Information Gathering - Web Edition:

Link to challenge: https://academy.hackthebox.com/module/144

(log in required)

Class: Tier II | Easy | Offensive

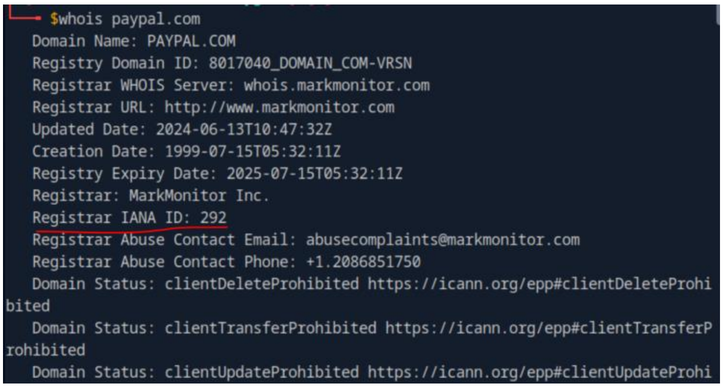
# WHOIS

**Utilising WHOIS:**

**Question:** Perform a WHOIS lookup against the paypal.com domain. What is the registrar Internet Assigned Numbers Authority (IANA) ID number?

**Answer:** 202

**Method:**



```
└──╼ $whois paypal.com
   Domain Name: PAYPAL.COM
   Registry Domain ID: 8017040_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.markmonitor.com
   Registrar URL: http://www.markmonitor.com
   Updated Date: 2024-06-13T10:47:32Z
   Creation Date: 1999-07-15T05:32:11Z
   Registry Expiry Date: 2025-07-15T05:32:11Z
   Registrar: MarkMonitor Inc.
   Registrar IANA ID: 292
   Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
   Registrar Abuse Contact Phone: +1.2086851750
   Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhi
bited
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferP
rohibited
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhi
```

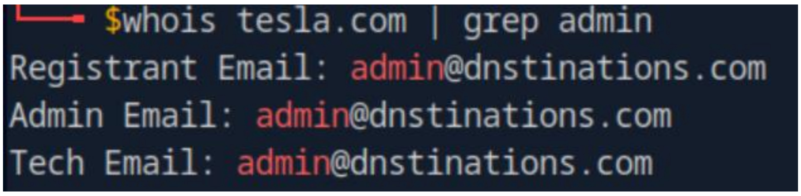
**Question:** What is the admin email contact for the tesla.com domain (also in-scope for the Tesla bug bounty program)?

**Answer:** admin@dnstinations.com

**Method:** we run the command:

```
whois tesla.com | grep admin
```

*I’ve added filtering for the keyword addmin to make the output more focusded. *:



```
└──╼ $whois tesla.com | grep admin
Registrant Email: admin@dnstinations.com
Admin Email: admin@dnstinations.com
Tech Email: admin@dnstinations.com
```

# DNS & Subdomains

**Digging DNS:**

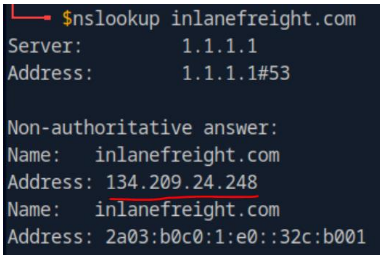
**Question:** Which IP address maps to inlanefreight.com?

**Answer:** 134.209.24.248

**Method:** we run

```
nslookup inlanefreight.com
```

and take the IPv4:



```
└──╼ $nslookup inlanefreight.com
Server:         1.1.1.1
Address:        1.1.1.1#53

Non-authoritative answer:
Name:   inlanefreight.com
Address: 134.209.24.248
Name:   inlanefreight.com
Address: 2a03:b0c0:1:e0::32c:b001
```

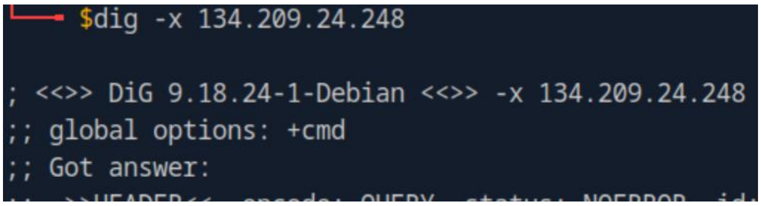
**Question:** Which domain is returned when querying the PTR record for 134.209.24.248?

**Answer:** inlanefreight.com

**Method:** we run

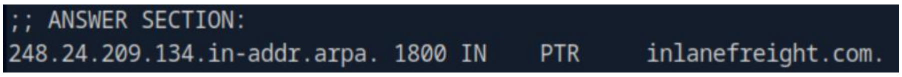
```
dig -x 134.209.24.248
```

where -x is the flag for reverse query:



```
└──╼ $dig -x 134.209.24.248

; <<>> DiG 9.18.24-1-Debian <<>> -x 134.209.24.248
;; global options: +cmd
;; Got answer:
```

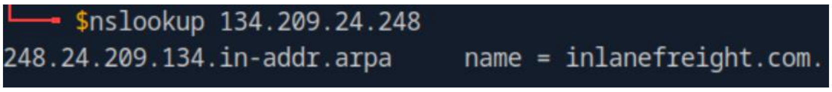
and in the answer we have this:



```
;; ANSWER SECTION:
248.24.209.134.in-addr.arpa. 1800 IN    PTR     inlanefreight.com.
```

Another option is:

```
nslookup 134.209.24.248
```



```
└──╼ $nslookup 134.209.24.248
248.24.209.134.in-addr.arpa     name = inlanefreight.com.
```

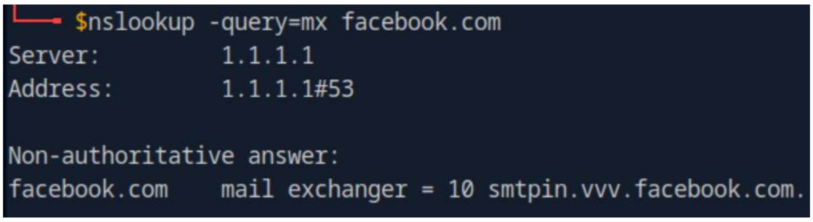
**Question:** What is the full domain returned when you query the mail records for facebook.com?

