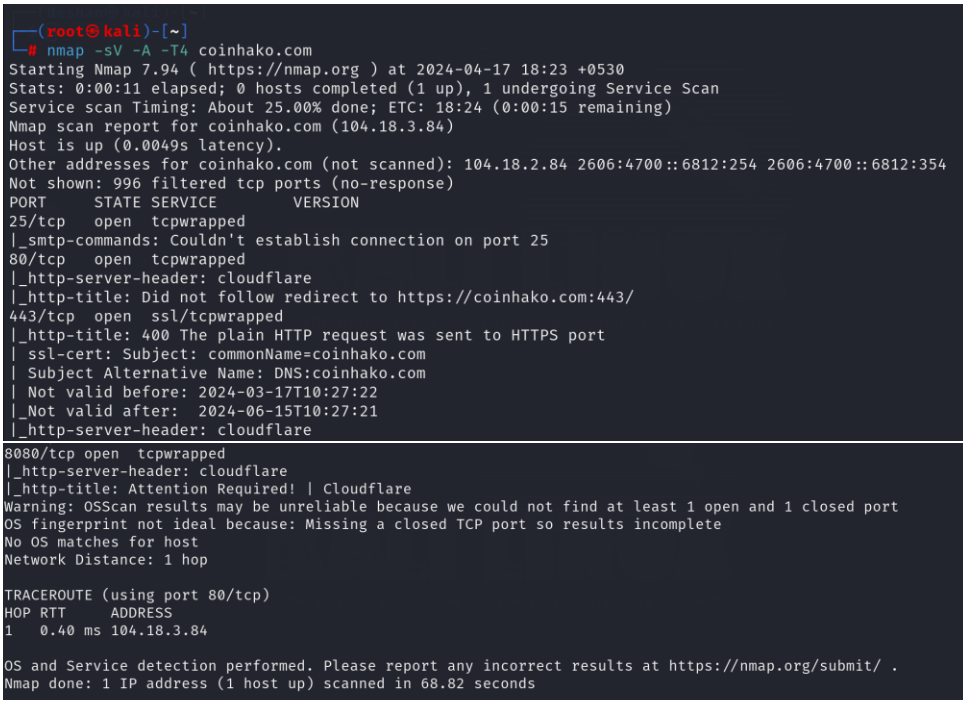
```
└─# wafw00f coinhako.com

                ( W00f! )
                                  404 Hack Not Found
                                                    405 Not Allowed
                                     403 Forbidden
                        502 Bad Gateway             500 Internal Error

                        ~ WAFW00F : v2.2.0 ~
        The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://coinhako.com
[+] The site https://coinhako.com is behind Cloudflare (Cloudflare Inc.) WAF.
[~] Number of requests: 2
```

Find open ports

- Nmap

Nmap, sometimes known as "Network Mapper," is a potent open-source network scanning program used for vulnerability analysis, security audits, and network discovery. It is particularly good at operating system detection, host discovery, port scanning, and service version detection. Because it supports custom scripting and automation, Nmap's scripting engine (NSE) is adaptable to a wide range of security activities and integration requirements. Because of its dependability, adaptability, and large feature set, network administrators, security experts, and penetration testers frequently use it for network reconnaissance, security audits, and network monitoring.

Proof of concept:

