VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

FACULTY OF FUNDAMENTAL SCIENCES

DEPARTMENT OF INFORMATION SYSTEMS

RECONNAISSANCE

Ethical Hacking Techniques

Prepared by: Simonas Riška

Checked by: lect. [redacted]

VILNIUS 2025

WHOIS

WHOIS is a widely used query and response protocol designed to access databases that store information about registered internet resources, for example, domain names, but it can also provide details about IP address blocks and autonomous systems.

Each WHOIS record typically contains domain name, registrar (company where the domain was registered), registrant contact (person or organization that registered the domain), administrative contact (the person responsible for managing the domain), technical contact (the person handling technical issues related to the domain), creation and expiration dates (when the domain was registered and when it's set to expire), name servers (servers that translate the domain name into an IP address).

WHOIS offers insights into the target organization's digital footprint and potential vulnerabilities, for example identifying key personnel (WHOIS records often reveal the names, email addresses, and phone numbers of individuals responsible for managing the domain and later on this information can be leveraged for social engineering attacks or to identify potential targets for phishing campaigns), discovering network infrastructure (technical details like name servers and IP addresses provide clues about the target's network infrastructure, this can help identify potential entry points or misconfigurations), historical data analysis (accessing historical WHOIS records through services can reveal changes in ownership, contact information, or technical details over time, this can be useful for tracking the evolution of the target's digital presence)

I used whois command for digitalocean.com (found it via bug bounty programme):

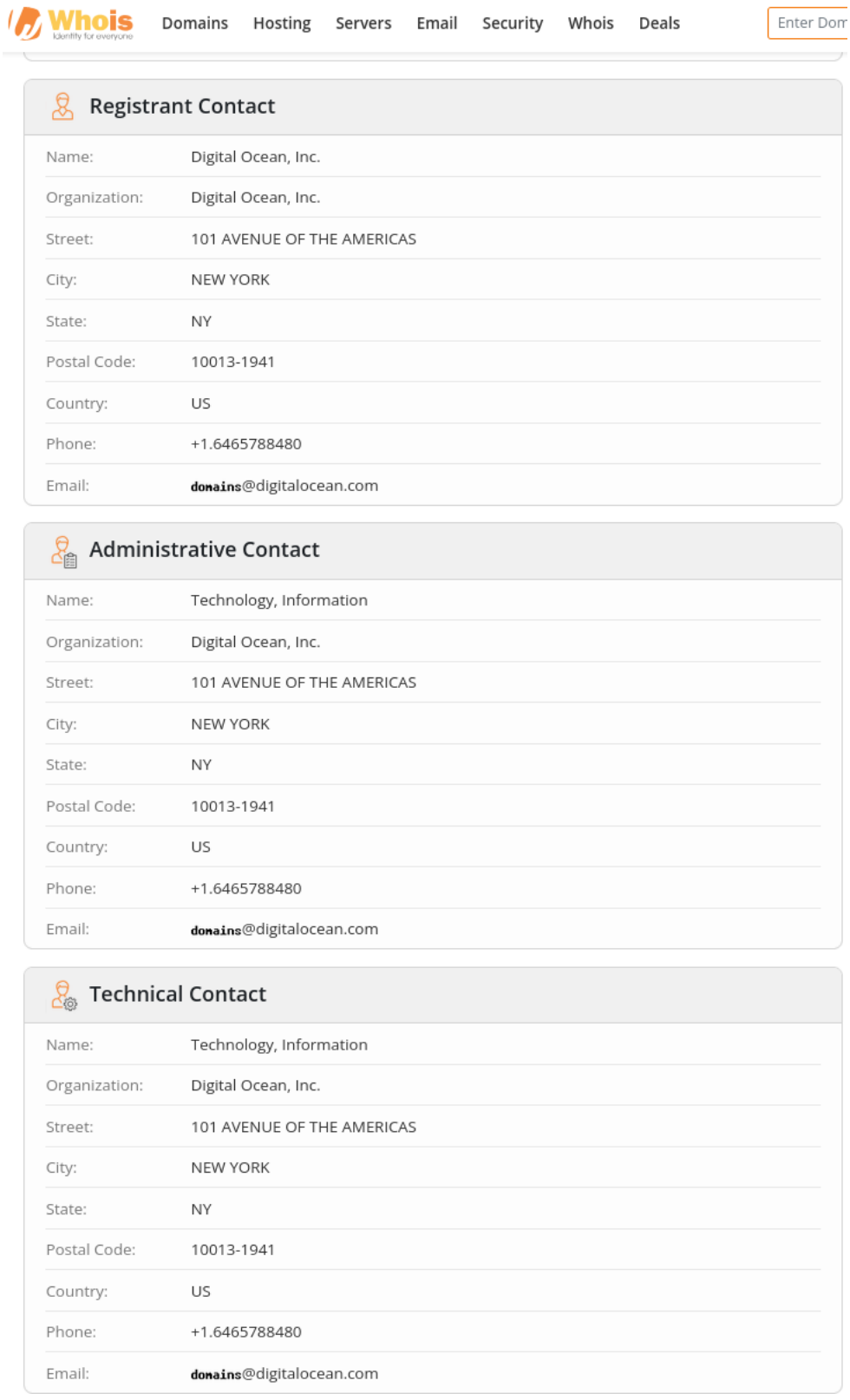
```
┌──(melkiad㉿kali)-[~]
└─$ whois digitalocean.com
   Domain Name: DIGITALOCEAN.COM
   Registry Domain ID: 24753628_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.networksolutions.com
   Registrar URL: http://networksolutions.com
   Updated Date: 2024-11-14T21:43:56Z
   Creation Date: 2000-04-12T10:36:48Z
   Registry Expiry Date: 2028-04-12T10:36:48Z
   Registrar: Network Solutions, LLC
   Registrar IANA ID: 2
   Registrar Abuse Contact Email: domain.operations@web.com
   Registrar Abuse Contact Phone: +1.8777228662
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Name Server: KIM.NS.CLOUDFLARE.COM
   Name Server: WALT.NS.CLOUDFLARE.COM
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2025-03-30T14:27:29Z <<<
```

The WHOIS output for digitalocean.com reveals several key details:

1. Domain Registration:
   Registrar: Network Solutions, LLC
   Creation Date: 2000-04-12T10:36:48Z
   Expiry Date: 2028-04-12T10:36:48Z

   These details indicate that the domain is registered with Network Solutions, LLC and has been active for a considerable period, suggesting its legitimacy and established online presence, the distant expiry date further reinforces its longevity.
2. Domain Status:
   clientTransferProhibited
   This status indicate that the domain is protected against unauthorized transfers on client side. This highlights an emphasis on security and control over the domain.
3. Name Servers:
   KIM.NS.CLOUDFARE.COM, WALT.NS.CLOUDFARE.COM
   These name servers are within the cloudfare.com domain, suggesting that Cloudfare manages its DNS infrastructure.

Also checking online version could be beneficial. It can be accessed via https://www.whois.com/whois/digitalocean.com.



Here, it can be checked domain owner (whois command on Terminal seems sometimes not giving results and just freezing). This information identifies Digital Ocean, Inc. as the organization behind digitalocean.com, also e-mail email contact for this domain. This is consistent with the expectation that digitalocean.com is owned by Digital Ocean, Inc.

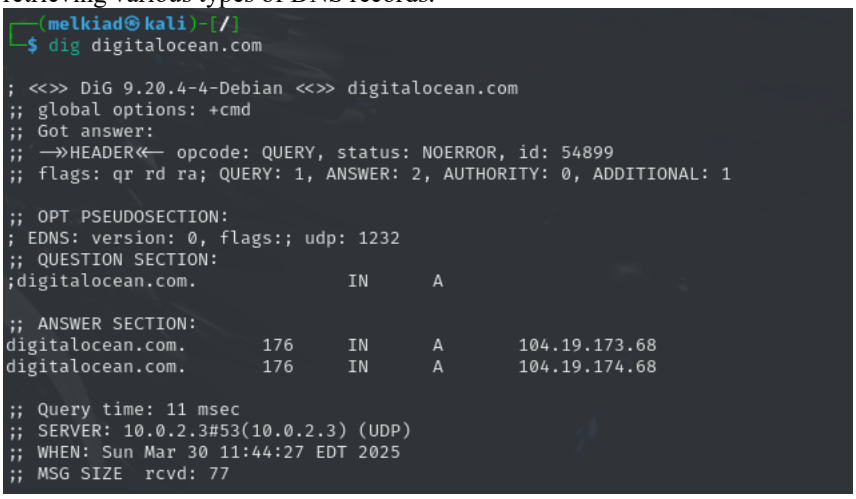
While the WHOIS record provides contact information for domain-related issues, it is not directly helpful in identifying individual employees or specific vulnerabilities. This highlights the need to combine WHOIS data with other reconnaissance techniques to understand the target's digital footprint comprehensively.

DNS

DNS can be leveraged to uncover vulnerabilities and gain access during a penetration test:

- Uncovering Assets: DNS records can reveal a wealth of information, including subdomains, mail servers, and name server records. For instance, a CNAME record pointing to an outdated server (dev.example.com CNAME oldserver.example.net) could lead to a vulnerable system.
- Mapping the Network Infrastructure: I can create a comprehensive map of the target's network infrastructure by analysing DNS data. For example, identifying the name servers (NS records) for a domain can reveal the hosting provider used, while an A record for loadbalancer.example.com can pinpoint a load balancer. This helps you understand how different systems are connected, identify traffic flow, and pinpoint potential choke points or weaknesses that could be exploited during a penetration test.
- Monitoring for Changes: Continuously monitoring DNS records can reveal changes in the target's infrastructure over time. For example, the sudden appearance of a new subdomain (vpn.example.com) might indicate a new entry point into the network, while a TXT record containing a value like _1password=... strongly suggests the organization is using 1Password, which could be leveraged for social engineering attacks or targeted phishing campaigns.