**Answer:** smtpin.vvv.facebook.com.

**Method:** we run the command

```
nslookup -query=mx facebook.com
```

where ‘mx’ is mail exchange:



```
└──╼ $nslookup -query=mx facebook.com
Server:         1.1.1.1
Address:        1.1.1.1#53

Non-authoritative answer:
facebook.com    mail exchanger = 10 smtpin.vvv.facebook.com.
```

**Subdomain Bruteforcing:**

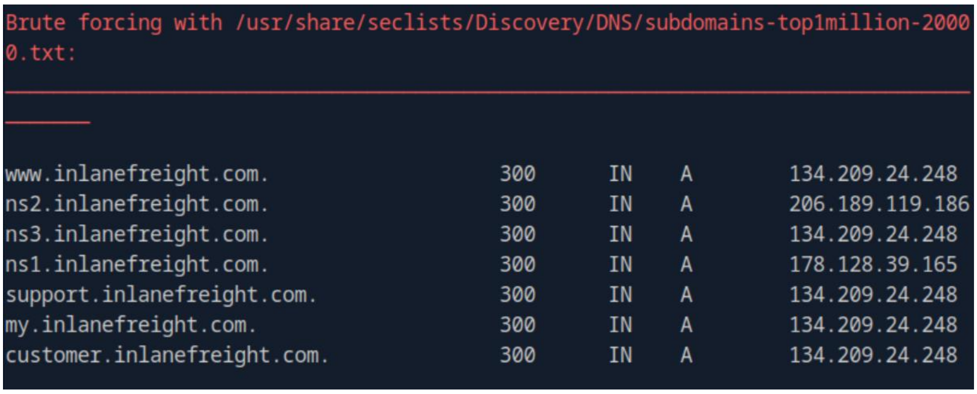
**Question:** Using the known subdomains for inlanefreight.com (www, ns1, ns2, ns3, blog, support, customer), find any missing subdomains by brute-forcing possible domain names. Provide your answer with the complete subdomain, e.g., www.inlanefreight.com.

**Answer:** my.inlanefreight.com

**Method:** we run the command:

```
dnsenum --enum inlanefreight.com -f
/usr/share/seclists/Discovery/DNS/subdomains-top1million-
20000.txt
```

in the command we enumerate the target ‘inlanefreight.com’, using the provided wordlist fit for subdomain enumeration:



```
Brute forcing with /usr/share/seclists/Discovery/DNS/subdomains-top1million-2000
0.txt:


www.inlanefreight.com.                  300      IN     A        134.209.24.248
ns2.inlanefreight.com.                  300      IN     A        206.189.119.186
ns3.inlanefreight.com.                  300      IN     A        134.209.24.248
ns1.inlanefreight.com.                  300      IN     A        178.128.39.165
support.inlanefreight.com.              300      IN     A        134.209.24.248
my.inlanefreight.com.                   300      IN     A        134.209.24.248
customer.inlanefreight.com.             300      IN     A        134.209.24.248
```

At the end we get to this list of subdomain, and we take the option (‘my’) that was not listed within the question.

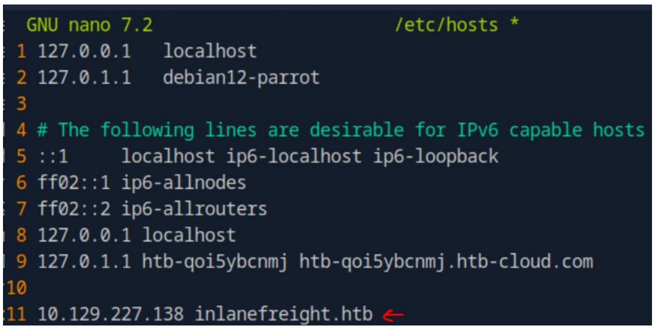
**DNS Zone Transfers:**

**Question:** After performing a zone transfer for the domain inlanefreight.htb on the target system, how many DNS records are retrieved from the target system's name server? Provide your answer as an integer, e.g, 123.

**Answer:** 22

**Method:** First, lets link our domain ‘inlanefreight.htb’ to the target machine’s IP On the pwnbox ‘etc/hosts’ – we will use the command:

```
sudo nano /etc/hosts
```

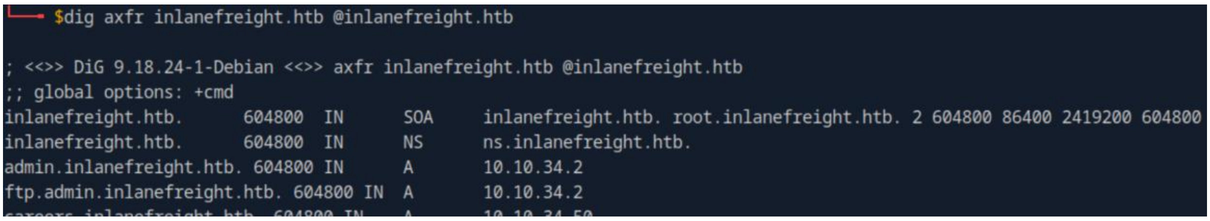


```
 GNU nano 7.2                       /etc/hosts *
127.0.0.1    localhost
127.0.1.1    debian12-parrot

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1 localhost
127.0.1.1 htb-qoi5ybcnmj htb-qoi5ybcnmj.htb-cloud.com

10.129.227.138 inlanefreight.htb
```