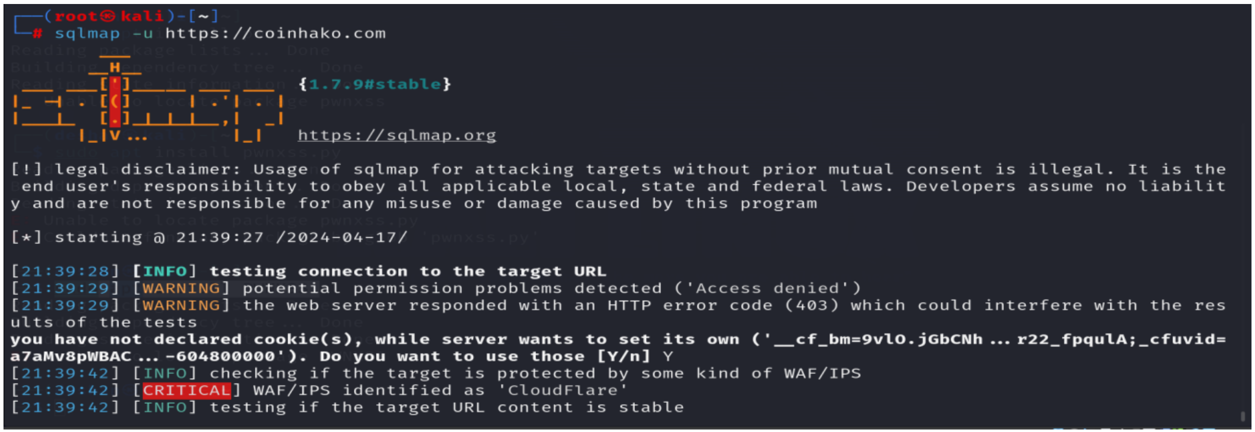


```
┌──(root㉿kali)-[~]
└─# nmap -sV -A -T4 coinhako.com
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-17 18:23 +0530
Stats: 0:00:11 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 25.00% done; ETC: 18:24 (0:00:15 remaining)
Nmap scan report for coinhako.com (104.18.3.84)
Host is up (0.0049s latency).
Other addresses for coinhako.com (not scanned): 104.18.2.84 2606:4700::6812:254 2606:4700::6812:354
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE        VERSION
25/tcp   open  tcpwrapped
|_smtp-commands: Couldn't establish connection on port 25
80/tcp   open  tcpwrapped
|_http-server-header: cloudflare
|_http-title: Did not follow redirect to https://coinhako.com:443/
443/tcp  open  ssl/tcpwrapped
|_http-title: 400 The plain HTTP request was sent to HTTPS port
| ssl-cert: Subject: commonName=coinhako.com
| Subject Alternative Name: DNS:coinhako.com
| Not valid before: 2024-03-17T10:27:22
|_Not valid after:  2024-06-15T10:27:21
|_http-server-header: cloudflare
8080/tcp open  tcpwrapped
|_http-server-header: cloudflare
|_http-title: Attention Required! | Cloudflare
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
No OS matches for host
Network Distance: 1 hop

TRACEROUTE (using port 80/tcp)
HOP RTT     ADDRESS
1   0.40 ms 104.18.3.84

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 68.82 seconds
```

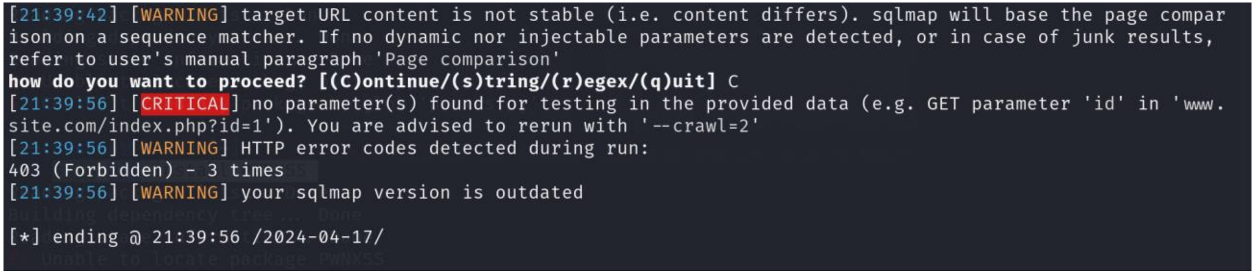
Exploitation

Sqlmap

Web applications with SQL injection vulnerabilities can be automatically found and exploited with SQLMap, an open-source penetration testing tool. It employs several detecting methods, supports several database management systems, and provides customization choices. Security experts and penetration testers find SQLMap useful as it automates testing, enumeration, data extraction, and reporting procedures during security assessments. It should, therefore, be utilized sensibly, morally, and with the appropriate authorization to test programs and systems.



```
┌──(root㉿kali)-[~]
└─# sqlmap -u https://coinhako.com
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.7.9#stable}
|_ -| . [(]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the
 end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liabilit
y and are not responsible for any misuse or damage caused by this program

[*] starting @ 21:39:27 /2024-04-17/

[21:39:28] [INFO] testing connection to the target URL
[21:39:29] [WARNING] potential permission problems detected ('Access denied')
[21:39:29] [WARNING] the web server responded with an HTTP error code (403) which could interfere with the res
ults of the tests
you have not declared cookie(s), while server wants to set its own ('__cf_bm=9vlO.jGbCNh...r22_fpqulA;_cfuvid=
a7aMv8pWBAC...-604800000'). Do you want to use those [Y/n] Y
[21:39:42] [INFO] checking if the target is protected by some kind of WAF/IPS
[21:39:42] [CRITICAL] WAF/IPS identified as 'CloudFlare'
[21:39:42] [INFO] testing if the target URL content is stable
```



```
[21:39:42] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the page compar
ison on a sequence matcher. If no dynamic nor injectable parameters are detected, or in case of junk results,
refer to user's manual paragraph 'Page comparison'
how do you want to proceed? [(C)ontinue/(s)tring/(r)egex/(q)uit] C
[21:39:56] [CRITICAL] no parameter(s) found for testing in the provided data (e.g. GET parameter 'id' in 'www.
site.com/index.php?id=1'). You are advised to rerun with '--crawl=2'
[21:39:56] [WARNING] HTTP error codes detected during run:
403 (Forbidden) - 3 times
[21:39:56] [WARNING] your sqlmap version is outdated

[*] ending @ 21:39:56 /2024-04-17/
```