The dig command (Domain Information Groper) is a versatile and powerful utility for querying DNS servers and retrieving various types of DNS records.



This output is the result of a DNS query using the dig command for the domain google.com. The command was executed on a system running DiG version 9.20.4.4-Debian. The output can be broken down into four key sections:

1. Header

   ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54899: This line indicates the type of query (QUERY), the successful status (NOERROR), and a unique identifier (54899) for this specific query.

   ;; flags: qr rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0: This describes the flags in the DNS header:

   qr: Query Response flag - indicates this is a response.
   rd: Recursion Desired flag - means recursion was requested.

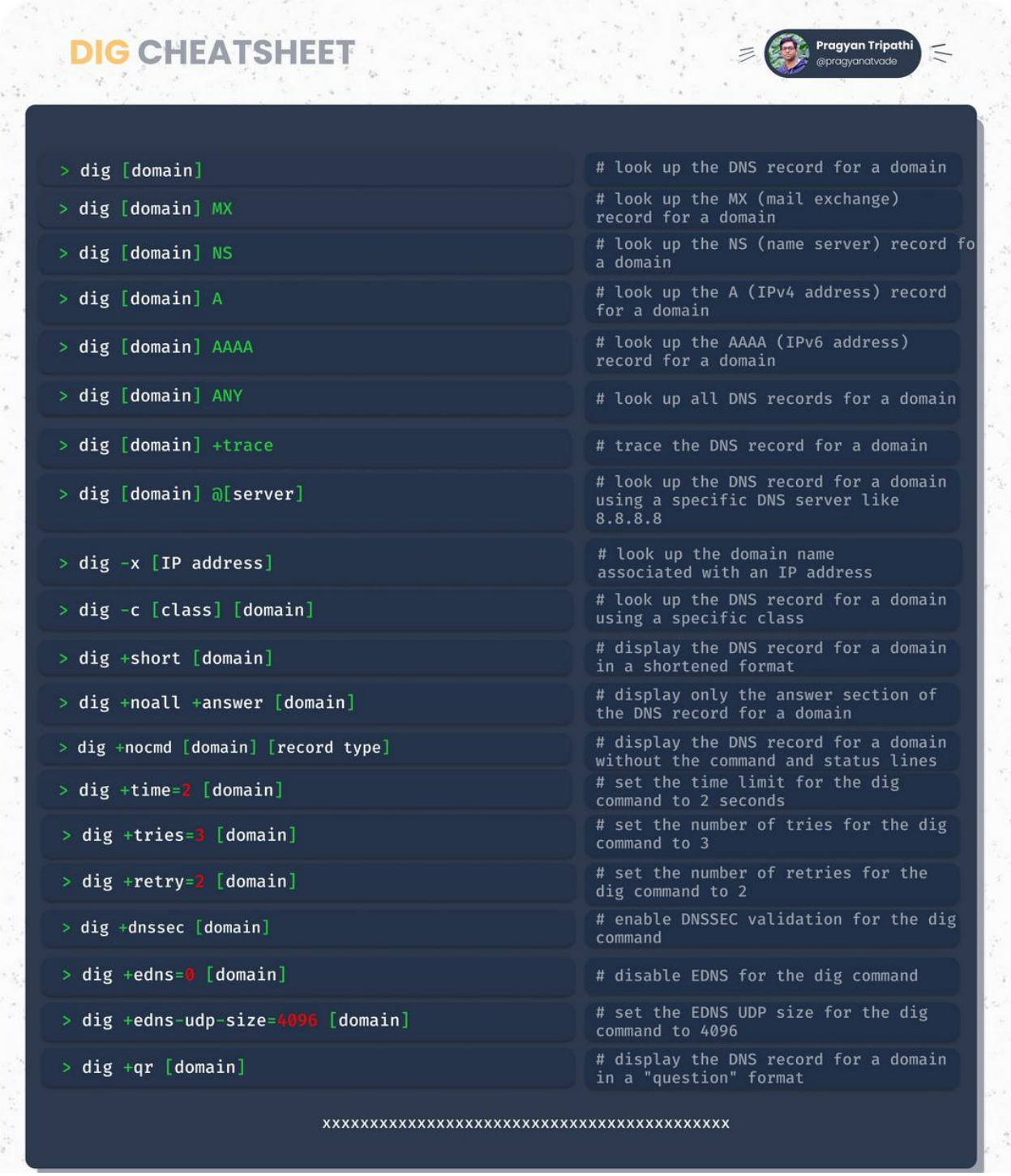
ad: Authentic Data flag - means the resolver considers the data authentic.
The remaining numbers indicate the number of entries in each section of the DNS response: 1 question, 2 answers, 0 authority records, and 1 additional records.

Recursion was supported (?).

2. Question section
;digitalocean.com. IN A: This line specifies the question: "What is the IPv4 address (A record) for digitalocean.com?"

3. Answer section
digitalocean.com. 176 IN A 104.19.173.68 and digitalocean.com 176 IN A 104.19.174.68: These are the answers to the query. It indicates that the IP addresses associated with digitalocean.com are 104.19.173.68 and . 104.19.174.68. The '176' represents the TTL (time-to-live), indicating how long the result can be cached before being refreshed.

4. Footer
;; Query time: 11 msec: This shows the time it took for the query to be processed and the response to be received (11 milliseconds).
;; SERVER: 10.0.2.3#53(10.0.2.3) (UDP): This identifies the DNS server that provided the answer and the protocol used (UDP).
;; WHEN: Sun Mar 30 11:44:27 EDT 2025: This is the timestamp of when the query was made.
;; MSG SIZE rcvd: 77: This indicates the size of the DNS message received (77 bytes).



Source: https://x.com/pragyanatvade/status/1615181182521556992

Subdomains

Subdomains often host valuable information and resources that aren't directly linked from the main website. This can include:

- Development and Staging Environments: Companies often use subdomains to test new features or updates before deploying them to the main site. Due to relaxed security measures, these environments sometimes contain vulnerabilities or expose sensitive information.

- Hidden Login Portals: Subdomains might host administrative panels or other login pages that are not meant to be publicly accessible. Attackers seeking unauthorised access can find these as attractive targets.
- Legacy Applications: Older, forgotten web applications might reside on subdomains, potentially containing outdated software with known vulnerabilities.
- Sensitive Information: Subdomains can inadvertently expose confidential documents, internal data, or configuration files that could be valuable to attackers.

Subdomain enumeration is the process of systematically identifying and listing these subdomains.
Active subdomain enumeration directly interacts with target domain's DNS servers to uncover subdomains with technique like brute-force enumeration, which involves systematically testing a list of potential subdomain names against the target domain. Tools like dnsenum, ffuf, and gobuster can automate this process, using wordlists of common subdomain names or custom-generated lists based on specific patterns.
Passive Subdomain Enumeration relies on external sources of information to discover subdomains without directly querying the target's DNS servers, for example, Certificate Transparency (CT) logs, public repositorie of SSL/TLS certificates - these certificates often include a list of associated subdomains in their Subject Alternative Name (SAN) field, providing a treasure trove of potential targets. Another passive approach involves utilising search engines like Google or DuckDuckGo. By employing specialised search operators (e.g., site:), you can filter results to show only subdomains related to the target domain.
Active enumeration offers more control and potential for comprehensive discovery but can be more detectable. Passive enumeration is stealthier but might not uncover all existing subdomains. Combining both approaches provides a more thorough and effective subdomain enumeration strategy.