When linked we will run the command:

```
dig axfr inlanefreight.htb @inlanefreight.htb
```

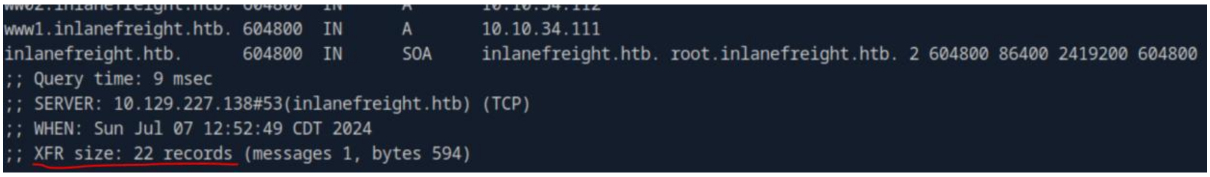
the command will attempt to perform a zone transfer from the name server ‘inlanefreight.htb’ for the entire domain of ‘inlanefreight.htb’:



```
└──╼ $dig axfr inlanefreight.htb @inlanefreight.htb

; <<>> DiG 9.18.24-1-Debian <<>> axfr inlanefreight.htb @inlanefreight.htb
;; global options: +cmd
inlanefreight.htb.      604800  IN      SOA     inlanefreight.htb. root.inlanefreight.htb. 2 604800 86400 2419200 604800
inlanefreight.htb.      604800  IN      NS      ns.inlanefreight.htb.
admin.inlanefreight.htb. 604800 IN      A       10.10.34.2
ftp.admin.inlanefreight.htb. 604800 IN  A       10.10.34.2
careers.inlanefreight.htb. 604800 IN    A       10.10.34.50
```

*

*



```
www2.inlanefreight.htb. 604800  IN      A       10.10.34.112
www1.inlanefreight.htb. 604800  IN      A       10.10.34.111
inlanefreight.htb.      604800  IN      SOA     inlanefreight.htb. root.inlanefreight.htb. 2 604800 86400 2419200 604800
;; Query time: 9 msec
;; SERVER: 10.129.227.138#53(inlanefreight.htb) (TCP)
;; WHEN: Sun Jul 07 12:52:49 CDT 2024
;; XFR size: 22 records (messages 1, bytes 594)
```

*The list of domains displayed within this output screenshots is partial. *

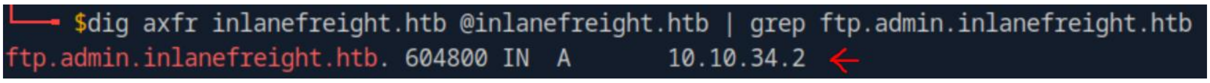
At the end of the output we see the amount of DNS records retrieved from the target name’s server.

**Question:** Within the zone record transferred above, find the ip address for ftp.admin.inlanefreight.htb. Respond only with the IP address, eg 127.0.0.1

**Answer:** 10.10.34.2

**Method:** lets add to the command in the question above a ‘grep’ filter for more targeted search:

```
dig axfr inlanefreight.htb @inlanefreight.htb | grep
ftp.admin.inlanefreight.htb
```



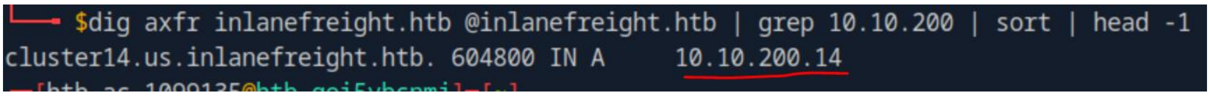
```
└──╼ $dig axfr inlanefreight.htb @inlanefreight.htb | grep ftp.admin.inlanefreight.htb
ftp.admin.inlanefreight.htb. 604800 IN  A       10.10.34.2
```

**Question:** Within the same zone record, identify the largest IP address allocated within the 10.10.200 IP range. Respond with the full IP address, eg 10.10.200.1

**Answer:** 10.10.200.14

**Method:** here we will add another ‘grep’ filter, sorting the list from highest to lowest with the command, and finishing it up with ‘head -1’ to display only the first line, which will include the highest IP within the specified range:

```
dig axfr inlanefreight.htb @inlanefreight.htb | grep
10.10.200 | sort | head -1
```



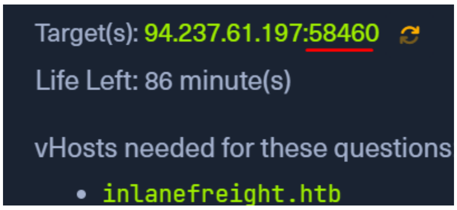
```
└──╼ $dig axfr inlanefreight.htb @inlanefreight.htb | grep 10.10.200 | sort | head -1
cluster14.us.inlanefreight.htb. 604800 IN A     10.10.200.14
┌─[htb-ac-1099135@htb-goi5vbcnmi]─[~]
```

**Virtual Hosts:**

**Question:** Brute-force vhosts on the target system. What is the full subdomain that is prefixed with "web"? Answer using the full domain, e.g. "x.inlanefreight.htb"

**Answer:** web17611.inlanefreight.htb

**Method:** First, when we initiate a target machine, we notice that the target machine also comes with a port:



Target(s): 94.237.61.197:58460

Life Left: 86 minute(s)

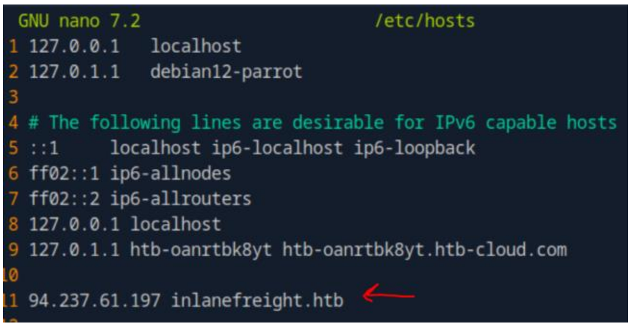
vHosts needed for these questions

- inlanefreight.htb

It will come in handy later on.