Vulnerability analysis phase

I used tool like netsparker to process and catch bugs and vulnerabilities based on OWASP top 10.

Targeted Domain – https://www.coinhako.com

Vulnerability title

Weak Ciphers Enabled (Confirmed)

Vulnerability description

The "Weak Ciphers Enabled" vulnerability stems from the use of outdated or insecure cryptographic ciphers in SSL/TLS configurations, exposing systems to cryptographic attacks. Weak ciphers like DES, RC4, and MD5 lack the necessary security standards and are prone to exploitation.

Impact assessment

- Attackers might decrypt SSL traffic between your server and your visitors.

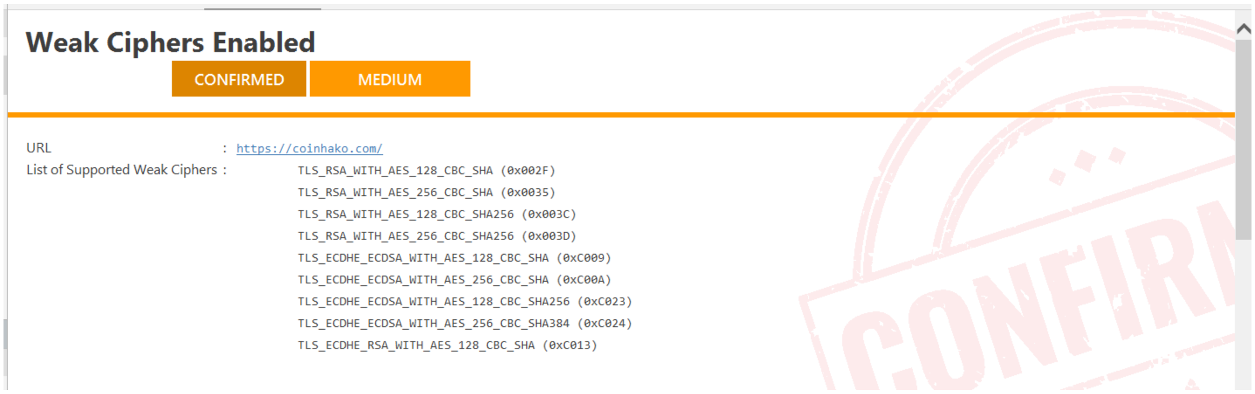
Affected components

- Server's SSL/TLS configuration, cryptographic ciphers, and client-side implementations affected.

How to mitigate?

- Disable outdated and insecure SSL/TLS protocol versions (e.g., SSL 2.0, SSL 3.0).
- Disable weak cryptographic ciphers (e.g., DES, RC4, MD5) and enable only strong ciphers recommended by security standards.
- Regularly update SSL/TLS configurations, server software, and cryptographic libraries to mitigate newly discovered vulnerabilities.

Weak cipher enabled (confirmed).



