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
    ii. Sublist3r
    iii. uniscan

c. DNS enumeration
    i. Dnsrecon
    ii. Nikto
    iii. Nslookup

d. Public devices enumeration
    i. Censys
    ii. Whatweb
    iii. Whois

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

f. Find open ports.
    i. Nmap

g. Exploitation
    i. sqlmap

vulnerability analysis phase

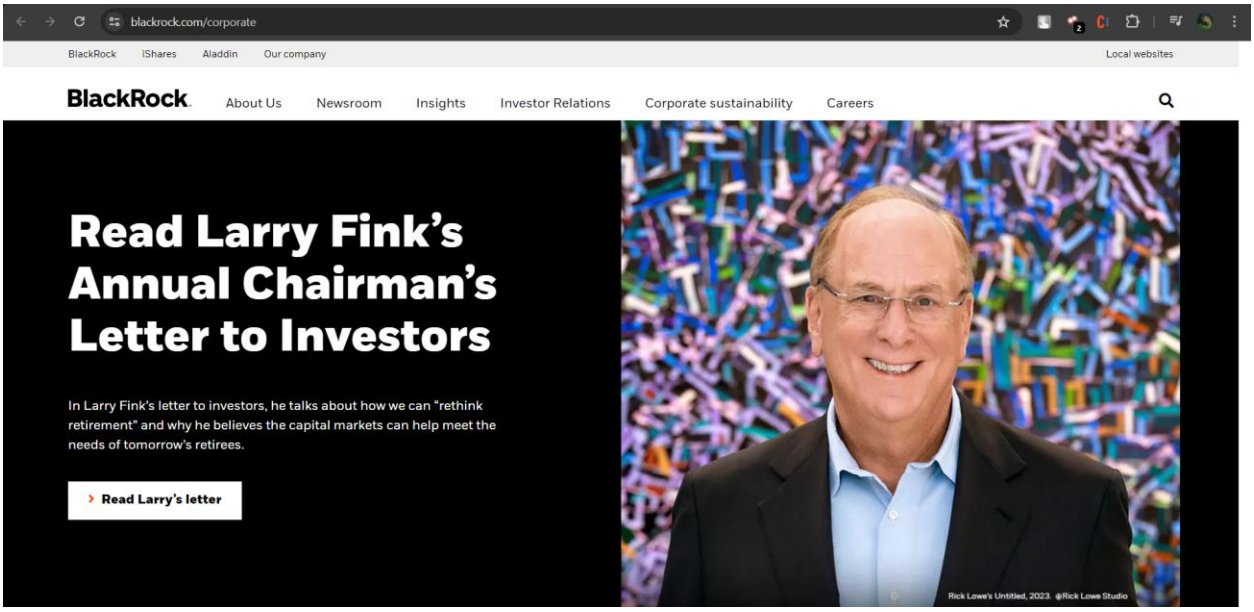
1. Target domain: http://blackrock.com

    a. CSP: Wildcard Directive
    b. Absence of Anti-CSRF Tokens
    c. CSP: script-src unsafe-eval
    d. CSP: style-src unsafe-inline
    e. Hidden File Found
    f. Cookie without SameSite Attribute

Conclusion

Scope:

Based in the US, BlackRock is a multinational investment management firm. Asset management, risk management, and advisory services are just a few of the many financial services they are renowned for providing. The worldwide clientele of BlackRock include governments, businesses, institutional investors, and private citizens. Information about their services, financial products, market insights, and corporate news are probably available on their website, http://blackrock.com.



In Scope:

Asset name	Type	Coverage	Max. severity	Bounty	Last update	Resolved Reports
*.ishares.com Researchers must use HackerOne email aliases when registering for accounts: [username]@wearehackerone.com. No credentials will be provided, but you can register for up to 3 accounts. Akamai DDOS	Other	In scope	Critical	Ineligible	May 12, 2023	0 (0%)
69.52.0.0/16 Akamai DDOS	CIDR	In scope	Critical	Ineligible	Jan 24, 2023	0 (0%)
*.isharesonline.com Researchers must use HackerOne email aliases when registering for accounts: [username]@wearehackerone.com. No credentials will be provided, but you can register for up to 3 accounts in case there is an option.	Other	In scope	Critical	Ineligible	Mar 27, 2024	0 (0%)
*.blackrock.com Researchers must use HackerOne email aliases when registering for accounts: [username]@wearehackerone.com. No credentials will be provided, but you can register for up to 3 accounts. Akamai DDOS apache Kubernetes Nginx	Other	In scope	Critical	Ineligible	Jul 13, 2023	56 (431%)

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:

- To get google website give this command.

- 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.

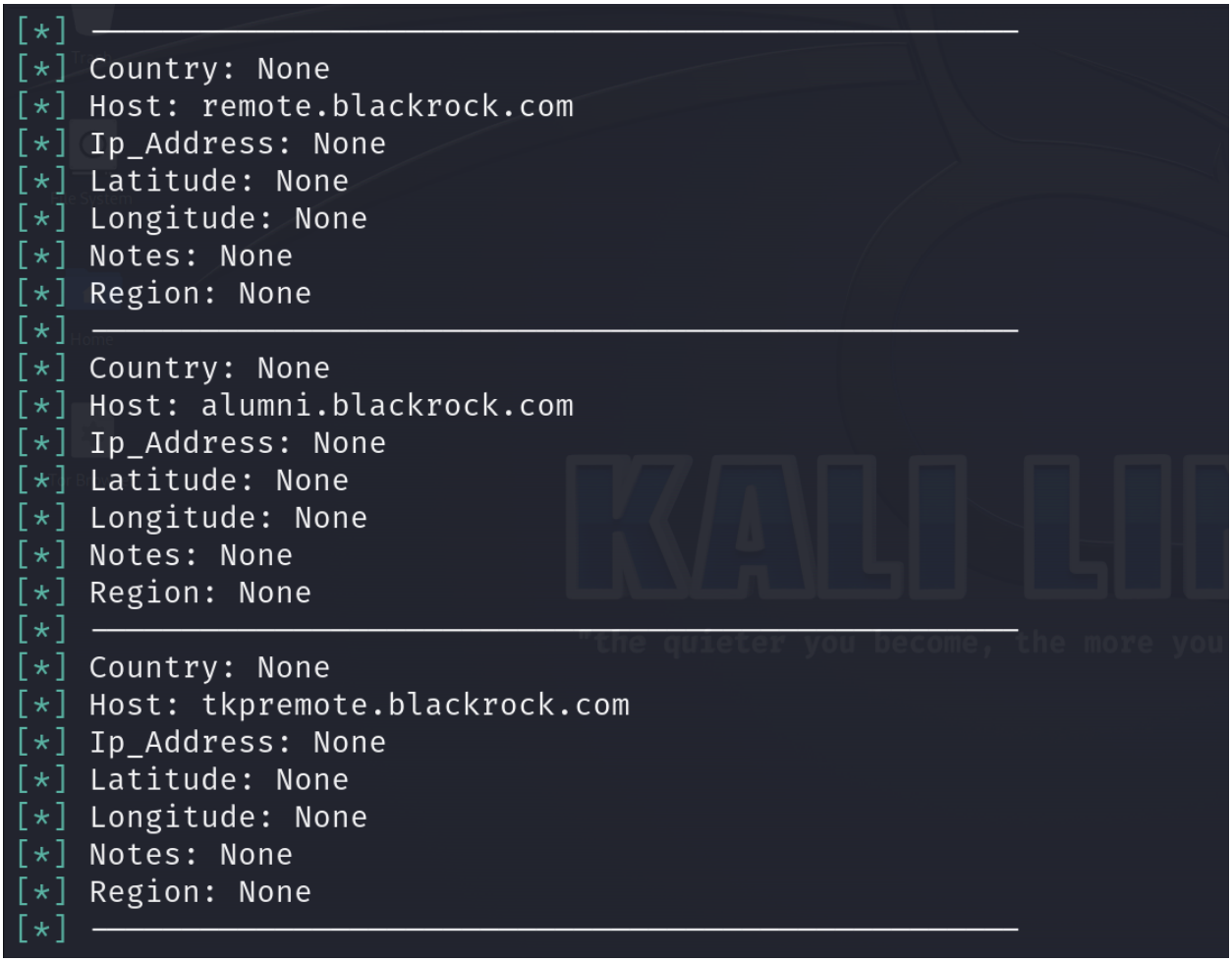
```
[recon-ng][default] > modules load recon/domains-hosts/google_site_web
[recon-ng][default][google_site_web] > info

       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] >
```