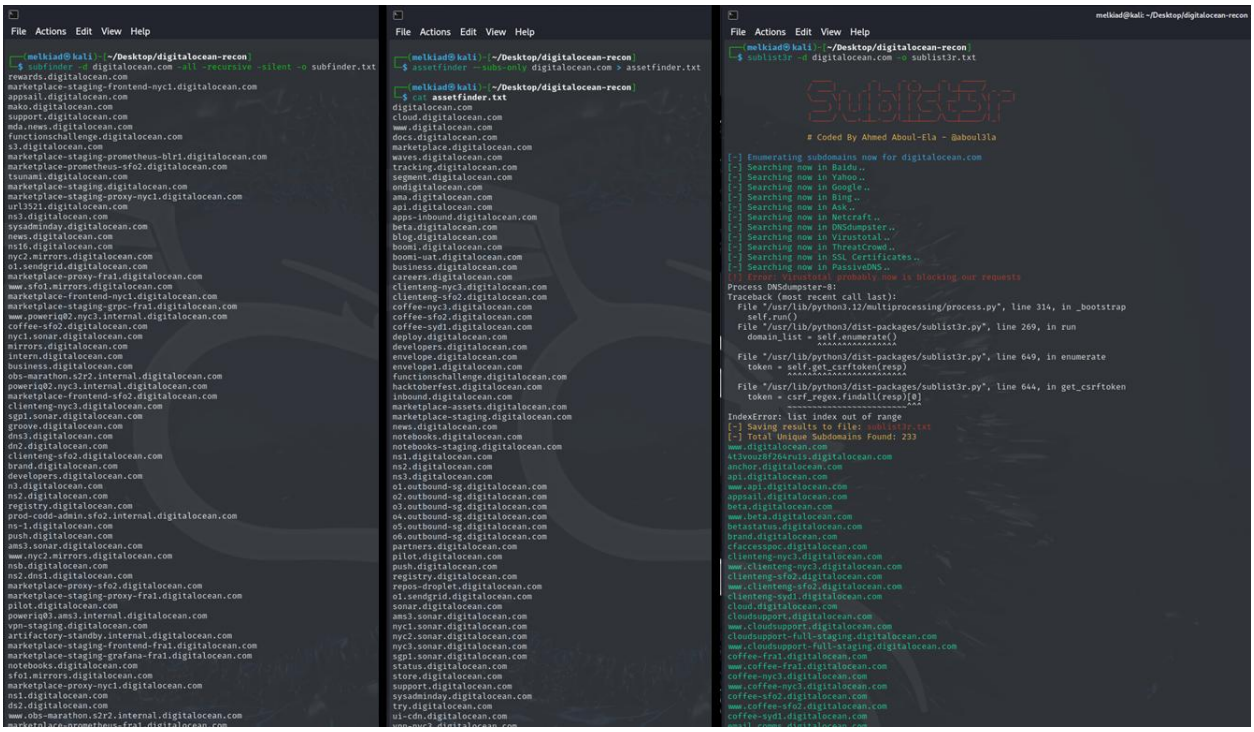
Subdomain bruteforcing

Subdomain Brute-Force Enumeration systematically tests predefined names against the target domain to identify valid subdomains. By using carefully crafted wordlists, you can significantly increase the efficiency and effectiveness of your subdomain discovery efforts.

The process breaks down into four steps:

1. Wordlist Selection: The process begins with selecting a wordlist containing potential subdomain names. These wordlists can be:
    - General-Purpose: Containing a broad range of common subdomain names (e.g., dev, staging, blog, mail, admin, test). This approach is useful when you don't know the target's naming conventions.
    - Targeted: Focused on specific industries, technologies, or naming patterns relevant to the target. This approach is more efficient and reduces the chances of false positives.
    - Custom: You can create your own wordlist based on specific keywords, patterns, or intelligence gathered from other sources.
2. Iteration and Querying: A script or tool iterates through the wordlist, appending each word or phrase to the main domain (e.g., example.com) to create potential subdomain names (e.g., dev.example.com, staging.example.com).
3. DNS Lookup: A DNS query is performed for each potential subdomain to check if it resolves to an IP address. This is typically done using the A or AAAA record type.
4. Filtering and Validation: If a subdomain resolves successfully, it's added to a list of valid subdomains. Further validation steps might be taken to confirm the subdomain's existence and functionality (e.g., by attempting to access it through a web browser).

For this task, I used subfinder, assetfinder and sublist3r.



```
┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ wc -l *.txt
  69 assetfinder.txt
 234 subfinder.txt
 233 sublist3r.txt
 536 total
```

```
┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ cat *.txt | sort -u > subs.txt

┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ wc -l subs.txt
335 subs.txt
```

DNS Zone Transfer

A DNS zone transfer is essentially a wholesale copy of all DNS records within a zone (a domain and its subdomains) from one name server to another. This process is essential for maintaining consistency and redundancy across DNS servers. However, if not adequately secured, unauthorised parties can download the entire zone file, revealing a complete list of subdomains, their associated IP addresses, and other sensitive DNS data.

Awareness of this vulnerability has grown, and most DNS server administrators have mitigated the risk. Modern DNS servers are typically configured to allow zone transfers only to trusted secondary servers, ensuring that sensitive zone data remains confidential. However, misconfigurations can still occur due to human error or outdated practices. This is why attempting a zone transfer (with proper authorisation) remains a valuable reconnaissance technique. Even if unsuccessful, the attempt can reveal information about the DNS server's configuration and security posture.

```
┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ dig ns digitalocean.com

; <<>> DiG 9.20.4-4-Debian <<>> ns digitalocean.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53739
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;digitalocean.com.              IN      NS

;; ANSWER SECTION:
digitalocean.com.       86400   IN      NS      kim.ns.cloudflare.com.
digitalocean.com.       86400   IN      NS      walt.ns.cloudflare.com.

;; Query time: 11 msec
;; SERVER: 10.0.2.3#53(10.0.2.3) (UDP)
;; WHEN: Mon Mar 31 13:37:05 EDT 2025
;; MSG SIZE  rcvd: 96
```

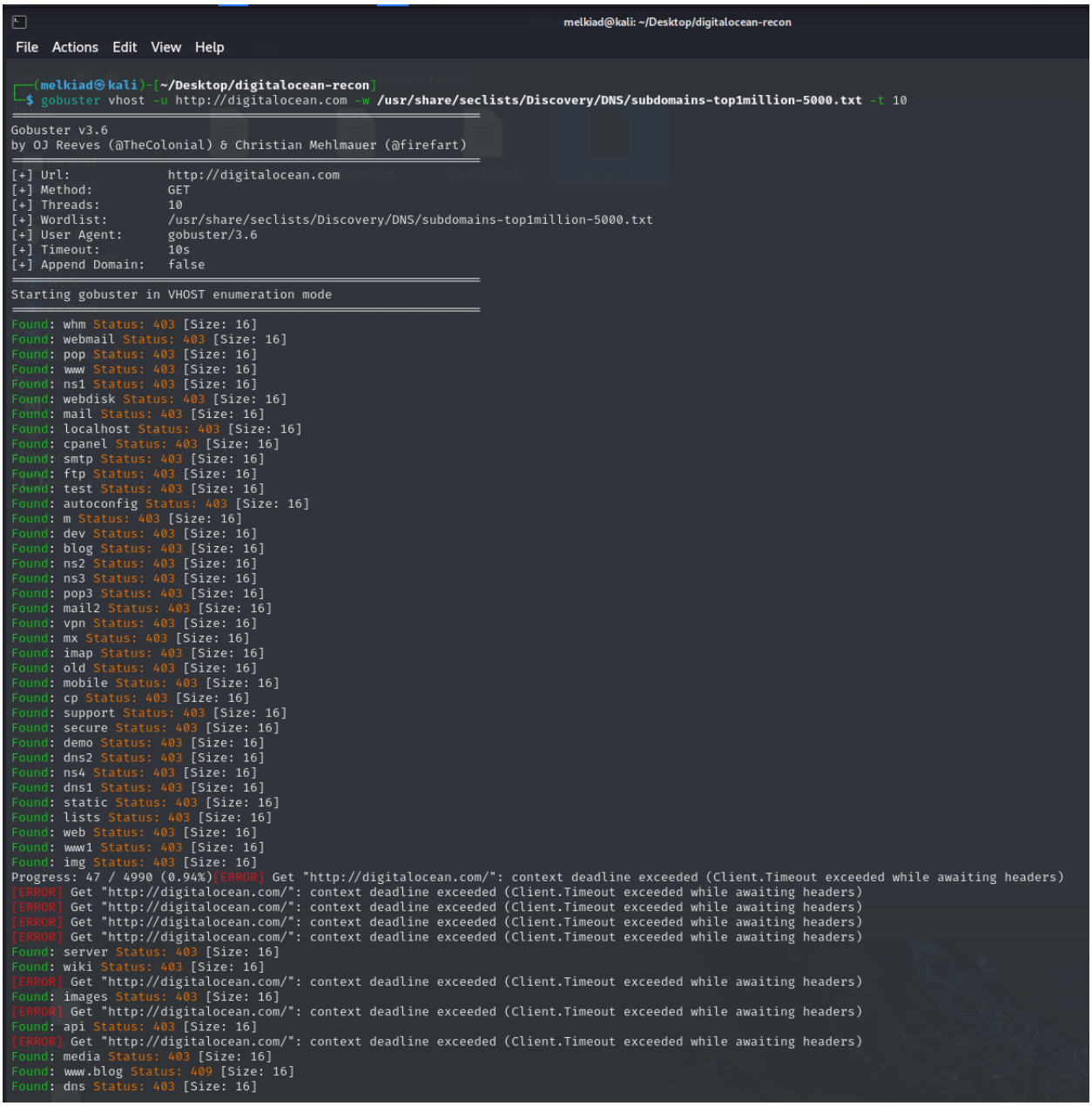
```
┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ dig axfr digitalocean.com @kim.ns.cloudflare.com

; <<>> DiG 9.20.4-4-Debian <<>> axfr digitalocean.com @kim.ns.cloudflare.com
;; global options: +cmd
; Transfer failed.

┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ dig axfr digitalocean.com @walt.ns.cloudflare.com

; <<>> DiG 9.20.4-4-Debian <<>> axfr digitalocean.com @walt.ns.cloudflare.com
;; global options: +cmd
; Transfer failed.
```

This time it failed due to Cloudfare security configuration.

Virtual Hosts

Once the DNS directs traffic to the correct server, the web server configuration becomes crucial in determining how the incoming requests are handled. Web servers like Apache, Nginx, or IIS are designed to host multiple websites or applications on a single server. They achieve this through virtual hosting, which allows them to differentiate between domains, subdomains, or even separate websites with distinct content.