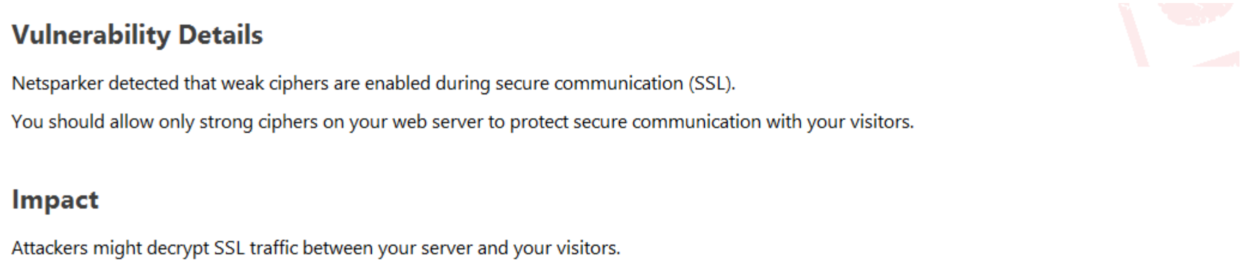
Weak Ciphers Enabled

CONFIRMED MEDIUM

URL : https://coinhako.com/

List of Supported Weak Ciphers :

```
TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003C)
TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003D)
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xC009)
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xC00A)
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xC023)
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xC024)
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)
```



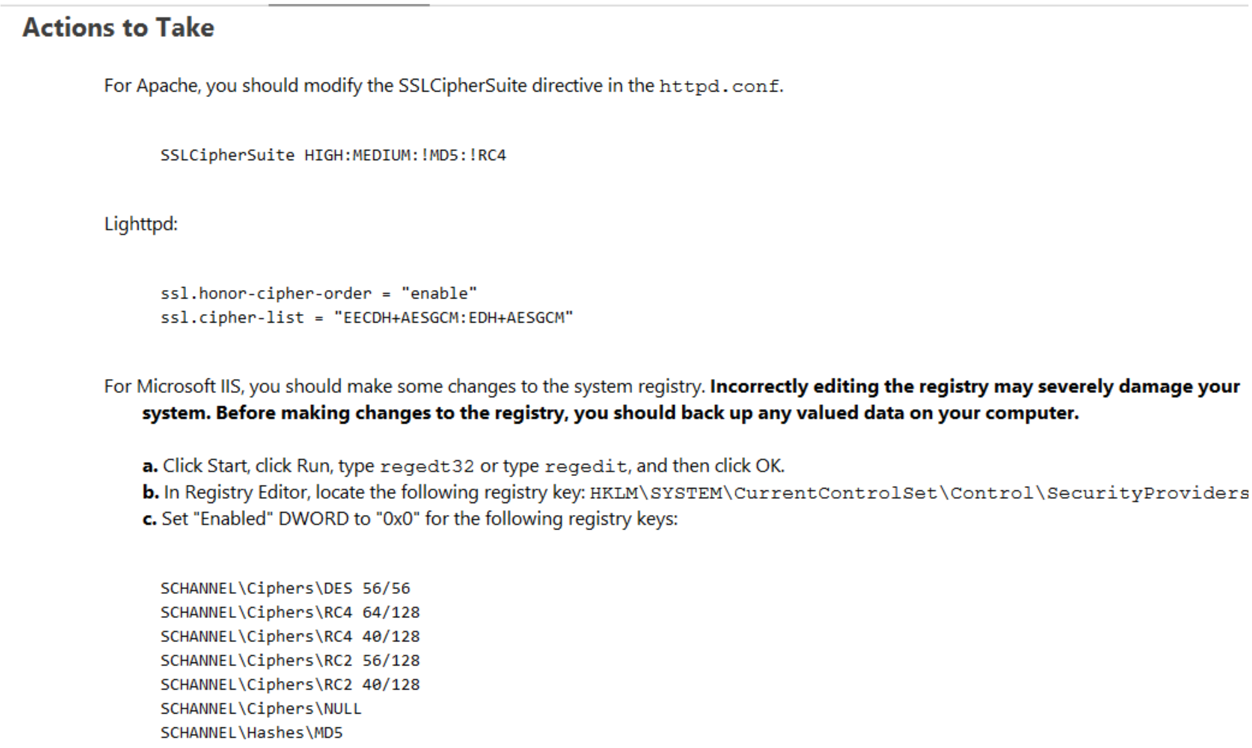
Vulnerability Details

Netsparker detected that weak ciphers are enabled during secure communication (SSL).

You should allow only strong ciphers on your web server to protect secure communication with your visitors.

Impact

Attackers might decrypt SSL traffic between your server and your visitors.