The target machine IP provided - we will link up with the vHost domain in 'etc/hosts':

```
sudo nano /etc/hosts
```



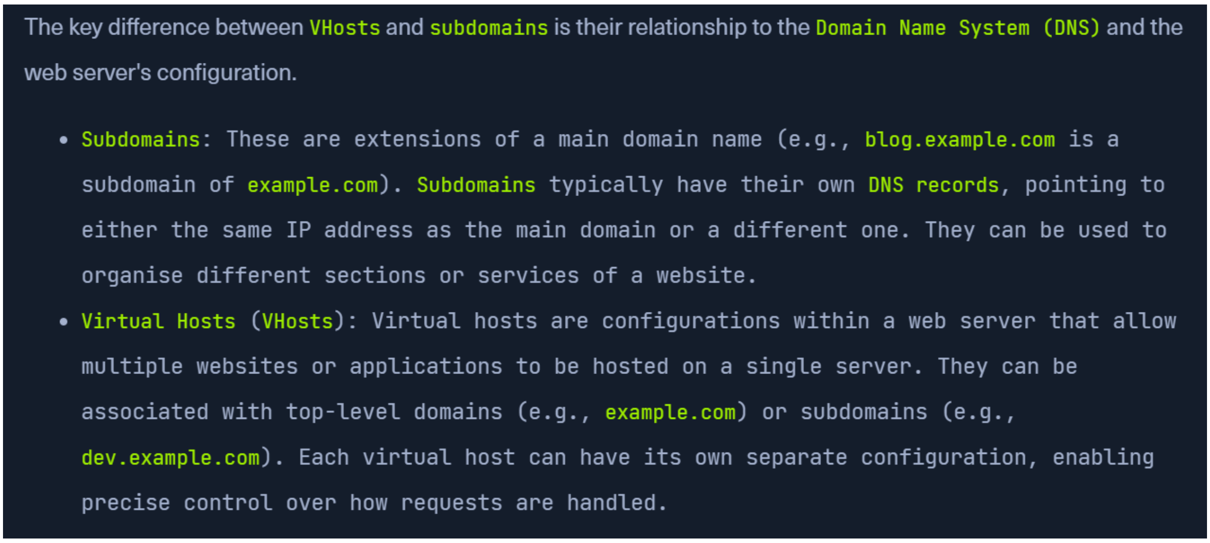
```
GNU nano 7.2                    /etc/hosts
127.0.0.1    localhost
127.0.1.1    debian12-parrot

# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1 localhost
127.0.1.1 htb-oanrtbk8yt htb-oanrtbk8yt.htb-cloud.com

94.237.61.197 inlanefreight.htb
```

When the configuration is complete – lets start with the virtual hosts bruteforce with the command:

```
gobuster vhost -u http://inlanefreight.htb:58460 -w
/usr/share/seclists/Discovery/DNS/subdomains-top1million-
110000.txt --append-domain
```

the command will brute force for Vhosts within the target of 'inlanefreight.htb:58460', we can notice it is the port provided for us on the target machine.

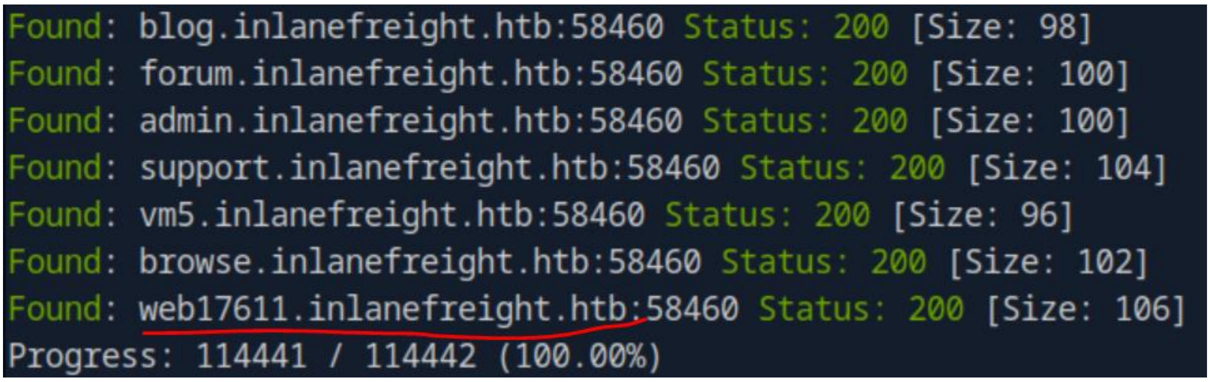
Before I will present the output results – its important to differentiate between virtual host and subdomain:



The key difference between VHosts and subdomains is their relationship to the Domain Name System (DNS) and the web server's configuration.