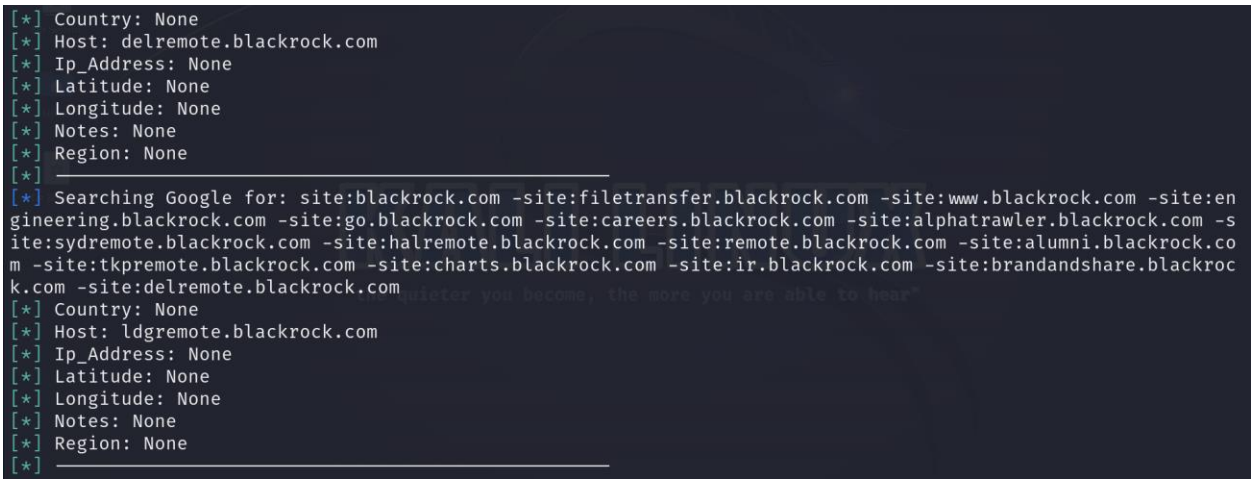
- Go to options and set source to our targeted domain [www. blackrock.com](http://www.blackrock.com) and run it.

```
      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  redis.com      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] > options set source blackrock.com
SOURCE => blackrock.com
[recon-ng][default][google_site_web] > run
```

```
BLACKROCK.COM
-------------
[*] Searching Google for: site:blackrock.com
[*] Country: None
[*] Host: filetransfer.blackrock.com
[*] Ip_Address: None
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] -------------------------------------------------------
[*] Country: None
[*] Host: www.blackrock.com
[*] Ip_Address: None
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] -------------------------------------------------------
[*] Country: None
[*] Host: engineering.blackrock.com
[*] Ip_Address: None
[*] Latitude: None
[*] Longitude: None
```


[*] ——
[*] Country: None
[*] Host: careers.blackrock.com
[*] Ip_Address: None
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ——
[*] Searching Google for: site:blackrock.com -site:filetransfer.blackrock.com -site:www.blackrock.com -site:engineering.blackrock.com -site:go.blackrock.com -site:careers.blackrock.com
[*] Country: None
[*] Host: alphatrawler.blackrock.com
[*] Ip_Address: None
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ——


[*] ——
[*] Country: None
[*] Host: remote.blackrock.com
[*] Ip_Address: None
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ——
[*] Country: None
[*] Host: alumni.blackrock.com
[*] Ip_Address: None
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ——
[*] Country: None
[*] Host: tkpremote.blackrock.com
[*] Ip_Address: None
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ——

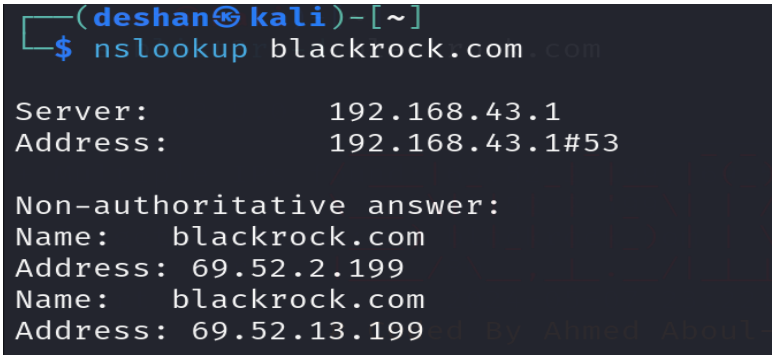


Getting alive subdomains

- 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.

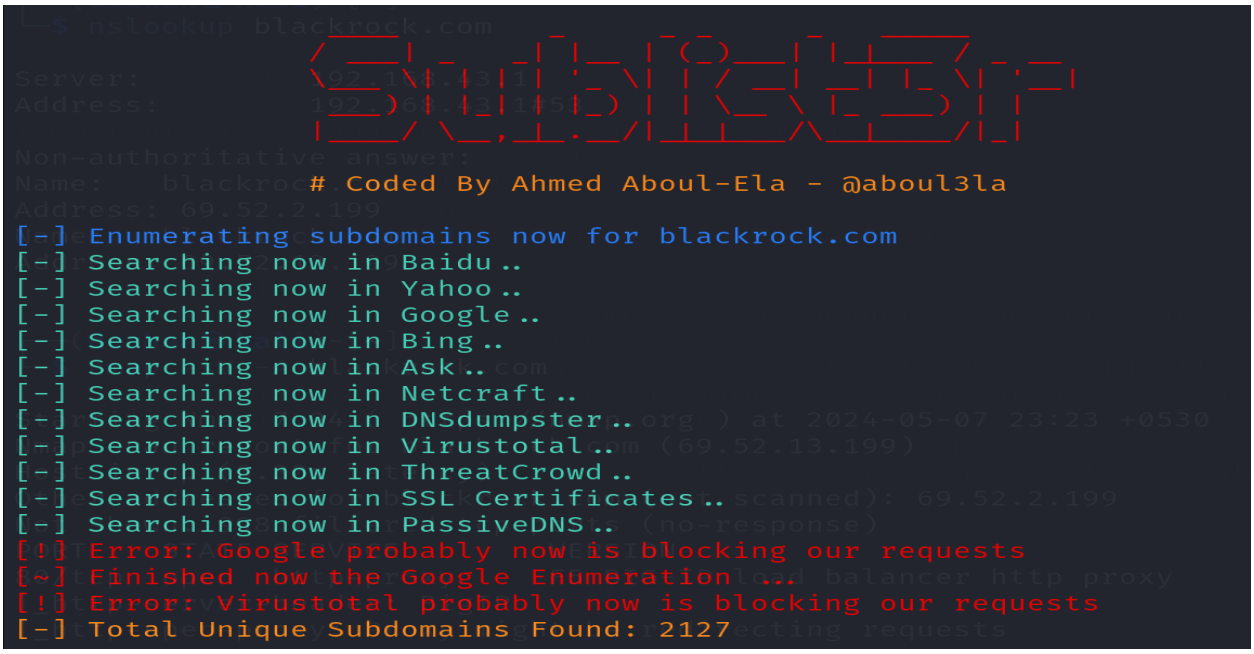
Proof of concept:



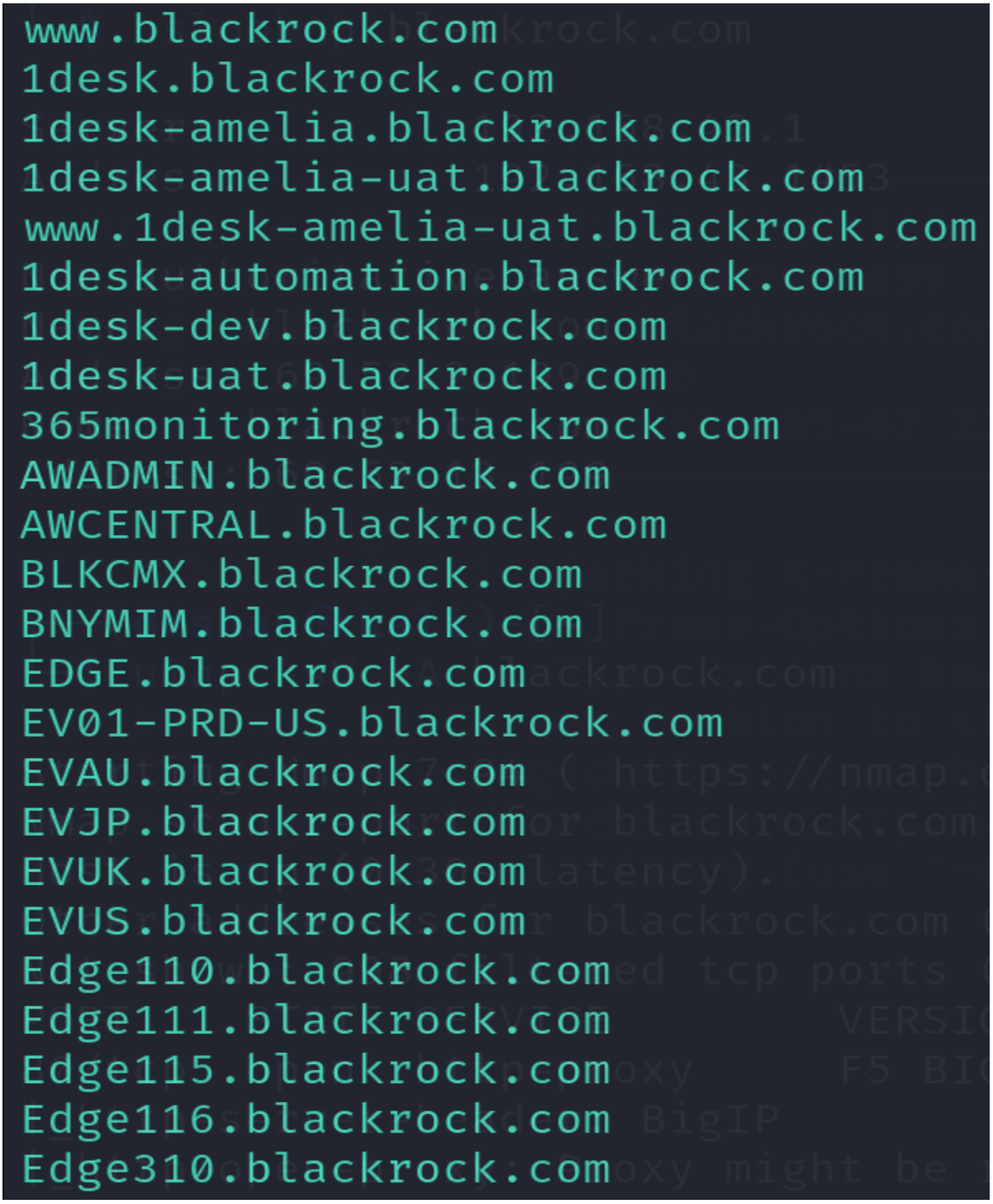
- Sublist3r

Sublist3r is an open-source subdomain enumeration tool used in cybersecurity for reconnaissance purposes. It assists security professionals, penetration testers, and researchers in identifying valid subdomains associated with a target domain. The tool employs various methods including passive search through search engine results, active search via DNS queries, and brute-force techniques using wordlists to enumerate subdomains. Sublist3r supports output in multiple formats, making it versatile for integration with other tools and workflows. It is valuable in expanding the attack surface during security assessments but should be used responsibly and with proper authorization.

Proof of concept:



Proof of concept:



Proof of concept:

```
aep.blackrock.com
www.aep.blackrock.com
aflac.blackrock.com
www.aflac.blackrock.com
aig.blackrock.com
www.aig.blackrock.com
akaan.blackrock.com
www.akaan.blackrock.com
aladdin.blackrock.com
app.aladdin.blackrock.com
www.app.aladdin.blackrock.com
images.aladdin.blackrock.com
www.images.aladdin.blackrock.com
aladdin-login.blackrock.com
www.aladdin-login.blackrock.com
aladdin-vault.blackrock.com
aladdinaccounting.blackrock.com
www.aladdinaccounting.blackrock.com
aladdinforesight.blackrock.com
demo.aladdinforesight.blackrock.com
demomuse2.aladdinforesight.blackrock.com
demomusw2.aladdinforesight.blackrock.com
dev.aladdinforesight.blackrock.com
devmuse2.aladdinforesight.blackrock.com
devmusw2.aladdinforesight.blackrock.com
```

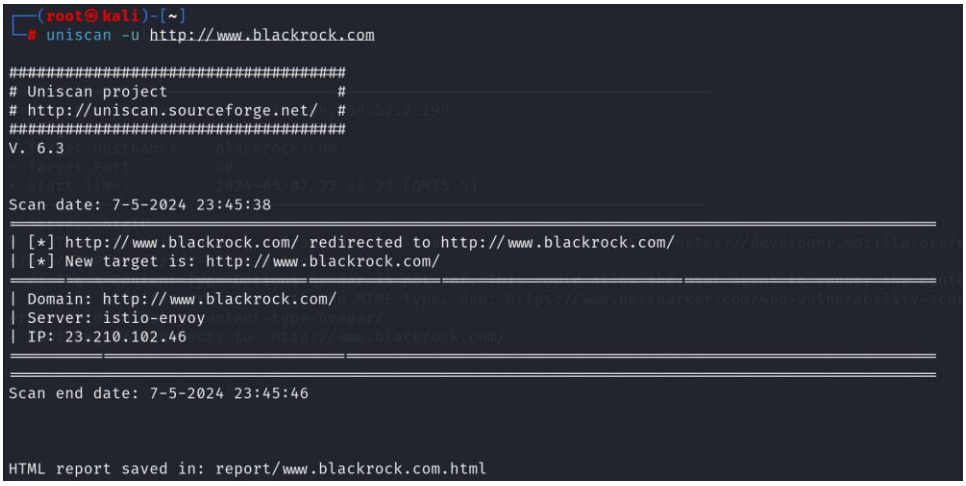
Proof of concept:

www.wellsfargo.blackrock.com
wfc.blackrock.com
www.wfc.blackrock.com
wireless.blackrock.com
wisayah.blackrock.com
www.wisayah.blackrock.com
wrberkley.blackrock.com
www.wrberkley.blackrock.com
wwpsh.blackrock.com
www.wwpsh.blackrock.com
www2.blackrock.com
www.www2.blackrock.com
xceptor.blackrock.com
xceptordr.blackrock.com
xceptorprod.blackrock.com
xterm.blackrock.com
xterm-ewd.blackrock.com
xterm-hal.blackrock.com
xxibanorte.blackrock.com
www.xxibanorte.blackrock.com
zeus.blackrock.com
www.zeus.blackrock.com

- Uniscan

A free penetration testing tool is Uniscan. This program is used to check for vulnerabilities in web applications. With the help of this scan, we can check the target online application for SQL injection, cross-site scripting (XSS), PHP injection, Remote file inclusion (LFI), remote command execution, web shell vulnerabilities, and backup files.