Actions to Take

For Apache, you should modify the SSLCipherSuite directive in the `httpd.conf`.

```
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4
```

Lighttpd:

```
ssl.honor-cipher-order = "enable"
ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM"
```

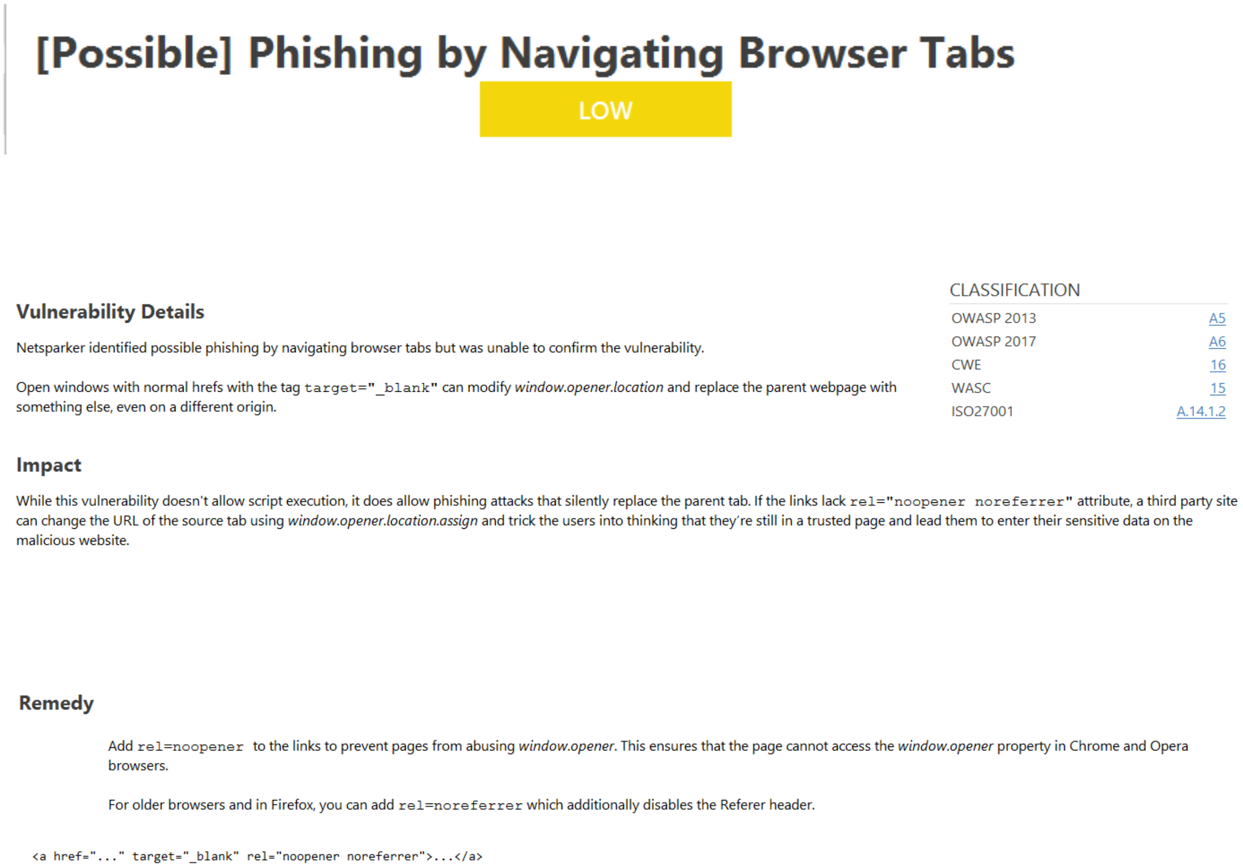
For Microsoft IIS, you should make some changes to the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**

a. Click Start, click Run, type `regedt32` or type `regedit`, and then click OK.
b. In Registry Editor, locate the following registry key: `HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`
c. Set "Enabled" DWORD to "0x0" for the following registry keys:

```
SCHANNEL\Ciphers\DES 56/56
SCHANNEL\Ciphers\RC4 64/128
SCHANNEL\Ciphers\RC4 40/128
SCHANNEL\Ciphers\RC2 56/128
SCHANNEL\Ciphers\RC2 40/128
SCHANNEL\Ciphers\NULL
SCHANNEL\Hashes\MD5
```

Other vulnerabilities were identified during the scan

- **[Possible] Phishing by Navigating Browser Tabs**



[Possible] Phishing by Navigating Browser Tabs

LOW

Vulnerability Details

Netsparker identified possible phishing by navigating browser tabs but was unable to confirm the vulnerability.