- Subdomains: These are extensions of a main domain name (e.g., blog.example.com is a subdomain of example.com). Subdomains typically have their own DNS records, pointing to either the same IP address as the main domain or a different one. They can be used to organise different sections or services of a website.
- Virtual Hosts (VHosts): Virtual hosts are configurations within a web server that allow multiple websites or applications to be hosted on a single server. They can be associated with top-level domains (e.g., example.com) or subdomains (e.g., dev.example.com). Each virtual host can have its own separate configuration, enabling precise control over how requests are handled.

and here is the output:



```
Found: blog.inlanefreight.htb:58460 Status: 200 [Size: 98]
Found: forum.inlanefreight.htb:58460 Status: 200 [Size: 100]
Found: admin.inlanefreight.htb:58460 Status: 200 [Size: 100]
Found: support.inlanefreight.htb:58460 Status: 200 [Size: 104]
Found: vm5.inlanefreight.htb:58460 Status: 200 [Size: 96]
Found: browse.inlanefreight.htb:58460 Status: 200 [Size: 102]
Found: web17611.inlanefreight.htb:58460 Status: 200 [Size: 106]
Progress: 114441 / 114442 (100.00%)
```

We can notice the full subdomain that is prefixed with ‘web’ is ‘web17611.inlanefreight.htb’.

*Note - in this result the vhosts are subdomains, but they don’t have to be generally. *

**Question:** Brute-force vhosts on the target system. What is the full subdomain that is prefixed with "vm"? Answer using the full domain, e.g. "x.inlanefreight.htb"

**Answer:** vm5.inlanefreight.htb

**Method:** from the same result above, we take the ‘vm5’ option which is the only answer whose ‘vm’ is prefix of.

**Question:** Brute-force vhosts on the target system. What is the full subdomain that is prefixed with "br"? Answer using the full domain, e.g. "x.inlanefreight.htb"

**Answer:** browse.inlanefreight.htb

**Method:** from the same result above, we take the ‘browse’ option which is the only answer whose ‘br’ is prefix of.

**Question:** Brute-force vhosts on the target system. What is the full subdomain that is prefixed with "a"? Answer using the full domain, e.g. "x.inlanefreight.htb"

**Answer:** admin.inlanefreight.htb

**Method:** from the same result above, we take the ‘admin’ option which is the only answer whose ‘a’ is prefix of.

**Question:** Brute-force vhosts on the target system. What is the full subdomain that is prefixed with "su"? Answer using the full domain, e.g. "x.inlanefreight.htb"

**Answer:** support.inlanefreight.htb

**Method:** from the same result above, we take the ‘support’ option which is the only answer whose ‘su’ is prefix of.

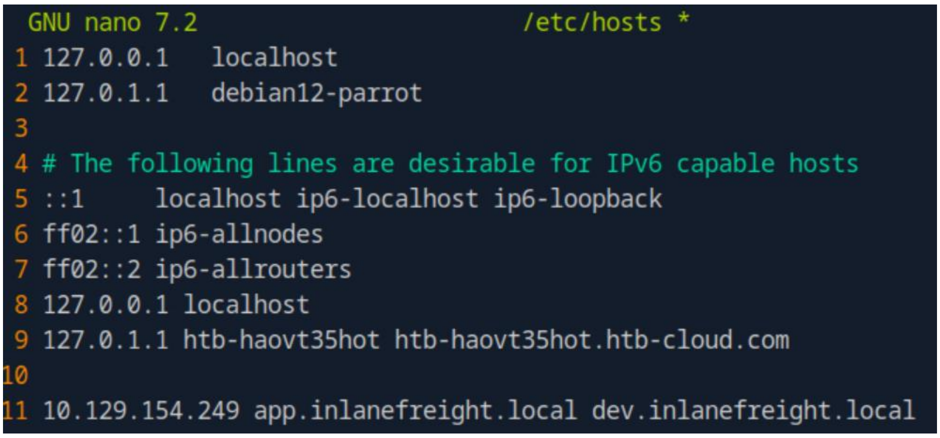
# Fingerprinting

**Fingerprinting:**

**Question:** Determine the Apache version running on app.inlanefreight.local on the target system. (Format: 0.0.0)

**Answer:** 2.4.41

**Method:** First, lets link the target IP with the requested hosts in ‘etc/hosts’ as in previous times (line 11 in the screenshot below):

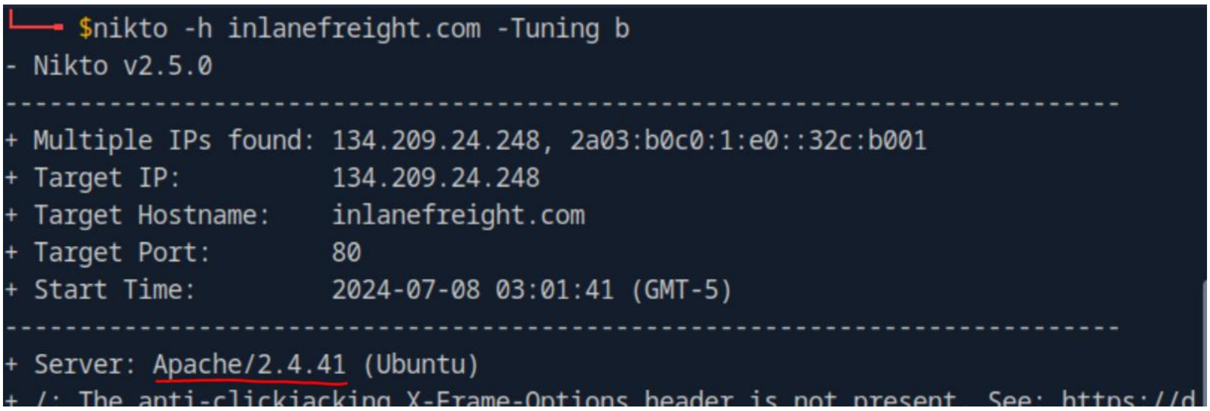


```
 GNU nano 7.2                         /etc/hosts *
1 127.0.0.1   localhost
2 127.0.1.1   debian12-parrot
3
4 # The following lines are desirable for IPv6 capable hosts
5 ::1     localhost ip6-localhost ip6-loopback
6 ff02::1 ip6-allnodes
7 ff02::2 ip6-allrouters
8 127.0.0.1 localhost
9 127.0.1.1 htb-haovt35hot htb-haovt35hot.htb-cloud.com
10
11 10.129.154.249 app.inlanefreight.local dev.inlanefreight.local
```