Proof of concept:

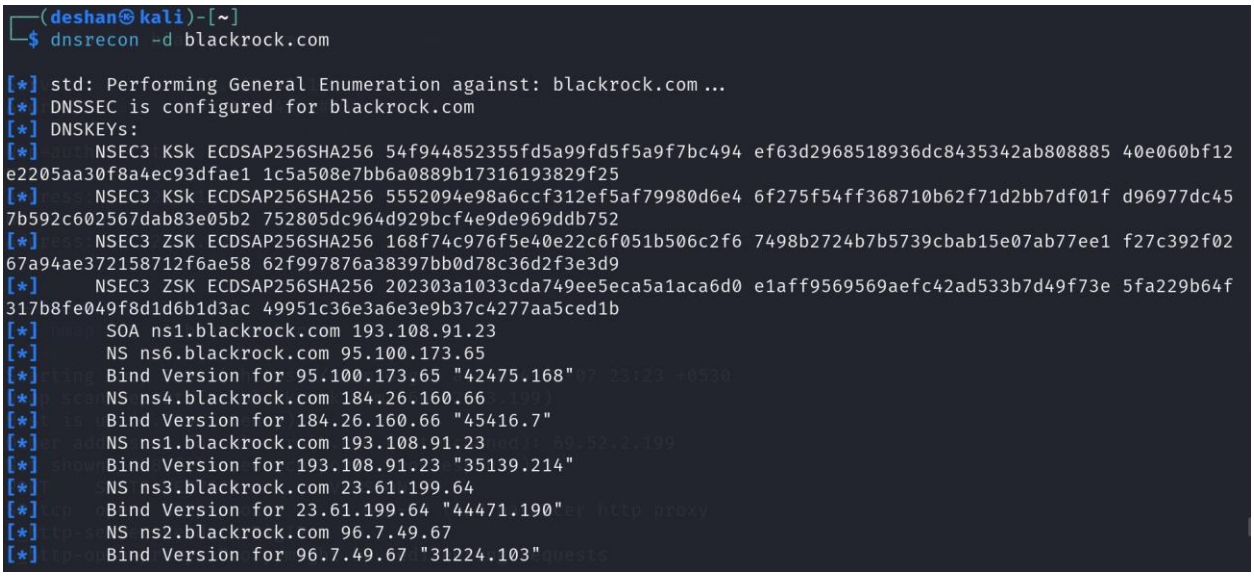


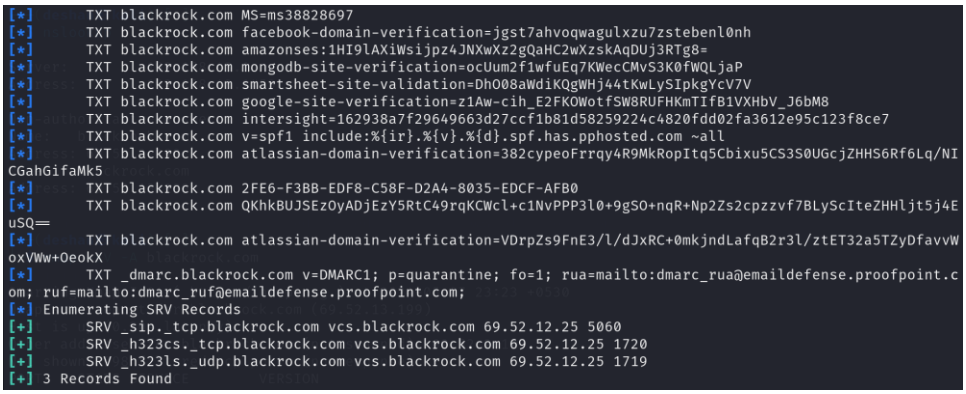
DNS enumeration

- 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:



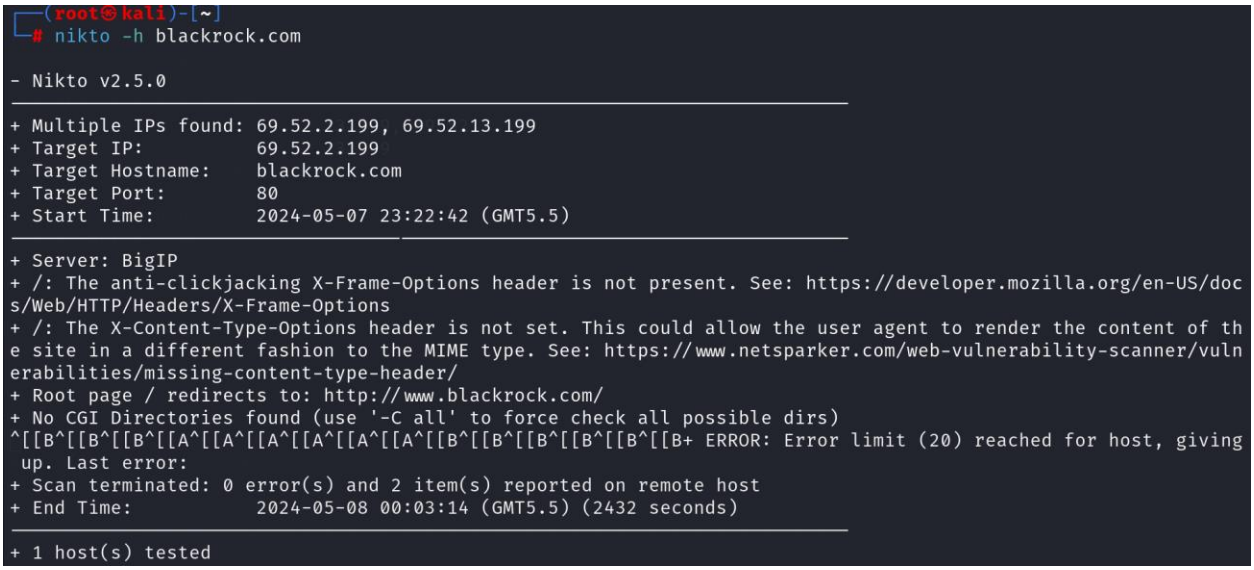


- Nikto

Nikto is an open-source web server scanner made to find possible security holes in applications and web servers. With its extensive scanning capabilities, it may

identify problems including out-of-date software, unsafe configurations, weak scripts, and incorrect SSL/TLS settings. In addition to producing comprehensive results after scanning, Nikto may be customized and scripted, connects with automated workflows, and receives community assistance and frequent database upgrades. For security experts performing web security assessments, it's a useful tool that helps with vulnerability discovery and mitigation to improve overall security posture.

Proof of concept:



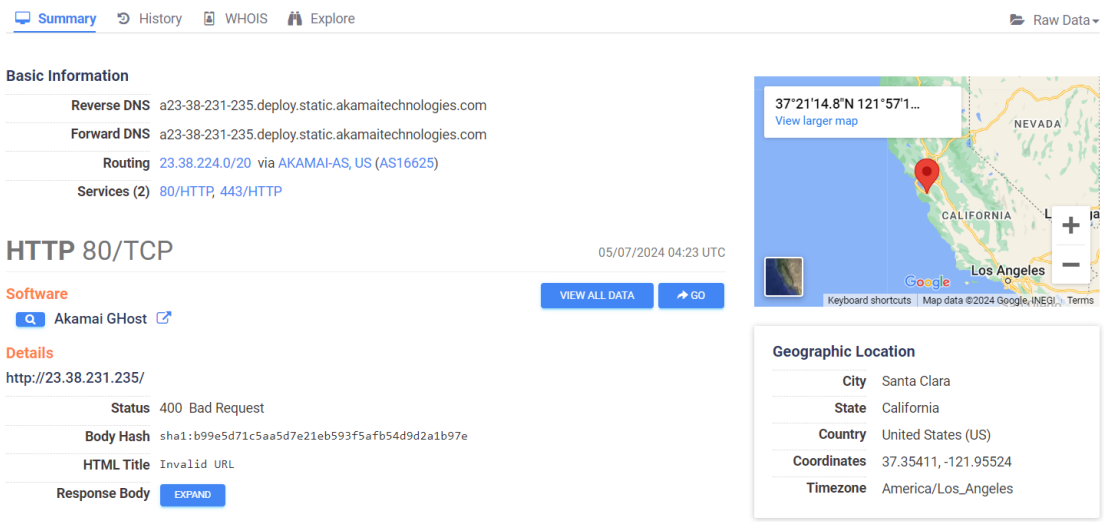
Public devices enumeration

- Censys

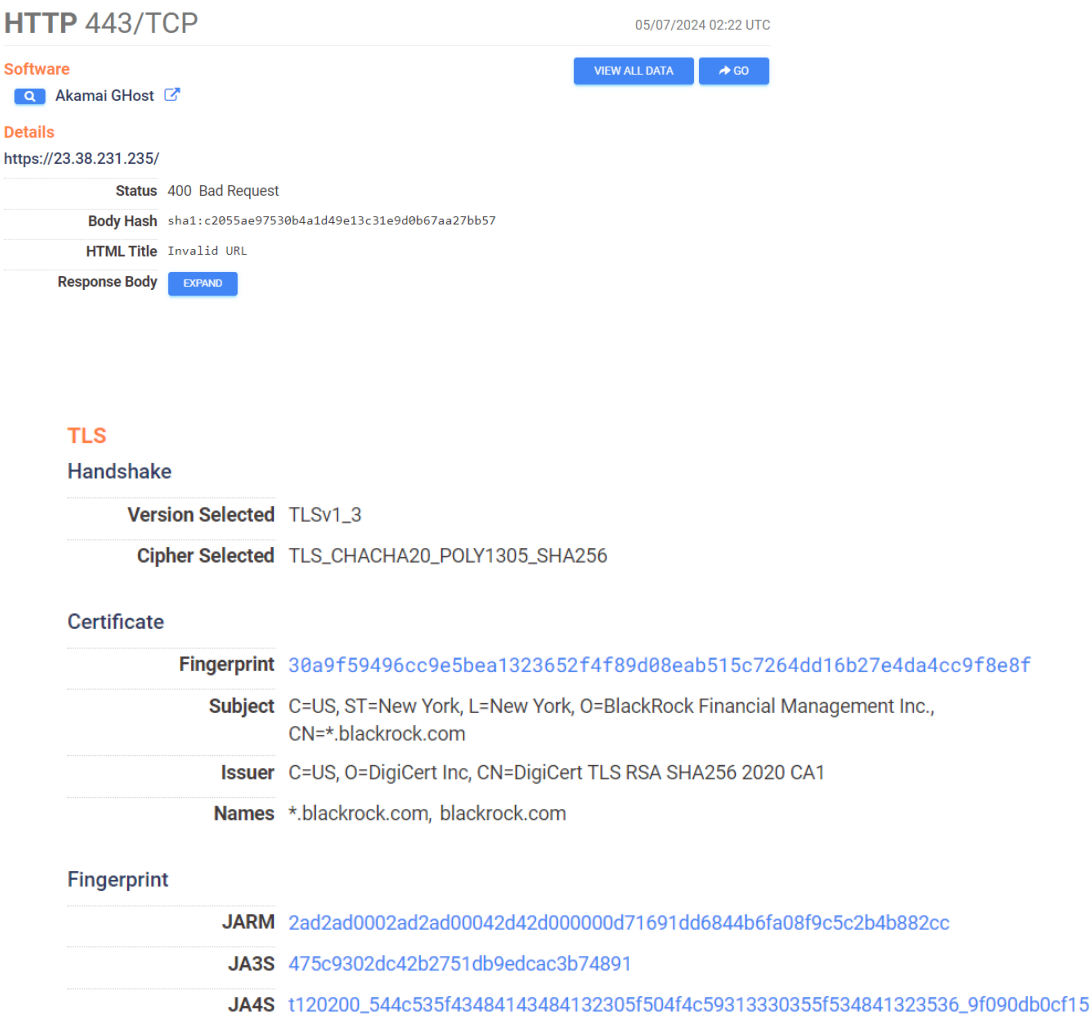
Censys is a potent internet scanning tool that scans and monitors devices, networks, and services online all the time. It performs thorough scans,

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:



Proof of concept:

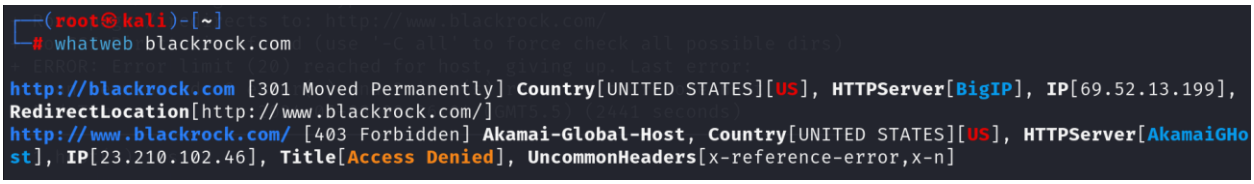


- Whatweb

WhatWeb is a powerful open-source website scanner that specializes in identifying the technologies utilized by websites. It excels in detecting web servers, frameworks, content management systems (CMS), programming languages, and more through its fingerprinting and HTTP header analysis capabilities. Security professionals leverage WhatWeb during reconnaissance and vulnerability

assessments to gather insights into target websites' technology stacks, assess security risks associated with detected technologies, and identify potential vulnerabilities or outdated components. Its scriptable and customizable nature, along with integration options, makes it a valuable tool for automating scanning workflows and enhancing security assessments.

Proof of concept:



- Whois

A key networking tool and protocol, whois is used to query databases and obtain data on IP addresses, domain names, and autonomous system numbers (ASNs). It offers information about domain registration, owner contact details, DNS information, IP address allocation, and ASN ownership. Whois is frequently used for domain research, network problems, IP geolocation, abuse investigations, and domain registration questions by network administrators, cybersecurity experts, domain registrars, and law enforcement organizations. Due to privacy concerns, registrars have started offering privacy services that conceal personal information from view in public Whois data. While Whois provides insightful information, users should be aware of potential limits across multiple Whois servers, privacy concerns, and data accuracy.

Proof of concept:

```
┌──(root㉿kali)-[~]
└─# whois blackrock.com

   Domain Name: BLACKROCK.COM
   Registry Domain ID: 2496064_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.corporatedomains.com
   Registrar URL: http://cscdbs.com
   Updated Date: 2023-03-13T10:27:01Z
   Creation Date: 1997-11-26T05:00:00Z
   Registry Expiry Date: 2024-11-25T05:00:00Z
   Registrar: CSC Corporate Domains, Inc.
   Registrar IANA ID: 299
   Registrar Abuse Contact Email: domainabuse@cscglobal.com
   Registrar Abuse Contact Phone: 8887802723
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
   Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
   Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
   Name Server: NS1.BLACKROCK.COM
   Name Server: NS2.BLACKROCK.COM
   Name Server: NS3.BLACKROCK.COM
   Name Server: NS4.BLACKROCK.COM
   Name Server: NS5.BLACKROCK.COM
   Name Server: NS6.BLACKROCK.COM
   DNSSEC: signedDelegation
```

```
Registry Domain ID: 2496064_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.corporatedomains.com
Registrar URL: www.cscprotectsbrands.com
Updated Date: 2023-03-10T13:04:58Z
Creation Date: 1997-11-26T00:00:00Z
Registrar Registration Expiration Date: 2024-11-25T05:00:00Z
Registrar: CSC CORPORATE DOMAINS, INC.
Sponsoring Registrar IANA ID: 299
Registrar Abuse Contact Email: domainabuse@cscglobal.com
Registrar Abuse Contact Phone: +1.8887802723
Domain Status: clientTransferProhibited http://www.icann.org/epp#clientTransferProhibited
Domain Status: serverDeleteProhibited http://www.icann.org/epp#serverDeleteProhibited
Domain Status: serverTransferProhibited http://www.icann.org/epp#serverTransferProhibited
Registry Registrant ID:
Registrant Name: Domain Administrator
Registrant Organization: BlackRock, Inc.
Registrant Street: 50 Hudson Yards
Registrant City: New York
Registrant State/Province: NY
Registrant Postal Code: 10001
Registrant Country: US
Registrant Phone: +1.2128105300
Registrant Phone Ext:
Registrant Fax: +1.6463109594
Registrant Fax Ext:
```