I attempted virtual host enumeration on digitalocean.com using gobuster and a popular wordlist. Since DigitalOcean is behind Cloudflare, I expected uniform responses for both valid and invalid subdomains.
gobuster vhost -u http://digitalocean.com -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -t 10



All discovered vhosts returned status 403 or 530 with the exact same response size [Size: 16]. This indicates that Cloudflare is responding with a generic error page, not the actual origin server. There's no distinguishable difference between real and fake virtual hosts, making this method ineffective against Cloudflare-protected domains.

To verify whether Gobuster was discovering real virtual hosts or just hitting Cloudflare defaults, I manually sent HTTP requests with custom Host headers using curl.
curl -H "Host: whm.digitalocean.com" http://digitalocean.com -i
curl -H "Host: asdf123.digitalocean.com" http://digitalocean.com -i

```
┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ curl -H "Host: whm.digitalocean.com" http://digitalocean.com -i
HTTP/1.1 530
Date: Mon, 31 Mar 2025 18:02:20 GMT
Content-Type: text/plain; charset=UTF-8
Content-Length: 16
Connection: keep-alive
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
Cache-Control: private, max-age=0, no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Expires: Thu, 01 Jan 1970 00:00:01 GMT
Server: cloudflare
CF-RAY: 9291c1d2e9757dfc-VNO

error code: 1016

┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ curl -H "Host: asdf123.digitalocean.com" http://digitalocean.com -i
HTTP/1.1 530
Date: Mon, 31 Mar 2025 18:02:28 GMT
Content-Type: text/plain; charset=UTF-8
Content-Length: 16
Connection: keep-alive
X-Frame-Options: SAMEORIGIN
Referrer-Policy: same-origin
Cache-Control: private, max-age=0, no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Expires: Thu, 01 Jan 1970 00:00:01 GMT
Server: cloudflare
CF-RAY: 9291c2094921bc10-VNO

error code: 1016
```

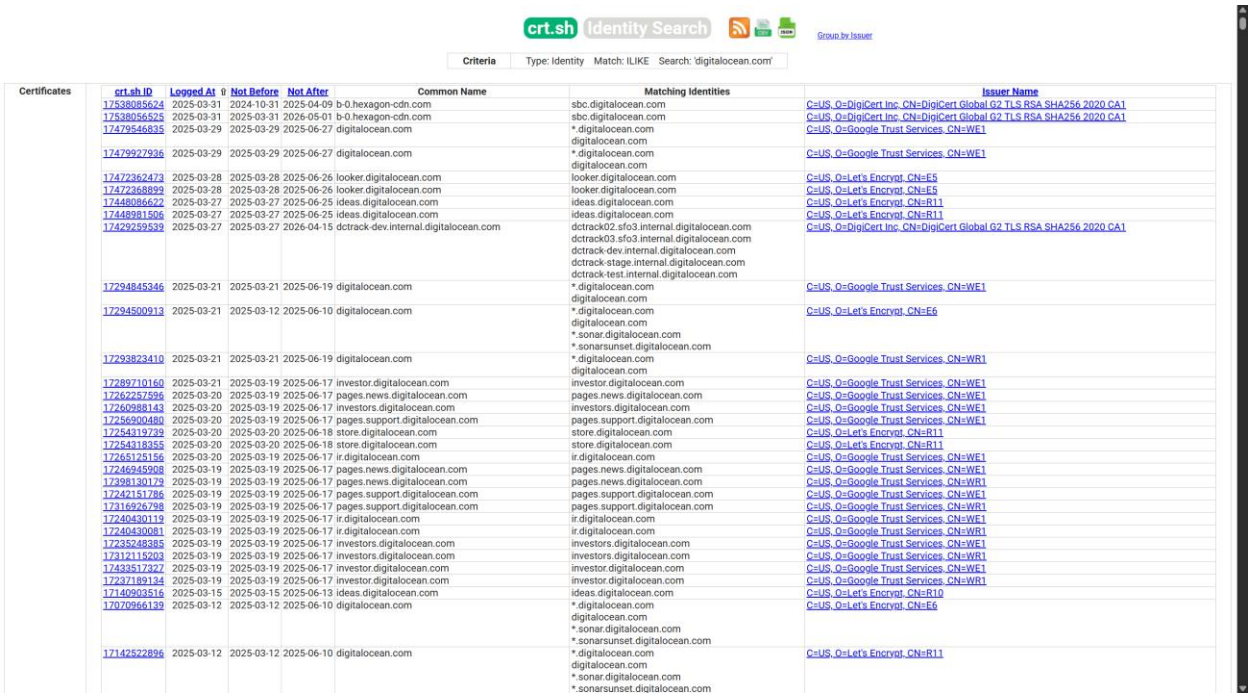
Both requests returned a **530 status code** with Content-Length: 16 and the same error code **1016** from Cloudflare. This confirms that Cloudflare is blocking resolution of unrecognized subdomains with identical error messages, making virtual host fuzzing useless in this case.

VHost enumeration against digitalocean.com is ineffective due to Cloudflare's protection. All subdomain attempts (real or fake) result in the same status code and body size, indicating generic fallback responses from the CDN. In such scenarios, I should pivot to passive subdomain enumeration (e.g., subfinder, crt.sh, amass) and live probing (httpx), rather than relying on vhost fuzzing.

Certificate Transparency Logs

Certificate Transparency (CT) logs are publicly accessible records of all TLS/SSL certificates issued by trusted Certificate Authorities. When a subdomain is included in an SSL certificate (e.g., dev.digitalocean.com), it gets exposed in CT logs — even if it was never meant to be public. This makes CT logs a powerful passive recon source for discovering hidden or internal subdomains without sending any direct traffic to the target.

I started by manually inspecting CT logs using crt.sh. This web interface allows view all certificates issued for subdomains under digitalocean.com.



crt.sh Identity Search

Group by Issuer

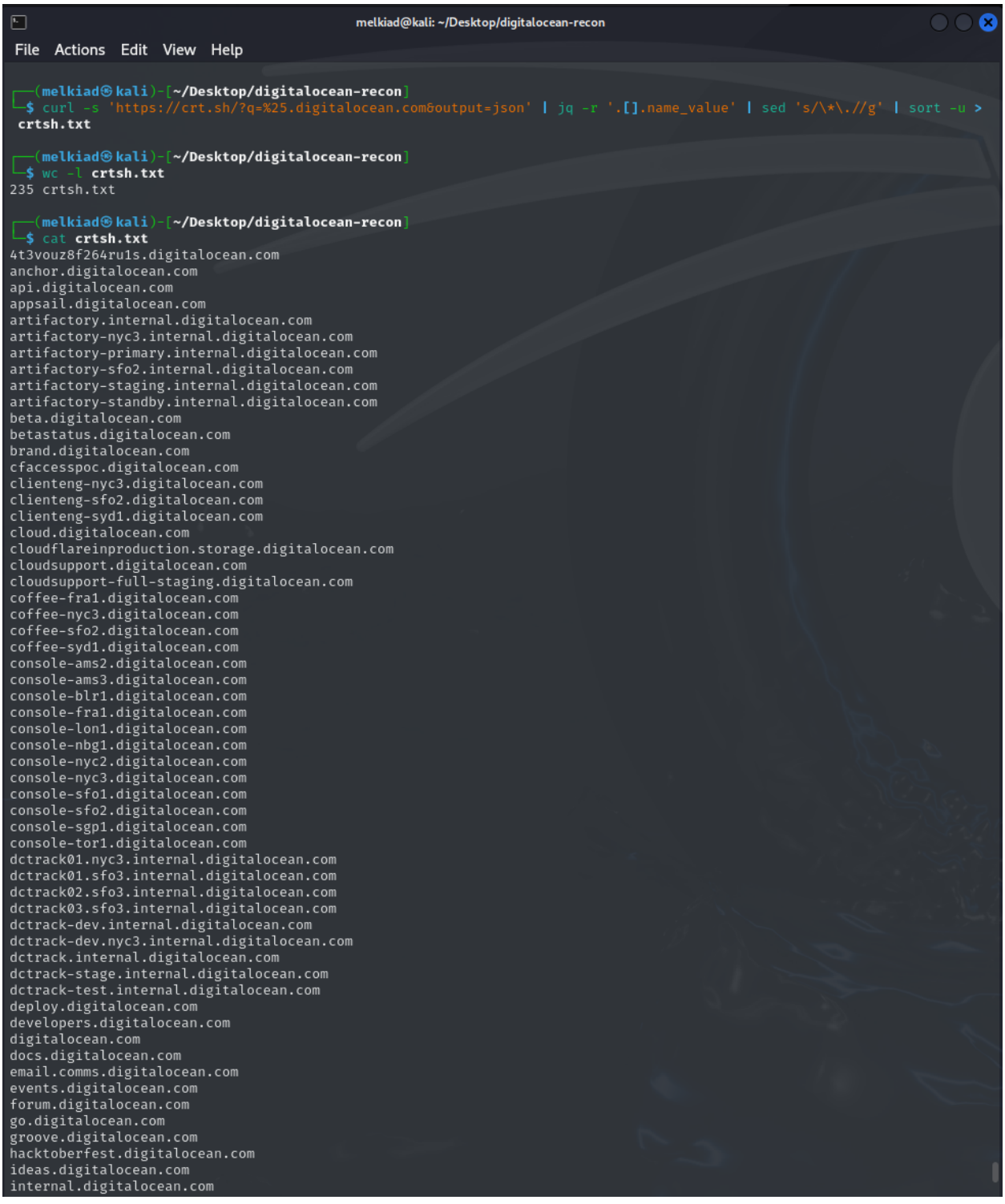
Criteria: Type: Identity Match: ILIKE Search: 'digitalocean.com'