When done, we will run the command:

```
nikto -h inlanefreight.com -Tuning b
```

using the tool ‘nikto’ open source scanner, where The -h flag specifies the target host. The -Tuning b flag tells Nikto to only run the Software Identification modules:



```
└──╼ $nikto -h inlanefreight.com -Tuning b
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Multiple IPs found: 134.209.24.248, 2a03:b0c0:1:e0::32c:b001
+ Target IP:          134.209.24.248
+ Target Hostname:    inlanefreight.com
+ Target Port:        80
+ Start Time:         2024-07-08 03:01:41 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.41 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://d
```

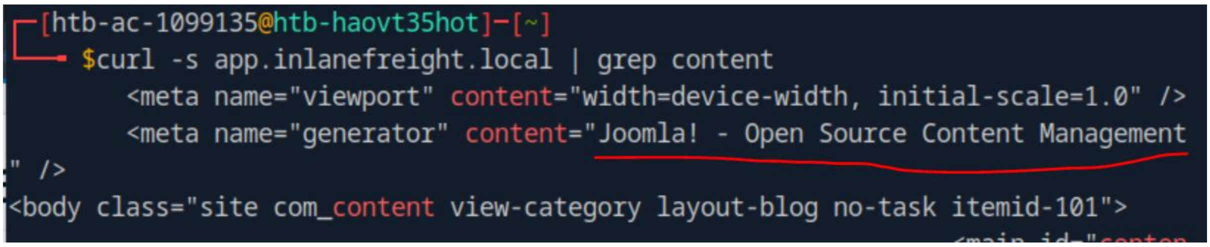
**Question:** Which CMS is used on app.inlanefreight.local on the target system? Respond with the name only, e.g., WordPress.

**Answer:** Joomla

**Method:** we run ‘curl’ on the command, filtering for ‘content:

```
curl -s app.inlanefreight.local | grep content
```

and within the deader we will find ‘Joomla’ as the open-source content management:



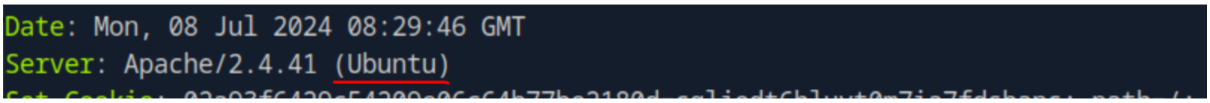
```
┌─[htb-ac-1099135@htb-haovt35hot]─[~]
└──╼ $curl -s app.inlanefreight.local | grep content
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <meta name="generator" content="Joomla! - Open Source Content Management
" />
<body class="site com_content view-category layout-blog no-task itemid-101">
```

**Question:** On which operating system is the dev.inlanefreight.local webserver running in the target system? Respond with the name only, e.g., Debian.

**Answer:** Ubuntu

**Method:** lets run curl -I on the target:

```
curl -I dev.inlanefreight.local
```



```
Date: Mon, 08 Jul 2024 08:29:46 GMT
Server: Apache/2.4.41 (Ubuntu)
```

Also,the ‘nikto’ command from the first question also gives the OS (you can refer to the ‘nikto’ results screenshot).

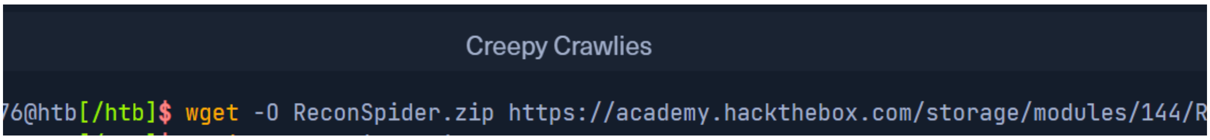
# Crawling

**Creepy Crawlies:**

**Question:** After spidering inlanefreight.com, identify the location where future reports will be stored. Respond with the full domain, e.g., files.inlanefreight.com.

**Answer:** inlanefreight-comp133.s3.amazonaws.htb

**Method:** Important note – the script downloaded within the section turtorial from



Creepy Crawlies

```
76@htb[/htb]$ wget -O ReconSpider.zip https://academy.hackthebox.com/storage/modules/144/R
```

*from module tutorial* is malfunctioning.

A working (similar) script can be downloaded from here.