Proof of concept:

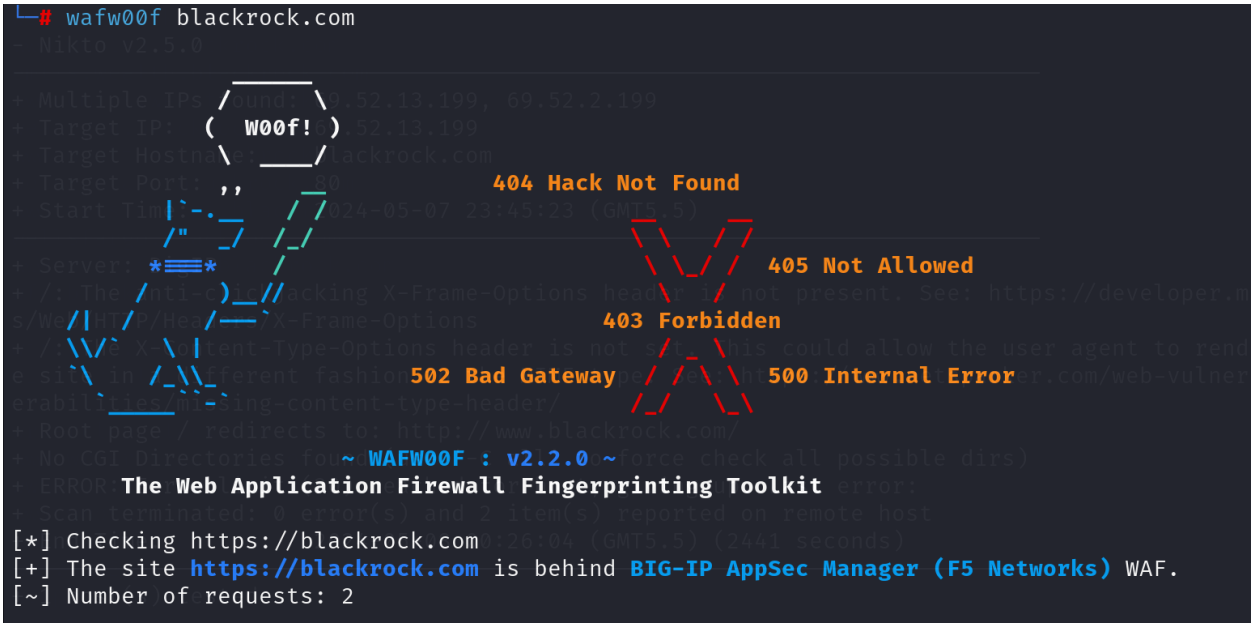
```
Admin City: New York
Admin State/Province: NY
Admin Postal Code: 10001
Admin Country: US
Admin Phone: +1.2128105300
Admin Phone Ext:
Admin Fax: +1.6463109594
Admin Fax Ext:
Admin Email: DNRAdmin@Blackrock.com
Registry Tech ID:
Tech Name: Domain Administrator
Tech Organization: BlackRock, Inc.
Tech Street: 50 Hudson Yards
Tech City: New York
Tech State/Province: NY
Tech Postal Code: 10001
Tech Country: US
Tech Phone: +1.2128105300
Tech Phone Ext:
Tech Fax: +1.6463109594
Tech Fax Ext:
Tech Email: DNRAdmin@Blackrock.com
Name Server: ns2.blackrock.com
Name Server: ns4.blackrock.com
Name Server: ns1.blackrock.com
```

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:



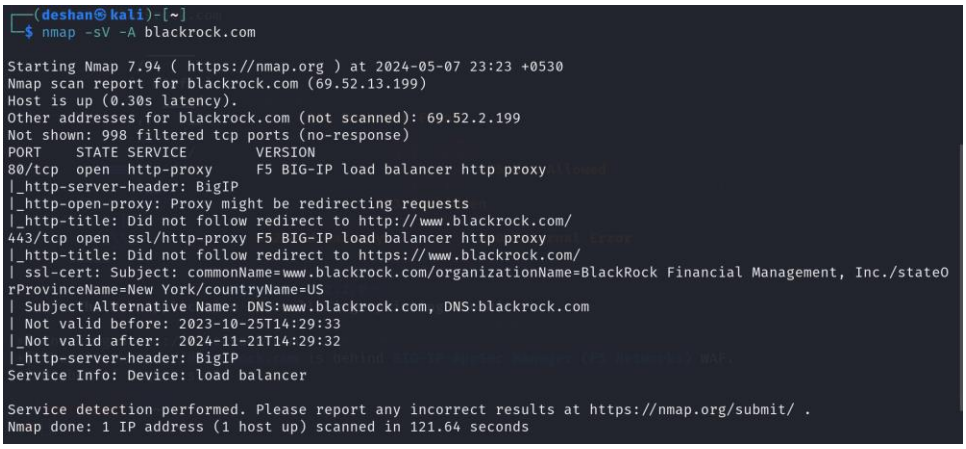
Find open ports.

- 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:



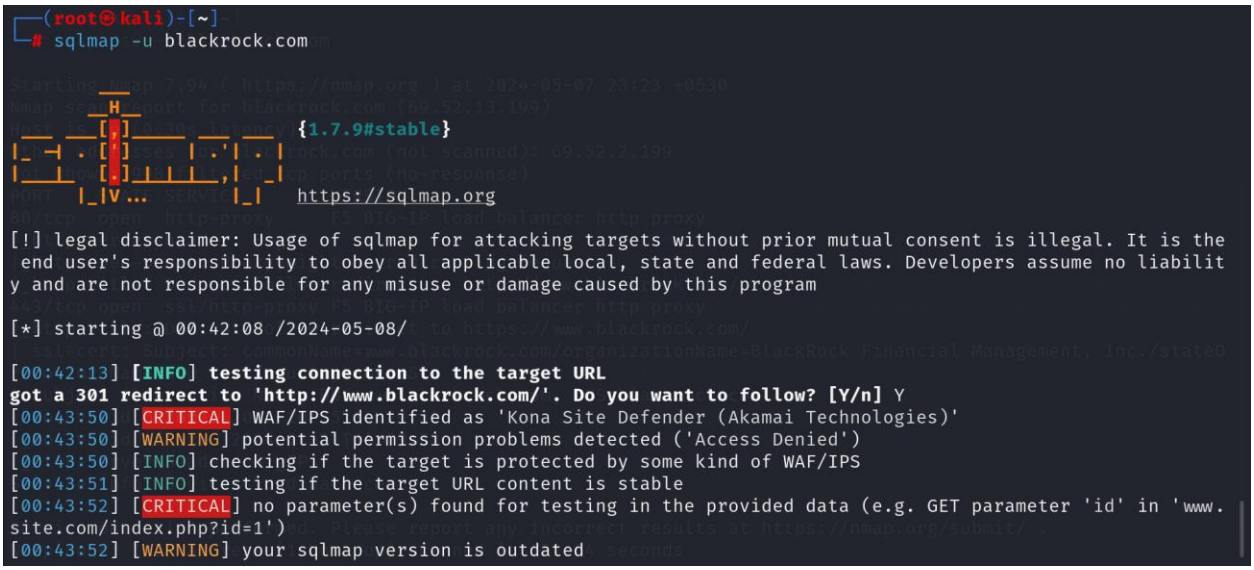
Exploitation

- Sqlmap

Web applications with SQL injection vulnerabilities can be automatically found and exploited with SQLMap, an open-source penetration testing tool. It employs a number of detecting methods, supports a number of 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.