Open windows with normal hrefs with the tag `target="_blank"` can modify *window.opener.location* and replace the parent webpage with something else, even on a different origin.

CLASSIFICATION

OWASP 2013	A5
OWASP 2017	A6
CWE	16
WASC	15
ISO27001	A.14.1.2

Impact

While this vulnerability doesn't allow script execution, it does allow phishing attacks that silently replace the parent tab. If the links lack `rel="noopener noreferrer"` attribute, a third party site can change the URL of the source tab using *window.opener.location.assign* and trick the users into thinking that they're still in a trusted page and lead them to enter their sensitive data on the malicious website.

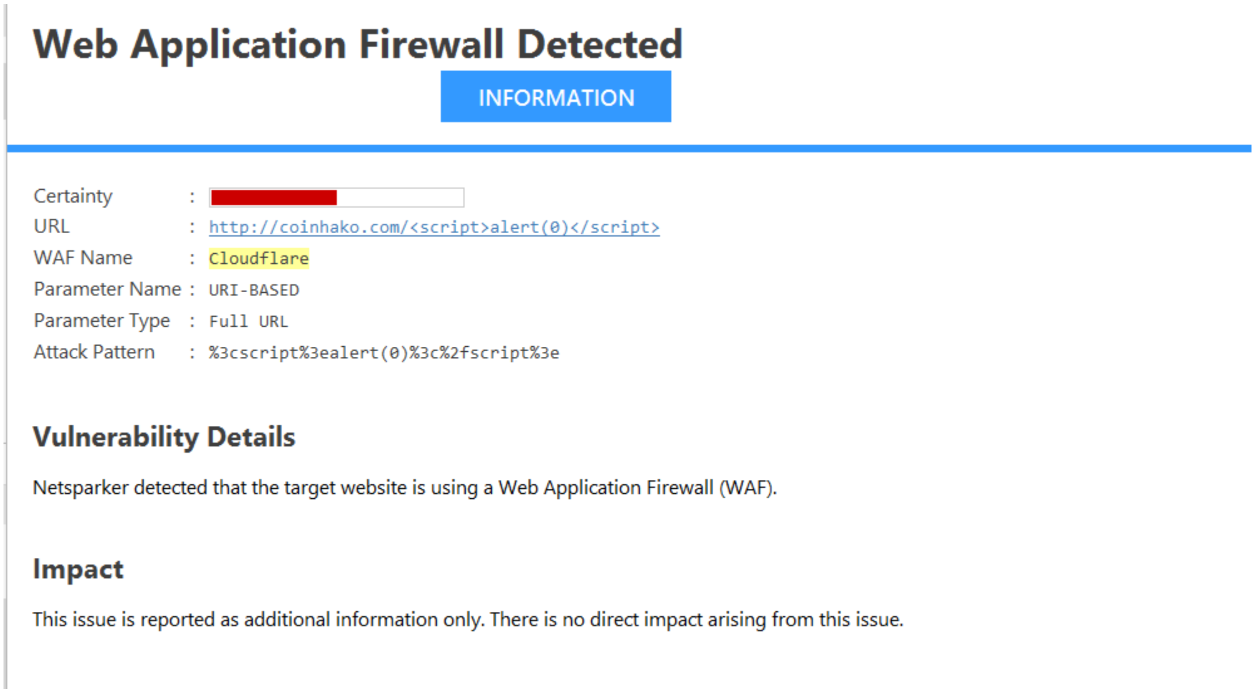
Remedy

Add `rel=noopener` to the links to prevent pages from abusing *window.opener*. This ensures that the page cannot access the *window.opener* property in Chrome and Opera browsers.

For older browsers and in Firefox, you can add `rel=noreferrer` which additionally disables the Referer header.

```
<a href="..." target="_blank" rel="noopener noreferrer">...</a>
```

- **Web Application Firewall Detected**



Web Application Firewall Detected

INFORMATION

Certainty :
URL : http://coinhako.com/<script>alert(0)</script>
WAF Name : Cloudflare
Parameter Name : URI-BASED
Parameter Type : Full URL
Attack Pattern : %3cscript%3ealert(0)%3c%2fscript%3e