Certificates

crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	Issuer Name
17538085624	2025-03-31	2024-10-31	2025-04-09	b-0.hexagon-cdn.com	sbc.digitalocean.com	C=US, O=DigiCert Inc, CN=DigiCert Global G2 TLS RSA SHA256 2020 CA1
17538056525	2025-03-31	2025-03-31	2026-05-01	b-0.hexagon-cdn.com	sbc.digitalocean.com	C=US, O=DigiCert Inc, CN=DigiCert Global G2 TLS RSA SHA256 2020 CA1
17479546835	2025-03-29	2025-03-29	2025-06-27	digitalocean.com	*.digitalocean.com digitalocean.com	C=US, O=Google Trust Services, CN=WE1
17479927936	2025-03-29	2025-03-29	2025-06-27	digitalocean.com	*.digitalocean.com digitalocean.com	C=US, O=Google Trust Services, CN=WE1
17472362473	2025-03-28	2025-03-28	2025-06-26	looker.digitalocean.com	looker.digitalocean.com	C=US, O=Let's Encrypt, CN=E5
17472368899	2025-03-28	2025-03-28	2025-06-26	looker.digitalocean.com	looker.digitalocean.com	C=US, O=Let's Encrypt, CN=E5
17448086622	2025-03-27	2025-03-27	2025-06-25	ideas.digitalocean.com	ideas.digitalocean.com	C=US, O=Let's Encrypt, CN=R11
17448981506	2025-03-27	2025-03-27	2025-06-25	ideas.digitalocean.com	ideas.digitalocean.com	C=US, O=Let's Encrypt, CN=R11
17429259539	2025-03-27	2025-03-27	2026-04-15	dctrack-dev.internal.digitalocean.com	dctrack02.sfo3.internal.digitalocean.com dctrack03.sfo3.internal.digitalocean.com dctrack-dev.internal.digitalocean.com dctrack-stage.internal.digitalocean.com dctrack-test.internal.digitalocean.com	C=US, O=DigiCert Inc, CN=DigiCert Global G2 TLS RSA SHA256 2020 CA1
17294845346	2025-03-21	2025-03-21	2025-06-19	digitalocean.com	*.digitalocean.com digitalocean.com	C=US, O=Google Trust Services, CN=WE1
17294500913	2025-03-21	2025-03-12	2025-06-10	digitalocean.com	*.digitalocean.com digitalocean.com *.sonar.digitalocean.com *.sonarsunset.digitalocean.com	C=US, O=Let's Encrypt, CN=E6
17293823410	2025-03-21	2025-03-21	2025-06-19	digitalocean.com	*.digitalocean.com digitalocean.com	C=US, O=Google Trust Services, CN=WR1
17289710160	2025-03-21	2025-03-19	2025-06-17	investor.digitalocean.com	investor.digitalocean.com	C=US, O=Google Trust Services, CN=WE1
17262257596	2025-03-20	2025-03-19	2025-06-17	pages.news.digitalocean.com	pages.news.digitalocean.com	C=US, O=Google Trust Services, CN=WE1
17260988143	2025-03-20	2025-03-19	2025-06-17	investors.digitalocean.com	investors.digitalocean.com	C=US, O=Google Trust Services, CN=WE1
17256900480	2025-03-20	2025-03-19	2025-06-17	pages.support.digitalocean.com	pages.support.digitalocean.com	C=US, O=Google Trust Services, CN=WE1
17254319739	2025-03-20	2025-03-20	2025-06-18	store.digitalocean.com	store.digitalocean.com	C=US, O=Let's Encrypt, CN=R11
17254318355	2025-03-20	2025-03-20	2025-06-18	store.digitalocean.com	store.digitalocean.com	C=US, O=Let's Encrypt, CN=R11
17265125156	2025-03-20	2025-03-19	2025-06-17	ir.digitalocean.com	ir.digitalocean.com	C=US, O=Google Trust Services, CN=WE1
17246945908	2025-03-19	2025-03-19	2025-06-17	pages.news.digitalocean.com	pages.news.digitalocean.com	C=US, O=Google Trust Services, CN=WE1
17398130179	2025-03-19	2025-03-19	2025-06-17	pages.news.digitalocean.com	pages.news.digitalocean.com	C=US, O=Google Trust Services, CN=WR1
17242151786	2025-03-19	2025-03-19	2025-06-17	pages.support.digitalocean.com	pages.support.digitalocean.com	C=US, O=Google Trust Services, CN=WE1
17316926798	2025-03-19	2025-03-19	2025-06-17	pages.support.digitalocean.com	pages.support.digitalocean.com	C=US, O=Google Trust Services, CN=WR1
17240430119	2025-03-19	2025-03-19	2025-06-17	ir.digitalocean.com	ir.digitalocean.com	C=US, O=Google Trust Services, CN=WE1
17240430081	2025-03-19	2025-03-19	2025-06-17	ir.digitalocean.com	ir.digitalocean.com	C=US, O=Google Trust Services, CN=WR1
17235248385	2025-03-19	2025-03-19	2025-06-17	investors.digitalocean.com	investors.digitalocean.com	C=US, O=Google Trust Services, CN=WE1
17312115203	2025-03-19	2025-03-19	2025-06-17	investors.digitalocean.com	investors.digitalocean.com	C=US, O=Google Trust Services, CN=WR1
17433517327	2025-03-19	2025-03-19	2025-06-17	investor.digitalocean.com	investor.digitalocean.com	C=US, O=Google Trust Services, CN=WE1
17237189134	2025-03-19	2025-03-19	2025-06-17	investor.digitalocean.com	investor.digitalocean.com	C=US, O=Google Trust Services, CN=WR1
17140903516	2025-03-15	2025-03-15	2025-06-13	ideas.digitalocean.com	ideas.digitalocean.com	C=US, O=Let's Encrypt, CN=R10
17070966139	2025-03-12	2025-03-12	2025-06-10	digitalocean.com	*.digitalocean.com digitalocean.com *.sonar.digitalocean.com *.sonarsunset.digitalocean.com	C=US, O=Let's Encrypt, CN=E6
17142522896	2025-03-12	2025-03-12	2025-06-10	digitalocean.com	*.digitalocean.com digitalocean.com *.sonar.digitalocean.com *.sonarsunset.digitalocean.com	C=US, O=Let's Encrypt, CN=R11

I discovered many subdomains related to internal environments, staging services, and region-based infrastructure — many of which were not immediately visible through traditional DNS brute-forcing.

To automate this process, I used a curl command to pull JSON data from crt.sh's API, parsed it using jq, cleaned wildcards, and saved the results into crtsh.txt.



The command returned 235 subdomains extracted from certificate logs. These were stored for merging with other sources. I verified the contents of crtsh.txt to see what kinds of subdomains were discovered. The file included highly valuable entries such as internal-api.digitalocean.com, console-nyc3.digitalocean.com, and coffee-nyc3.digitalocean.com, many of which suggest potentially sensitive or internal services.

To create a master list of discovered subdomains, I merged output from assetfinder, subfinder, sublister, and crtsh.txt using sort -u to remove duplicates:

```
┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ cat assetfinder.txt crtsh.txt subfinder.txt sublist3r.txt | sort -u > final-subs.txt

┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ wc -l final-subs.txt
335 final-subs.txt

┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ wc -l subs.txt
335 subs.txt
```

This gave me 335 unique subdomains — a well-deduplicated and consolidated list for further probing. I double-checked whether final-subs.txt and subs.txt (my previously deduped file) had any differences by merging them again into combined-subs.txt.

Finally, I merged them to check if the count would differ somehow:

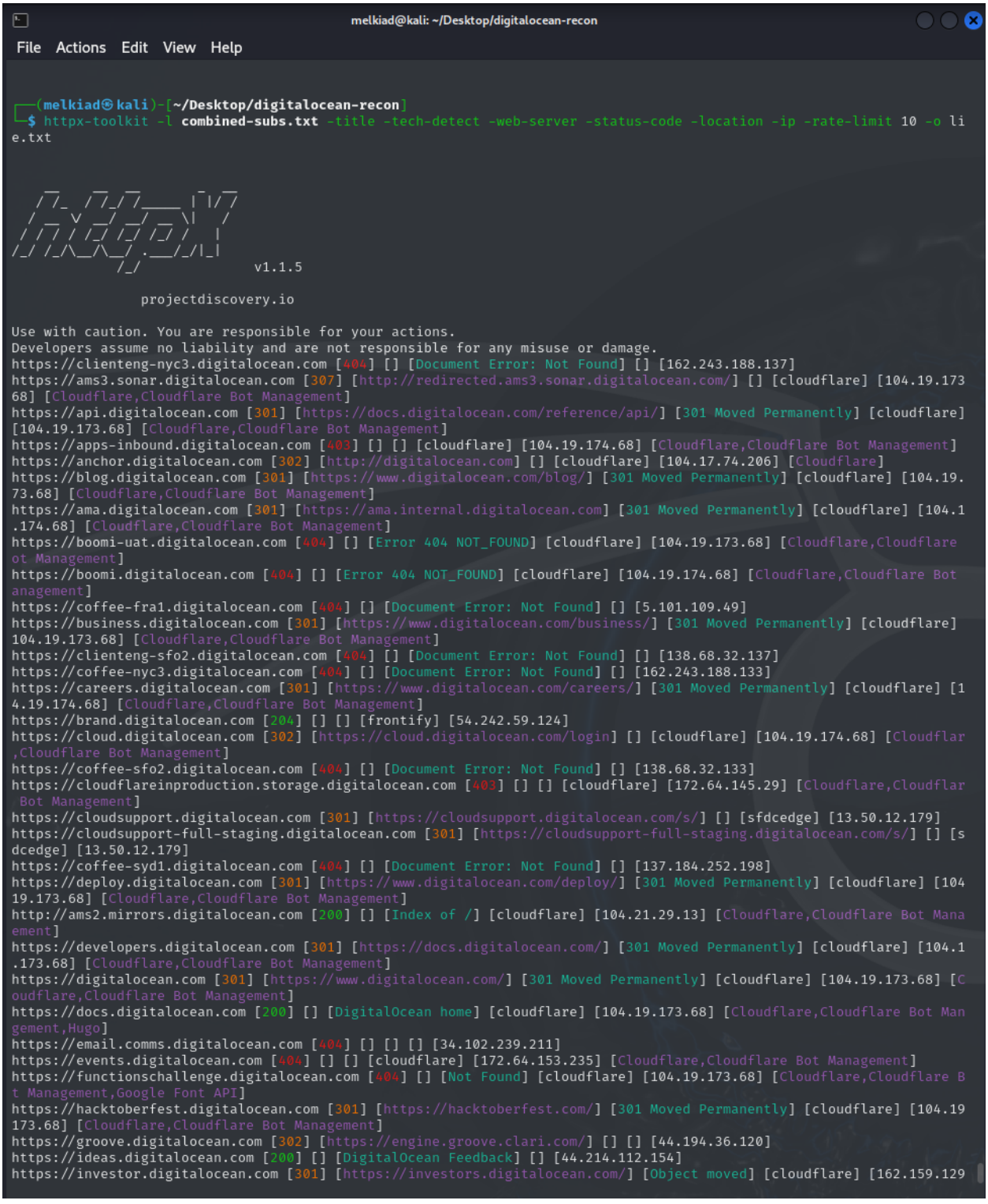
```
┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ cat final-subs.txt subs.txt | sort -u > combined-subs.txt

┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ wc -l combined-subs.txt
335 combined-subs.txt
```

The result confirmed that crtsh.txt didn't add any new unique entries — but it was still worth doing, especially for domains where CT logs might reveal more than the tools can fetch.

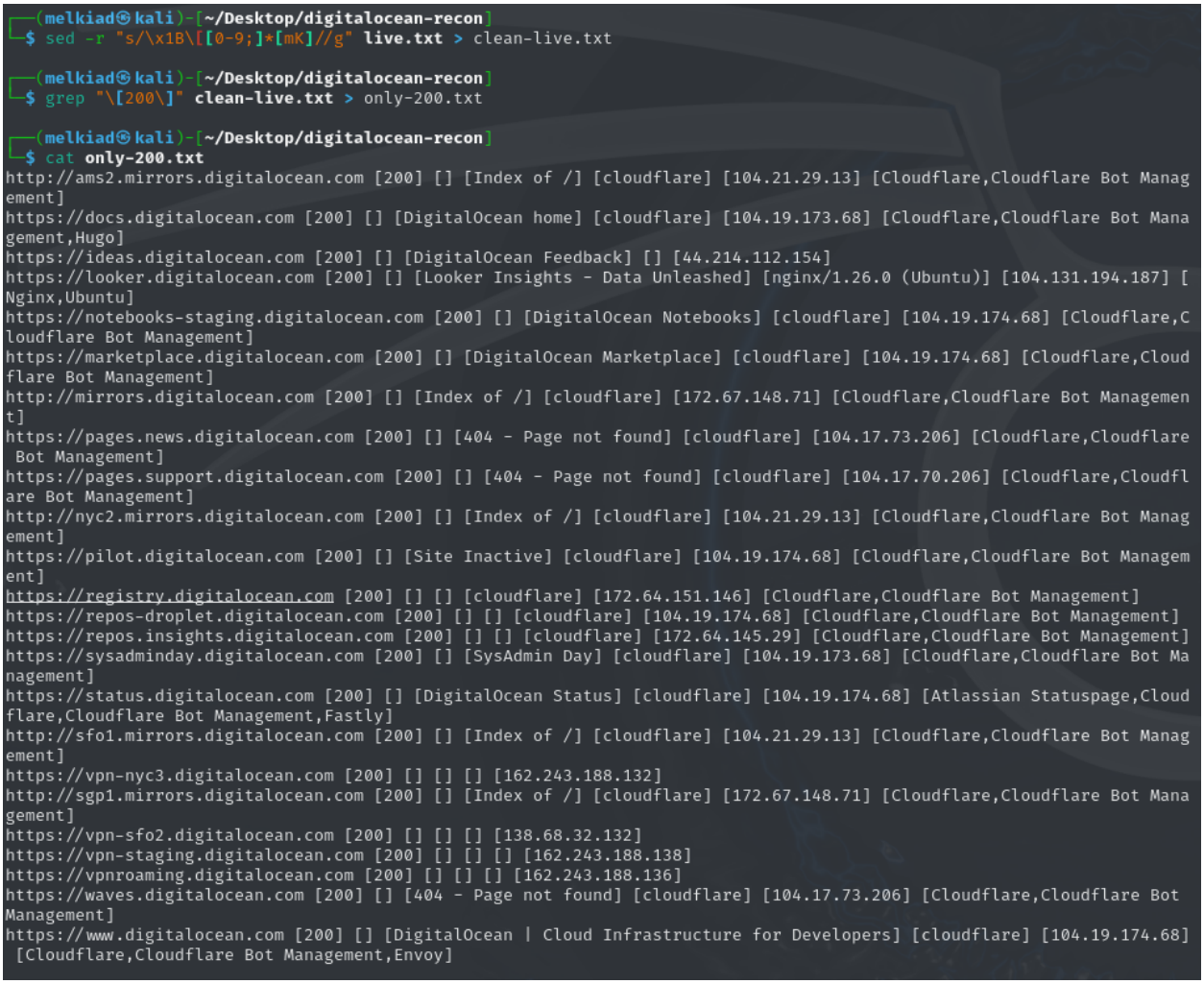
Fingerprinting

I performed banner grabbing using curl -I on http://digitalocean.com, https://digitalocean.com, and https://www.digitalocean.com. All returned 301 or 200 responses with standard headers indicating the use of Cloudflare. No backend technology or server details were exposed.

To identify which subdomains from our wordlist actually respond with live services, I used httpx-toolkit. It allowed us to enumerate HTTP responses and extract useful metadata such as status codes, IP addresses, tech stack, redirects, and titles. This respected the maximum rate of 10 requests per second as required by the bug bounty program. The output was saved in live.txt.



The raw output from httpx included terminal color codes, which are not helpful when processing data further. To clean this, I stripped ANSI escape sequences and filtered for all subdomains that responded with HTTP 200 OK. The result is a clean list of subdomains that responded successfully, including metadata such as response title, hosting provider, IP address, and detected technologies.



Next, I extracted just the URLs from this filtered list so I could easily use them with other tools for further recon:

```
┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ cut -d'[' -f1 only-200.txt | sed 's/ *$//' > only-urls.txt

┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ cat only-urls.txt
http://ams2.mirrors.digitalocean.com
https://docs.digitalocean.com
https://ideas.digitalocean.com
https://looker.digitalocean.com
https://notebooks-staging.digitalocean.com
https://marketplace.digitalocean.com
http://mirrors.digitalocean.com
https://pages.news.digitalocean.com
https://pages.support.digitalocean.com
http://nyc2.mirrors.digitalocean.com
https://pilot.digitalocean.com
https://registry.digitalocean.com
https://repos-droplet.digitalocean.com
https://repos.insights.digitalocean.com
https://sysadminday.digitalocean.com
https://status.digitalocean.com
http://sfo1.mirrors.digitalocean.com
https://vpn-nyc3.digitalocean.com
http://sgp1.mirrors.digitalocean.com
https://vpn-sfo2.digitalocean.com
https://vpn-staging.digitalocean.com
https://vpnroaming.digitalocean.com
https://waves.digitalocean.com
https://www.digitalocean.com
```

To gather banner and HTTP response headers I used this command to perform a simple HEAD request on each URL with a 10-second timeout, saving the responses to banners.txt:

```
┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ cat only-urls.txt | xargs -n1 curl -I --max-time 10 > banners.txt
```

I checked banners.txt file and the key observations were that majority return HTTP/1.1 200 OK or HTTP/2 200, indicating live and responsive servers, some return 403 Forbidden – likely due to WAF or access control, a few return 404 Not Found, suggesting unused or removed endpoints.
Speaking of cookies & session identifiers most responses set the __cf_bm cookie – evidence of Cloudflare Bot Management.
Server: cloudflare appears on nearly all domains → confirming Cloudflare CDN and WAF presence.
Most endpoints implement solid security headers X-XSS-Protection, X-Frame-Options, Content-Security-Policy, Strict-Transport-Security (that forces HTTPS for long durations), X-Content-Type-Options: nosniff. This shows well-configured security hygiene, which is expected for a major cloud provider.