Proof of concept:



vulnerability analysis phase

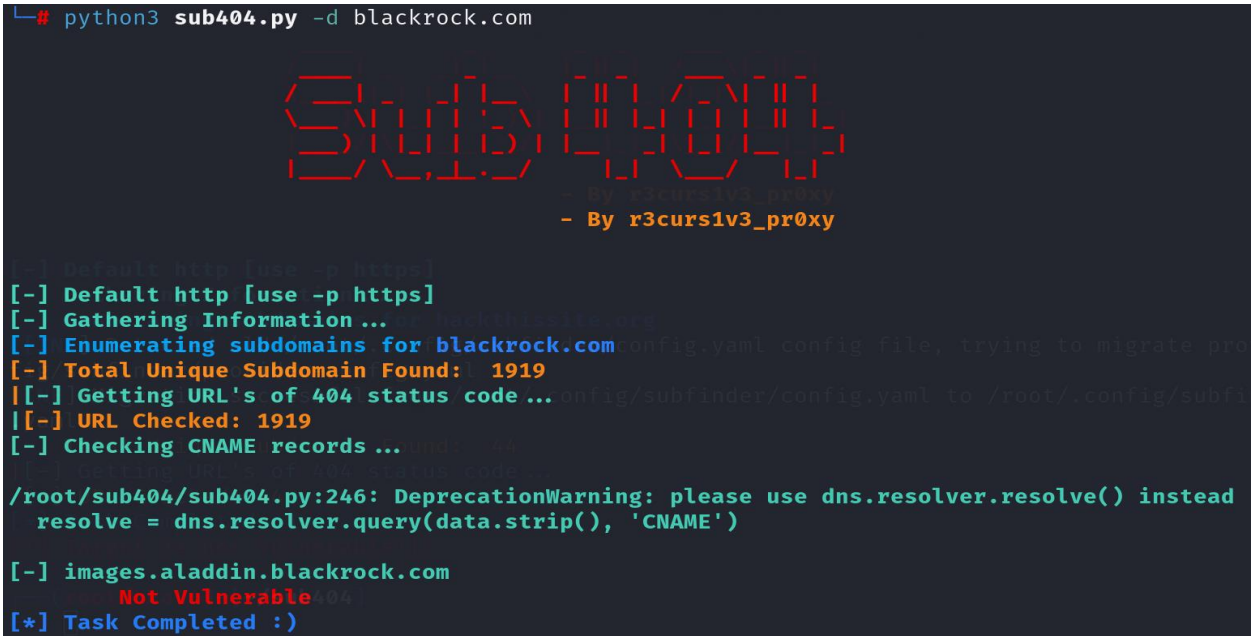
Targeted Domain: - http://blackrock.com

I used tools like sub404, ZAP to process and catch bugs and vulnerabilities based on OWASP top 10.

Sub 404 used to Test any subdomain takeovers that happen.

I’m testing my target domain indrive.com to find any vulnerabilities in their subdomains. After this test no vulnerabilities are found in these subdomains.

Proof of concept:



- ZAP (zed attack proxy)

OWASP created the open-source Zed Attack Proxy (ZAP) tool for online application security testing. Users can examine and alter HTTP/HTTPS communication between web browsers and programs by using it as an intercepting proxy. ZAP provides spidering to map application structures in addition to active and passive scanning for vulnerabilities like SQL injection and XSS. It facilitates input validation testing with fuzzing, controls sessions for authentication verification, and offers automation via scripting and APIs. ZAP helps with risk

management and safe application development by providing thorough reports on vulnerabilities found.

Vulnerability title

CSP: Wildcard Directive

- Vulnerability description

"CSP: Wildcard Directive" refers to using wildcards in Content Security Policy (CSP) directives to specify allowed content sources for web applications. The wildcard (`*`) can be used to allow content from any source, but it's important to use it judiciously due to security concerns. For instance, `script-src *` allows scripts from any origin, increasing potential security risks. Web developers should carefully craft CSP policies to allow content only from trusted sources necessary for their application's functionality, balancing security and functionality.

- Impact assessment

The impact assessment of CSP wildcard directives involves evaluating the security risks, compliance implications, and mitigation strategies associated with using wildcard (*) directives in Content Security Policy (CSP) configurations for web applications. Key considerations include assessing security vulnerabilities such as XSS attacks, conducting risk analysis for potential breaches, ensuring alignment with security standards and regulations, rigorous testing and validation, developing

mitigation strategies, educating stakeholders, and establishing monitoring and response mechanisms. Through a comprehensive assessment, organizations can enhance web application security, mitigate risks, and maintain compliance with industry standards.

- Affected components

The security and operation of online applications can be greatly impacted by the usage of wildcard (*) directives in Content Security Policy (CSP) setups. An overview of the impacted elements and important factors is provided below:

1. information Sources: By affecting the loading of scripts, stylesheets, fonts, pictures, and media from any source, wildcard directives may expose the program to unapproved or harmful information.

2. Security Risks: Because more content sources are permitted, there is a higher chance of data injection vulnerabilities, Cross-Site Scripting (XSS) attacks, and unauthorized content execution.

3. Standards and Compliance: In order to prevent legal issues and compliance infractions, wildcard usage must be in line with industry standards, security best practices, and regulatory requirements.

4. Risk Assessment: Determine the possibility and consequences of security lapses brought on by wildcard directives, such as data loss and harm to one's reputation.

5. Mitigation Strategies: To lower security risks and adjust to changing threats, use scope restrictions, nonce/hashes for inline material, and regular policy reviews.

6. Testing and Validation: * To find and fix vulnerabilities brought about by wildcard directives, do comprehensive security testing, vulnerability scanning, and manual audits.

7. Monitoring and Reaction: To identify and address CSP breaches, security events, and illegal content loads, establish ongoing monitoring, recording, and incident response protocols.

Organizations can enhance web application security and mitigate the risks associated with wildcard directives in CSP configurations by taking these factors into account and putting the right mitigation techniques in place.

- How to mitigate?

To mitigate the risks associated with wildcard (*) directives in Content Security Policy (CSP) configurations, follow these key strategies:

1. Limit Wildcard Usage:Specify wildcard directives for specific content types or trusted domains/subdomains to reduce the attack surface.

2. Nonce and Hashes: Use CSP nonces or hashes to ensure that only trusted inline content is executed, reducing the risk of unauthorized content.

3. Strict CSP Policies: Set strict `default-src` directives and initially deploy CSP in "report-only" mode to monitor policy violations before enforcing restrictions.

4. Regular Policy Reviews: Conduct regular reviews of CSP policies to align with evolving security requirements and mitigate emerging threats.

5. Security Testing: Use automated vulnerability scanning tools to detect and remediate CSP-related vulnerabilities, including wildcard misuse.

6. Education and Training: Educate developers on secure CSP configuration practices, risks associated with wildcards, and effective implementation of nonce/hashes.

7. Monitoring and Incident Response: Enable logging for CSP violations, monitor reports for potential incidents, and have an incident response plan to address violations promptly.

Implementing these strategies helps organizations reduce security risks, maintain compliance, and enhance overall web application security when using wildcard directives in CSP configurations.

Proof of concept:

http://blackrock.com (3)

CSP: Wildcard Directive (1)

▾ GET http://blackrock.com

Alert tags	▪ OWASP_2021_A05 ▪ OWASP_2017_A06
Alert description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) Cross Site Scripting (XSS), and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
Other info	The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: script-src, style-src, img-src, connect-src, frame-src, font-src, media-src, manifest-src, worker-src, form-action The directive(s): form-action are among the directives that do not fallback to default-src, missing/excluding them is the same as allowing anything.

Proof of concept:

Request

▾ Request line and header section (226 bytes)

```
GET http://blackrock.com HTTP/1.1
host: blackrock.com
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
pragma: no-cache
cache-control: no-cache
```

▾ Request body (0 bytes)

Response

▸ Status line and header section (2848 bytes)

▸ Response body (71176 bytes)

Parameter

Content-Security-Policy

Evidence	`default-src https://www.blackrock.com/QkrzXjBCwZBgf/lXbXVW2wae/n3lg/uOarhc6VzQLE/UR4dAQ/CCp/MPXFMayA 'nonce-840d1c5413fbf38d224539f615e76c72' https:; font-src https: data:; img-src https: data:; base-uri 'self'; object-src 'self'; media-src https: blob:; child-src https: blob:; worker-src https: blob:; frame-ancestors 'self' https://*.blackrock.com https://*.ishares.com; style-src https: 'unsafe-inline'; script-src 'nonce-840d1c5413fbf38d224539f615e76c72' https: 'unsafe-eval' 'nonce-9PzjMV3FNtjzUVfnWHgPDg==';`
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.