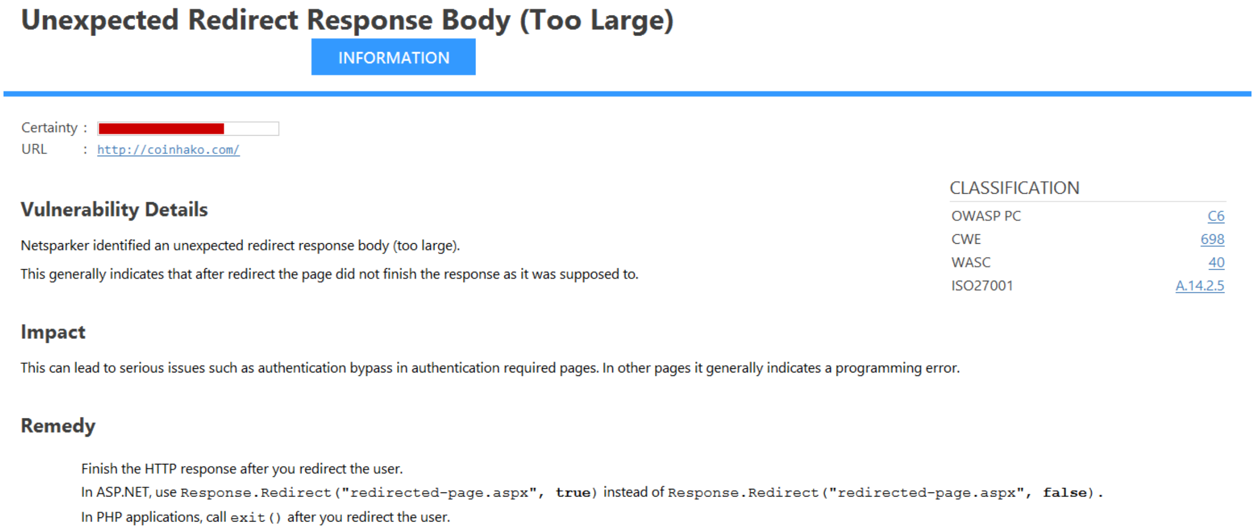
Vulnerability Details

Netsparker detected that the target website is using a Web Application Firewall (WAF).

Impact

This issue is reported as additional information only. There is no direct impact arising from this issue.

- **Unexpected Redirect Response Body (Too Large)**



Unexpected Redirect Response Body (Too Large)

INFORMATION

Certainty :
URL : http://coinhako.com/

Vulnerability Details

Netsparker identified an unexpected redirect response body (too large).

This generally indicates that after redirect the page did not finish the response as it was supposed to.

CLASSIFICATION

OWASP PC	C6
CWE	698
WASC	40
ISO27001	A.14.2.5

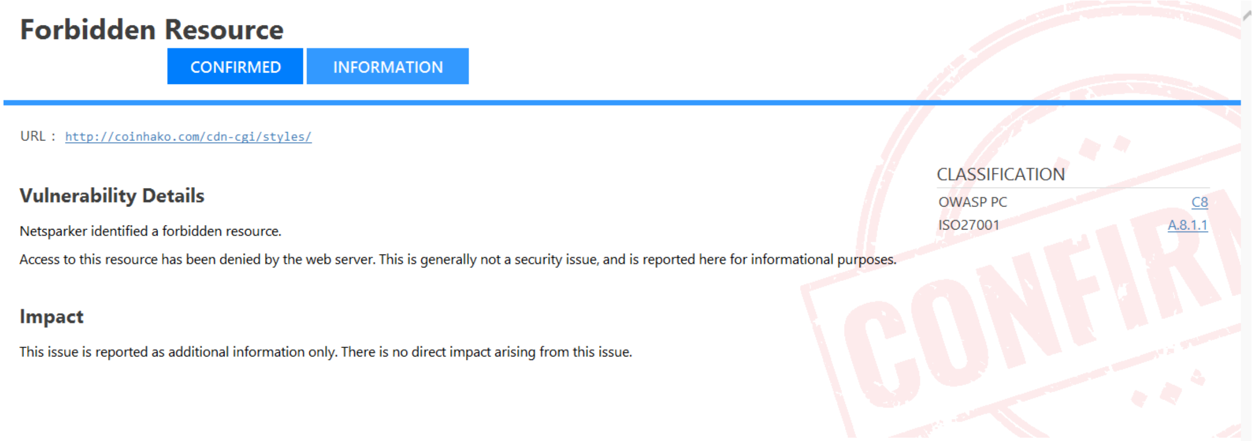
Impact

This can lead to serious issues such as authentication bypass in authentication required pages. In other pages it generally indicates a programming error.