I used wafw00f to detect the presence and type of Web Application Firewalls. Majority of endpoints are protected by Cloudflare WAF, some subdomains (e.g., notebooks-staging, sysadmin) are behind AWS Elastic Load Balancer, a few endpoints had no WAF detected.

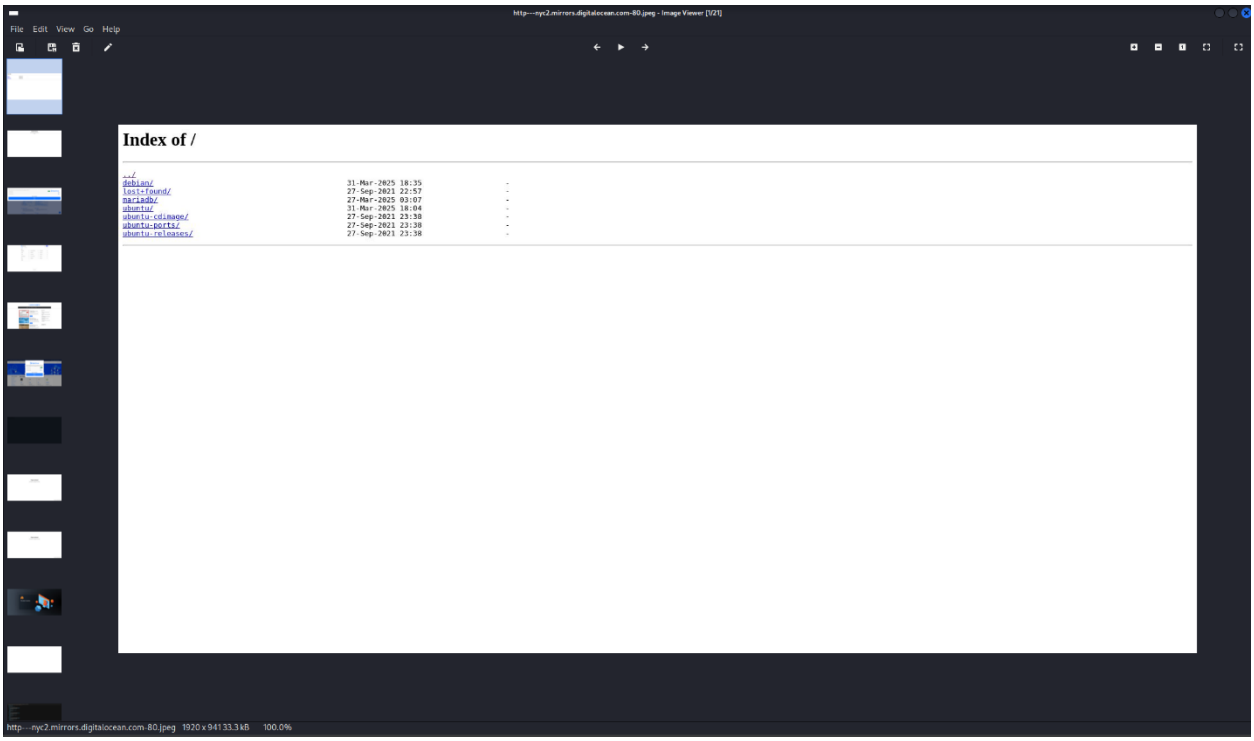

melkiad@kali: ~/Desktop/digitalocean-recon
File Actions Edit View Help
┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ wafw00f -i only-urls.txt -o waf.txt
Woof!
~ WAFW00F : v2.3.1 ~
The Web Application Firewall Fingerprinting Toolkit
[*] Checking http://ams2.mirrors.digitalocean.com
[+] The site http://ams2.mirrors.digitalocean.com is behind Cloudflare (Cloudflare Inc.) WAF.
[~] Number of requests: 2
[*] Checking https://docs.digitalocean.com
[+] The site https://docs.digitalocean.com is behind Cloudflare (Cloudflare Inc.) WAF.
[~] Number of requests: 2
[*] Checking https://ideas.digitalocean.com
[+] Generic Detection results:
[-] No WAF detected by the generic detection
[~] Number of requests: 7
[*] Checking https://looker.digitalocean.com
[+] Generic Detection results:
[-] No WAF detected by the generic detection
[~] Number of requests: 7
[*] Checking https://notebooks-staging.digitalocean.com
[+] The site https://notebooks-staging.digitalocean.com is behind AWS Elastic Load Balancer (Amazon) WAF.
[~] Number of requests: 2
[*] Checking https://marketplace.digitalocean.com
[+] The site https://marketplace.digitalocean.com is behind Cloudflare (Cloudflare Inc.) WAF.
[~] Number of requests: 2
[*] Checking http://mirrors.digitalocean.com
[+] The site http://mirrors.digitalocean.com is behind Cloudflare (Cloudflare Inc.) WAF.
[~] Number of requests: 2
[*] Checking https://pages.news.digitalocean.com
[+] The site https://pages.news.digitalocean.com is behind Cloudflare (Cloudflare Inc.) WAF.
[~] Number of requests: 2
[*] Checking https://pages.support.digitalocean.com
[+] The site https://pages.support.digitalocean.com is behind Cloudflare (Cloudflare Inc.) WAF.
[~] Number of requests: 2
[*] Checking http://nyc2.mirrors.digitalocean.com
[+] The site http://nyc2.mirrors.digitalocean.com is behind Cloudflare (Cloudflare Inc.) WAF.
[~] Number of requests: 2
[*] Checking https://pilot.digitalocean.com
[+] The site https://pilot.digitalocean.com is behind Cloudflare (Cloudflare Inc.) WAF.
[~] Number of requests: 2
[*] Checking https://registry.digitalocean.com
[+] The site https://registry.digitalocean.com is behind Cloudflare (Cloudflare Inc.) WAF.
[~] Number of requests: 2
[*] Checking https://repos-droplet.digitalocean.com
[+] The site https://repos-droplet.digitalocean.com is behind AWS Elastic Load Balancer (Amazon) WAF.
[~] Number of requests: 2
[*] Checking https://repos.insights.digitalocean.com
[+] The site https://repos.insights.digitalocean.com is behind AWS Elastic Load Balancer (Amazon) WAF.
[~] Number of requests: 2
[*] Checking https://sysadminday.digitalocean.com
[+] The site https://sysadminday.digitalocean.com is behind AWS Elastic Load Balancer (Amazon) WAF.
[~] Number of requests: 2
[*] Checking https://status.digitalocean.com

To visually inspect the website I used gowitness:

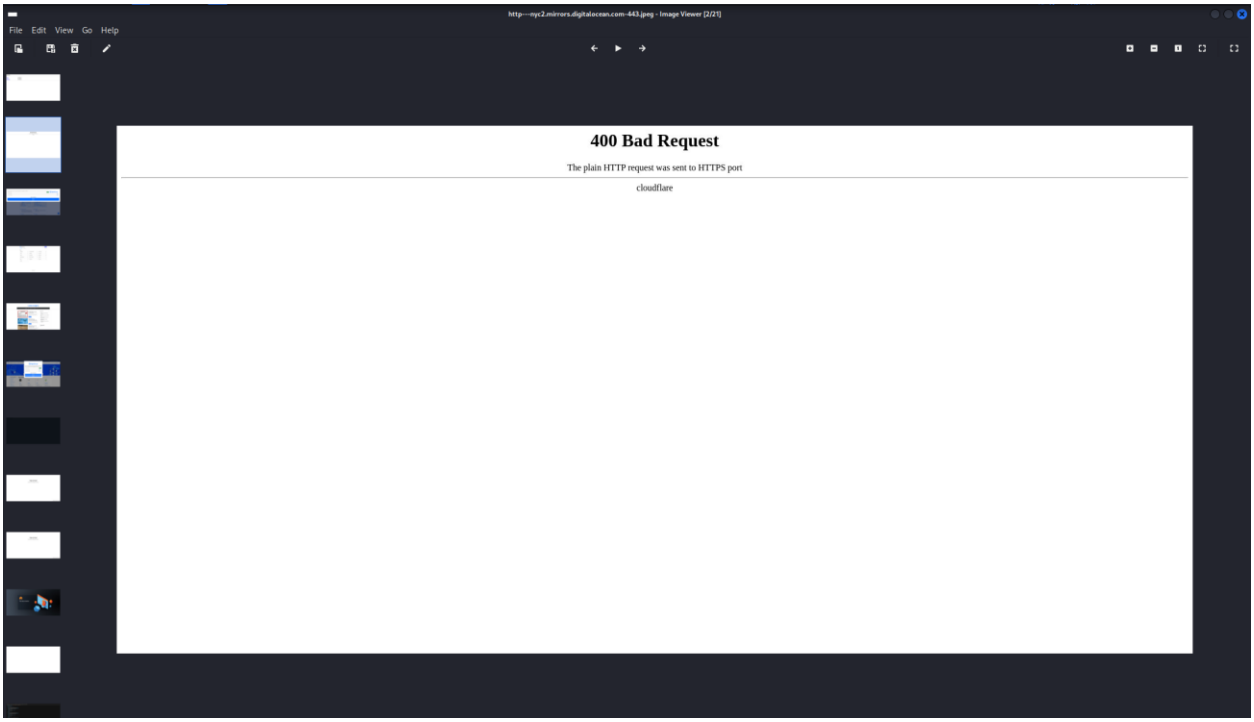
```
┌──(melkiad㉿kali)-[~/Desktop/digitalocean-recon]
└─$ gowitness scan file -f only-urls.txt --threads 3 --delay 1 --screenshot-path screenshots

2025/03/31 15:59:18 WARN no writers have been configured. to persist probe results, add writers usin
g --write-* flags
2025/03/31 15:59:21 ERROR: unhandled node event *dom.EventScrollableFlagUpdated
2025/03/31 15:59:23 ERROR: unhandled node event *dom.EventScrollableFlagUpdated
2025/03/31 15:59:24 INFO result 🤖 target=https://ideas.digitalocean.com:443 status-code=200 title="
DigitalOcean Feedback" have-screenshot=true
2025/03/31 15:59:24 INFO result 🤖 target=https://docs.digitalocean.com:443 status-code=200 title="D
ocs Home :: DigitalOcean Documentation" have-screenshot=true
2025/03/31 15:59:25 INFO result 🤖 target=https://looker.digitalocean.com:443 status-code=200 title=
"Looker Insights - Data Unleashed" have-screenshot=true
2025/03/31 15:59:26 INFO result 🤖 target=https://notebooks-staging.digitalocean.com:443 status-code
=200 title="DigitalOcean Notebooks" have-screenshot=true
2025/03/31 15:59:28 ERROR: unhandled node event *dom.EventScrollableFlagUpdated
2025/03/31 15:59:28 ERROR: unhandled page event *page.EventFrameSubtreeWillBeDetached
2025/03/31 15:59:28 ERROR: unhandled page event *page.EventFrameSubtreeWillBeDetached
2025/03/31 15:59:31 INFO result 🤖 target=https://marketplace.digitalocean.com:443 status-code=200 t
itle="DigitalOcean Marketplace" have-screenshot=true
2025/03/31 15:59:33 INFO result 🤖 target=https://pages.news.digitalocean.com:443 status-code=200 ti
tle="404 - Page not found" have-screenshot=true
2025/03/31 15:59:36 INFO result 🤖 target=https://pages.support.digitalocean.com:443 status-code=200
 title="404 - Page not found" have-screenshot=true
2025/03/31 15:59:37 INFO result 🤖 target=http://nyc2.mirrors.digitalocean.com:80 status-code=200 ti
tle="Index of /" have-screenshot=true
2025/03/31 15:59:39 INFO result 🤖 target=http://nyc2.mirrors.digitalocean.com:443 status-code=400 t
itle="400 The plain HTTP request was sent to HTTPS port" have-screenshot=true
2025/03/31 15:59:43 INFO result 🤖 target=https://pilot.digitalocean.com:443 status-code=200 title="
Site Inactive" have-screenshot=true
2025/03/31 15:59:45 INFO result 🤖 target=https://registry.digitalocean.com:443 status-code=200 titl
e="" have-screenshot=true
2025/03/31 15:59:50 INFO result 🤖 target=https://repos-droplet.digitalocean.com:443 status-code=200
 title="" have-screenshot=true
2025/03/31 15:59:59 INFO result 🤖 target=https://repos.insights.digitalocean.com:443 status-code=20
0 title="" have-screenshot=true
2025/03/31 16:00:02 INFO result 🤖 target=https://sysadminday.digitalocean.com:443 status-code=200 t
itle="SysAdmin Day" have-screenshot=true
2025/03/31 16:00:04 ERROR: unhandled node event *dom.EventScrollableFlagUpdated
2025/03/31 16:00:06 INFO result 🤖 target=https://status.digitalocean.com:443 status-code=200 title=
"DigitalOcean Status" have-screenshot=true
2025/03/31 16:00:26 INFO result 🤖 target=http://mirrors.digitalocean.com:443 status-code=522 title=
"" have-screenshot=false
2025/03/31 16:00:28 INFO result 🤖 target=https://vpn-nyc3.digitalocean.com:443 status-code=200 titl
e="" have-screenshot=true
2025/03/31 16:00:31 INFO result 🤖 target=https://vpn-sfo2.digitalocean.com:443 status-code=200 titl
e="" have-screenshot=true
2025/03/31 16:00:33 INFO result 🤖 target=https://vpn-staging.digitalocean.com:443 status-code=200 t
itle="" have-screenshot=true
2025/03/31 16:00:51 INFO result 🤖 target=https://vpnroaming.digitalocean.com:443 status-code=200 ti
tle="" have-screenshot=true
2025/03/31 16:00:52 ERROR: unhandled node event *dom.EventScrollableFlagUpdated
2025/03/31 16:00:52 INFO result 🤖 target=https://waves.digitalocean.com:443 status-code=200 title="
404 - Page not found" have-screenshot=true
2025/03/31 16:00:54 INFO result 🤖 target=https://www.digitalocean.com:443 status-code=200 title="Di
gitalOcean | Cloud Infrastructure for Developers" have-screenshot=true
```

After looking at results, it seems that 10 are live and legitimate services, about 8 are errors and misconfigurations and 4 potential risks or opportunities.

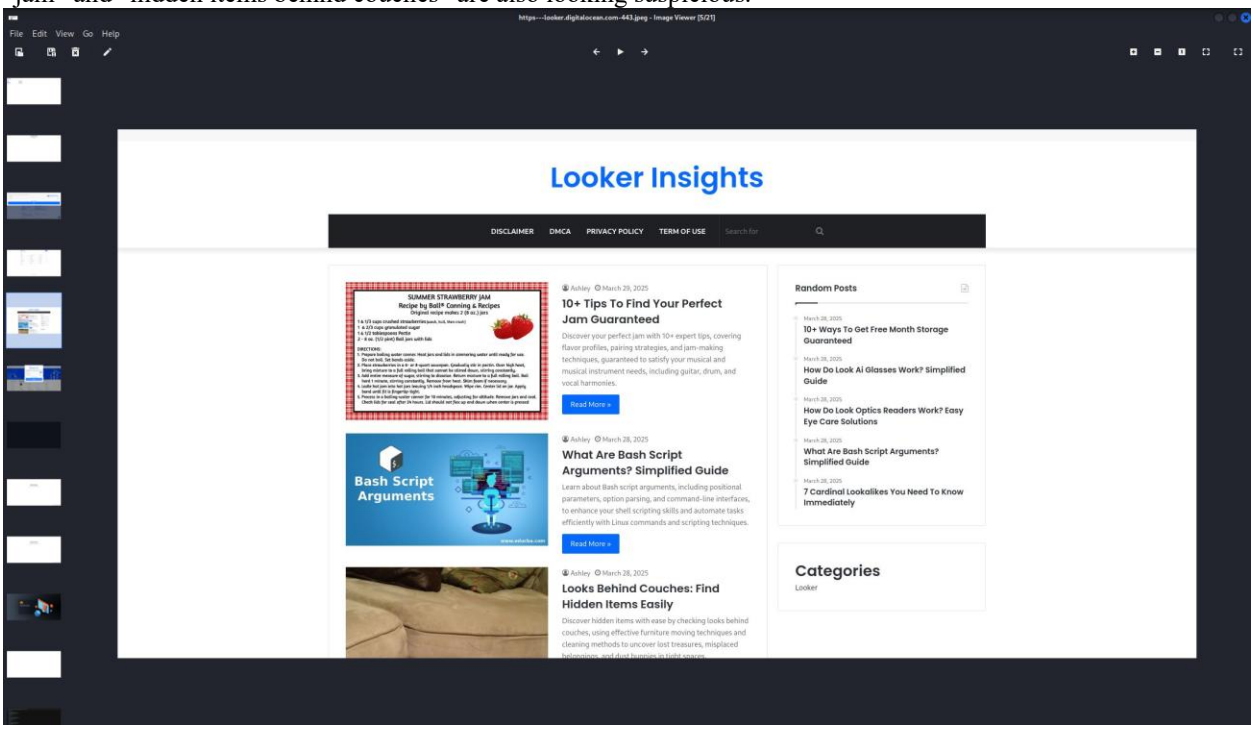
One of the most revealing findings was the presence of open directory listings. These pages serve the contents of a directory without any HTML front-end or access restrictions, commonly labeled with the title “Index of /”. This behavior often indicates a misconfigured web server where directory listing is enabled by default. While not inherently a vulnerability, it can leak internal files, scripts, or configuration data that were not intended for public access. From a security standpoint, this is worth noting as it may lead to information leakage (e.g., accessible logs, backups, scripts), unintended access to internal tools or deprecated assets, potential fingerprinting of backend technology via file types/extensions



Also, another URL returned a 400 Bad Request error. This suggests that the web server received a malformed request it couldn't process – potentially due to tool not sending expected headers (e.g., missing Host or User-Agent). This is useful in mapping the behavior of load balancers, CDNs, or security filters. While 400 errors may seem unhelpful, their consistent occurrence could signal strict input validation, WAF or reverse proxy misconfigurations or opportunities to test alternate headers or methods (e.g., GET, OPTIONS) to elicit different responses



While conducting passive reconnaissance using gowitness and curl, I discovered the subdomain looker.digitalocean.com. Upon further inspection, it turns out to host a blog-style interface. Also, the posts about “jam” and “hidden items behind couches” are also looking suspicious.



Two endpoints served raw XML formatted responses resembling Amazon S3 bucket listings. These include <ListBucketResult>, <Key>, and <LastModified> tags — consistent with AWS-style object storage. These buckets revealed directory structures and filenames without needing authentication.