- Absence of Anti-CSRF Tokens

Proof of concept:

▾ GET http://blackrock.com

Alert tags	▪ OWASP_2021_A01 ▪ WSTG-v42-SESS-05 ▪ OWASP_2017_A05
Alert description	No Anti-CSRF tokens were found in a HTML submission form. A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf. CSRF attacks are effective in a number of situations, including: * The victim has an active session on the target site. * The victim is authenticated via HTTP auth on the target site. * The victim is on the same local network as the target site.

Proof of concept:

	CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.
Other info	No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "autocompleteUrl" "doTickerSearch" "featuredSearchTitle" "featuredSearchUrl" "generalAutocompleteTitle" "generalAutocompleteUrl" "glsDocumentsTitle" "glsDocumentsUrl" "isVi20Search" "productAutocompleteTitle" "productAutocompleteUrl" "searchText" "seeAll" "submit" "summaryAutocompleteTitle" "summaryAutocompleteUrl" "unifiedAutocomplete" "videoSearchTitle" "videoSearchUrl"].

Request	▾ Request line and header section (226 bytes)
	
	`GET http://blackrock.com HTTP/1.1`
	`host: blackrock.com`
	`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`
	`pragma: no-cache`
	`cache-control: no-cache`
	
	▾ Request body (0 bytes)

Response	▸ Status line and header section (2848 bytes) ▸ Response body (71176 bytes)
Evidence	`<form id="searchForm" action="/corporate/search /summary-search-results" class="siteSearch" role="search">`

Proof of concept:

Solution

Phase: Architecture and Design

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

For example, use anti-CSRF packages such as the OWASP CSRFGuard.

Phase: Implementation

Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script.

Phase: Architecture and Design

Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330).

Note that this can be bypassed using XSS.

Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation.

Note that this can be bypassed using XSS.

Use the ESAPI Session Management control.

This control includes a component for CSRF.

Do not use the GET method for any request that triggers a state change.

Phase: Implementation

Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.

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

Proof of concept:

▾ GET http://blackrock.com

Alert tags	▪ OWASP_2021_A05 ▪ OWASP_2017_A06
Alert description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) Cross Site Scripting (XSS), and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
Other info	script-src includes unsafe-eval.

Request	▾ Request line and header section (226 bytes)
	`GET http://blackrock.com HTTP/1.1`
	`host: blackrock.com`
	`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`
	`pragma: no-cache`
	`cache-control: no-cache`
	▸ Request body (0 bytes)

Response	▸ Status line and header section (2848 bytes) ▸ Response body (71176 bytes)
Parameter	`Content-Security-Policy`
Evidence	`default-src https://www.blackrock.com/QkrzXjBCwZBgf/lXbXVW2wae/n3lg/uOarhc6VzQLE/UR4dAQ/CCp/MPXFMayA 'nonce-840d1c5413fbf38d224539f615e76c72' https:; font-src https: data:; img-src https: data:; base-uri 'self'; object-src 'self'; media-src https: blob:; child-src https: blob:; worker-src https: blob:; frame-ancestors 'self' https://*.blackrock.com https://*.ishares.com; style-src https: 'unsafe-inline'; script-src 'nonce-840d1c5413fbf38d224539f615e76c72' https: 'unsafe-eval' 'nonce-9PzjMV3FNtjzUVfnWHgPDg==';`
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.

CSP: style-src unsafe-inline

Proof of concept:

▾ GET http://blackrock.com

Alert tags	▪ OWASP_2021_A05 ▪ OWASP_2017_A06
Alert description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) Cross Site Scripting (XSS), and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
Other info	style-src includes unsafe-inline.

Request	▾ Request line and header section (226 bytes)
	
	`GET http://blackrock.com HTTP/1.1`
	`host: blackrock.com`
	`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`
	`pragma: no-cache`
	`cache-control: no-cache`
	
	▾ Request body (0 bytes)

Response	▸ Status line and header section (2848 bytes) ▸ Response body (71176 bytes)
Parameter	`Content-Security-Policy`
Evidence	`default-src https://www.blackrock.com/QkrzXjBCwZBgf/lXbXVW2wae/n3lg/uOarhc6VzQLE/UR4dAQ/CCp/MPXFMayA 'nonce-840d1c5413fbf38d224539f615e76c72' https:; font-src https: data:; img-src https: data:; base-uri 'self'; object-src 'self'; media-src https: blob:; child-src https: blob:; worker-src https: blob:; frame-ancestors 'self' https://*.blackrock.com https://*.ishares.com; style-src https: 'unsafe-inline'; script-src 'nonce-840d1c5413fbf38d224539f615e76c72' https: 'unsafe-eval' 'nonce-9PzjMV3FNtjzUVfnWHgPDg==';`
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.

- **Hidden File Found**

Proof of concept:

Hidden File Found (1)

▾ GET http://blackrock.com/.hg

Alert tags	▪ OWASP_2021_A05 ▪ WSTG-v42-CONF-05 ▪ OWASP_2017_A06
Alert description	A sensitive file was identified as accessible or available. This may leak administrative, configuration, or credential information which can be leveraged by a malicious individual to further attack the system or conduct social engineering efforts.
Request	▾ Request line and header section (230 bytes) GET http://blackrock.com/.hg HTTP/1.1 host: blackrock.com 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 pragma: no-cache cache-control: no-cache ▸ Request body (0 bytes)
Response	▾ Status line and header section (132 bytes) HTTP/1.0 301 Moved Permanently Location: http://www.blackrock.com/.hg Server: BigIP Connection: Keep-Alive Content-Length: 0 ▾ Response body (0 bytes)
Evidence	HTTP/1.0 301 Moved Permanently
Solution	Consider whether or not the component is actually required in production, if it isn't then disable it. If it is then ensure access to it requires appropriate authentication and authorization, or limit exposure to internal systems or specific source IPs, etc.

- **Cookie without SameSite Attribute**

Proof of concept:

Cookie without SameSite Attribute (1)

▾ GET http://blackrock.com

Alert tags	▪ OWASP_2021_A01 ▪ WSTG-v42-SESS-02 ▪ OWASP_2017_A05
Alert description	A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.

Request	▾ Request line and header section (226 bytes)
	`GET http://blackrock.com HTTP/1.1`
	`host: blackrock.com`
	`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`
	`pragma: no-cache`
	`cache-control: no-cache`
	▾ Request body (0 bytes)

Response	▸ Status line and header section (2848 bytes) ▸ Response body (71176 bytes)
Parameter	`STICKY_SESSION_COOKIE_BLK_CORP01_LIVE`
Evidence	`Set-Cookie: STICKY_SESSION_COOKIE_BLK_CORP01_LIVE`
Solution	Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.

Conclusion

The document outlines a thorough security assessment and vulnerability analysis conducted on the http://blackrock.com domain, focusing on key

areas such as information gathering, reconnaissance, exploitation, and mitigation strategies. Various tools like Recon-ng, Nslookup, Sublist3r, Uniscan, Dnsrecon, Nikto, Nmap, sqlmap, and others were employed to identify potential security weaknesses and vulnerabilities. OWASP top 10 vulnerabilities were analyzed, including Broken Access Control, Cryptographic Failures, Injection, Insecure Design, and more. Mitigation strategies such as limiting wildcard usage in CSP configurations, implementing strict policies, and conducting regular security testing were recommended to address identified risks. Proof of concepts were also provided to demonstrate vulnerabilities like Anti-CSRF Token Absence, CSP Configuration Issues, Hidden Files, and Cookie Security Issues. Overall, the assessment highlights the importance of proactive security measures to safeguard web applications against potential threats and attacks.