Remedy

Finish the HTTP response after you redirect the user.
In ASP.NET, use `Response.Redirect("redirected-page.aspx", true)` instead of `Response.Redirect("redirected-page.aspx", false)`.
In PHP applications, call `exit()` after you redirect the user.

- ## Forbidden Resource



Forbidden Resource

URL : http://coinhako.com/cdn-cgi/styles/

Vulnerability Details

Netsparker identified a forbidden resource.

Access to this resource has been denied by the web server. This is generally not a security issue, and is reported here for informational purposes.

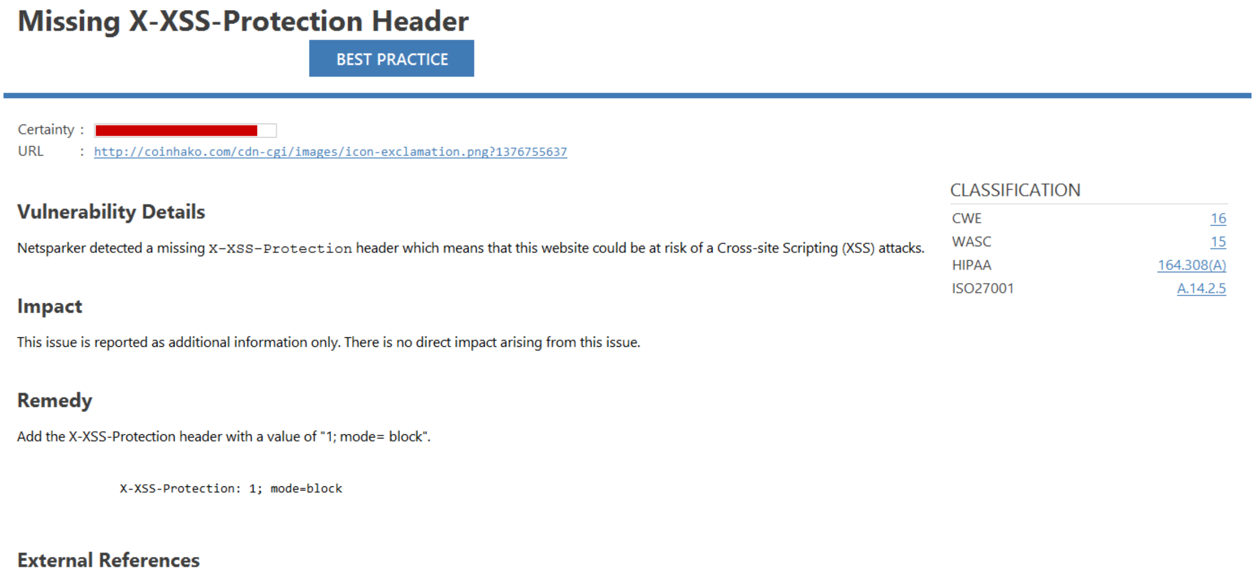
Impact

This issue is reported as additional information only. There is no direct impact arising from this issue.

CLASSIFICATION

OWASP PC	C8
ISO27001	A.8.1.1

- ## Missing X-XSS-Protection Header



Missing X-XSS-Protection Header

Certainty :
URL : http://coinhako.com/cdn-cgi/images/icon-exclamation.png?1376755637

Vulnerability Details

Netsparker detected a missing `X-XSS-Protection` header which means that this website could be at risk of a Cross-site Scripting (XSS) attacks.

Impact

This issue is reported as additional information only. There is no direct impact arising from this issue.

Remedy

Add the X-XSS-Protection header with a value of "1; mode= block".

```
X-XSS-Protection: 1; mode=block
```

External References

CLASSIFICATION

CWE	16
WASC	15
HIPAA	164.308(A)
ISO27001	A.14.2.5

`

Conclusion

The analysis underscores the importance of addressing weak ciphers, security misconfigurations, and potential phishing risks to enhance the overall security posture of the "coinhako.com" domain. Implementing the recommended mitigations and best practices will help mitigate risks and protect against common web application vulnerabilities.

The comprehensive analysis and actionable recommendations provided a structured approach to improving security and addressing potential threats effectively. Continuously monitoring and updating security measures will further strengthen the resilience of the system against evolving security challenges.