Install scrapy module

```
sudo apt install python3-scrapy
```

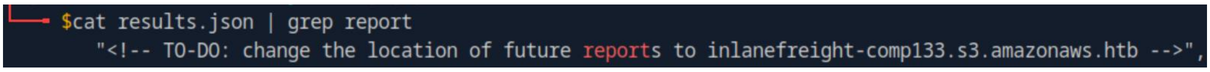
and run the script:

```
python3 ReconSpider.py http://inlanefreight.com
```

after a minute or 2 the script will finish running and we will have the output file 'results.json' – open it with grep filter for 'report'

```
cat results.json | grep report
```

and we can observe the result:



```
└── $cat results.json | grep report
        "<!-- TO-DO: change the location of future reports to inlanefreight-comp133.s3.amazonaws.htb -->",
```

# Web Archives

**Web Archives:**

**Question:** How many Pen Testing Labs did HackTheBox have on the 8th August 2018? Answer with an integer, eg 1234.

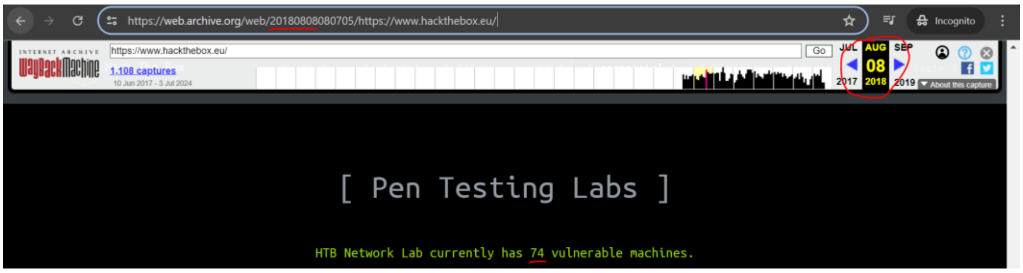
**Answer:** 74

**Method:** back in 8th August 2018 the hackthebox domain was 'eu'.

Lets go to https://web.archive.org/web/20180808080705/https://www.hackthebox.eu/

(pay attention that the url contains the string '20180808' which means 2018.08.08 in format YYYY-MM-DD), and search for the word 'labs':

Somewhere in the page we will encounter this:



https://web.archive.org/web/20180808080705/https://www.hackthebox.eu/

https://www.hackthebox.eu/

[ Pen Testing Labs ]

HTB Network Lab currently has 74 vulnerable machines.

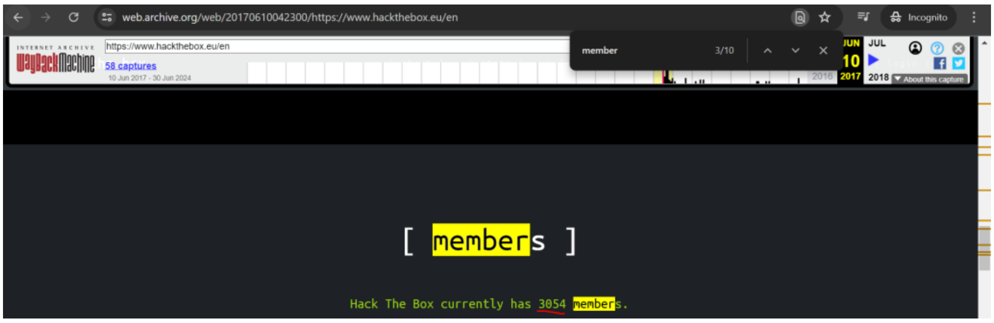
**Question:** How many members did HackTheBox have on the 10th June 2017? Answer with an integer, eg 1234.

**Answer:** 3054

**Method:** lets set the date to 10th June 2017:

https://web.archive.org/web/20170610042300/https://www.hackthebox.eu/en

looking for the keyword 'member' – we will eventually encounter this:



web.archive.org/web/20170610042300/https://www.hackthebox.eu/en

https://www.hackthebox.eu/en

member 3/10

[ members ]

Hack The Box currently has 3054 members.

**Question:** Going back to March 2002, what website did the facebook.com domain redirect too? Answer with the full domain, eg http://www.facebook.com/

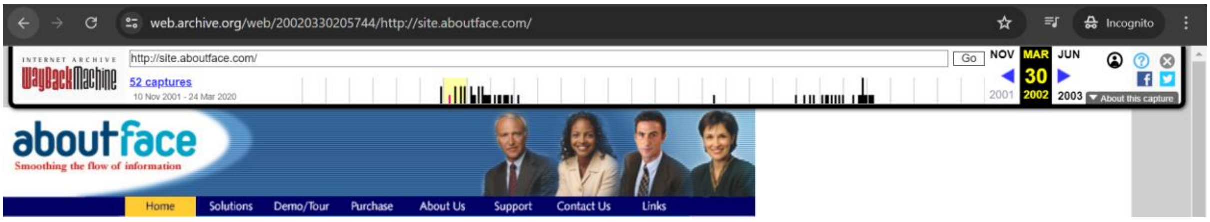
**Answer:** http://site.aboutface.com/

**Method:** entering in the URL the address:

https://web.archive.org/web/20020330205744/http://www.facebook.com

(remember the '200203' dating format in the url)

will direct us to:

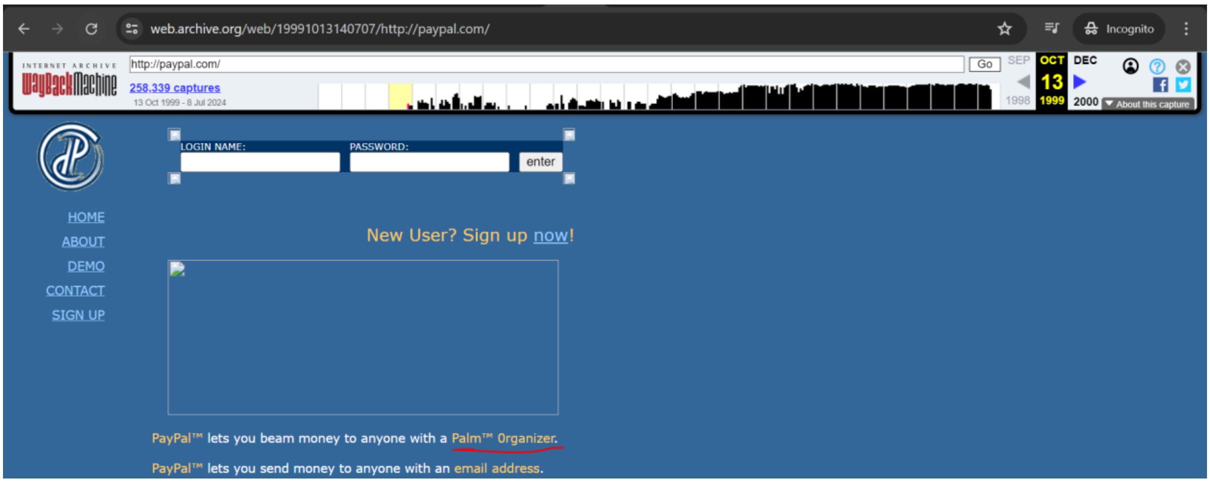


**Question:** According to the paypal.com website in October 1999, what could you use to "beam money to anyone"? Answer with the product name, eg My Device, remove the ™ from your answer.

**Answer:** Palm Organizer

**Method:** lets go to:

https://web.archive.org/web/19991013140707/http://paypal.com/



New User? Sign up now!

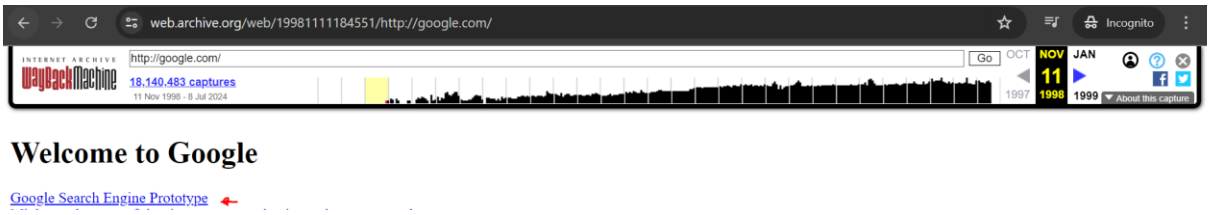
PayPal™ lets you beam money to anyone with a Palm™ Organizer.

PayPal™ lets you send money to anyone with an email address.

**Question:** Going back to November 1998 on google.com, what address hosted the non-alpha "Google Search Engine Prototype" of Google? Answer with the full address, eg http://google.com

**Answer:** http://google.stanford.edu/

**Method:** go to https://web.archive.org/web/19981111184551/http://google.com/

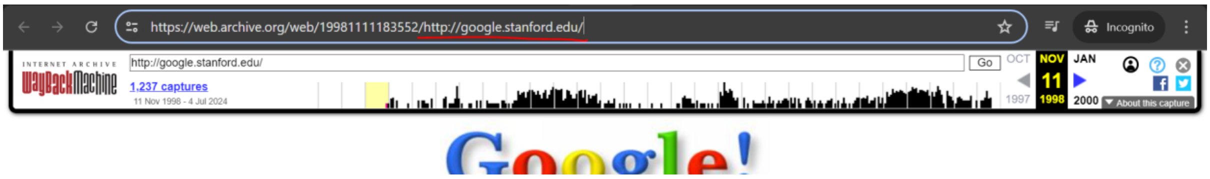


Welcome to Google

Google Search Engine Prototype

Go to the marked link:

And we have the answer here:

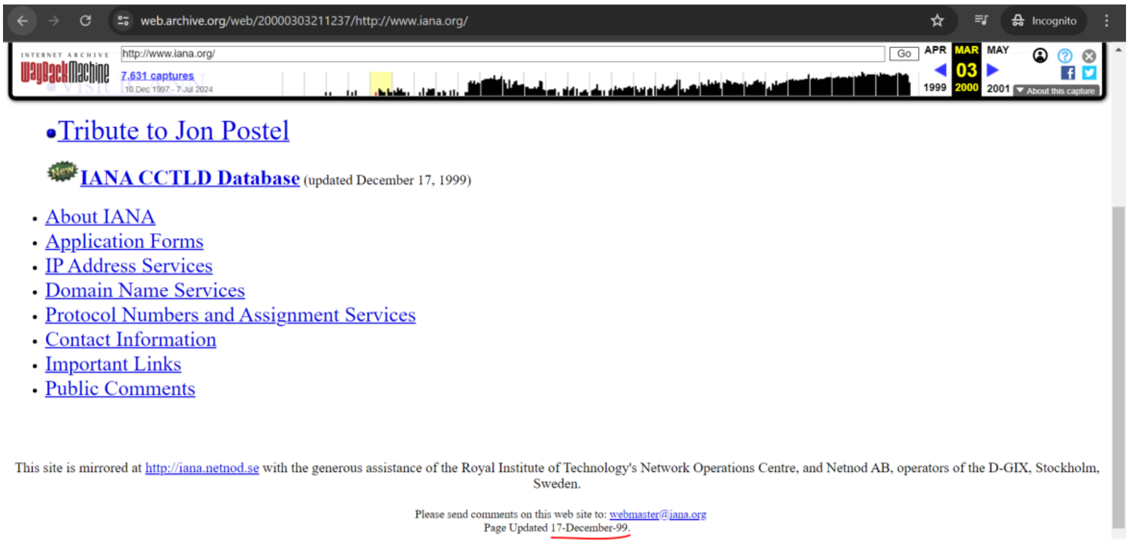


**Question:** Going back to March 2000 on www.iana.org, when exacty was the site last updated? Answer with the date in the footer, eg 11-March-99

**Answer:** 17-December-99

**Method:**

https://web.archive.org/web/20000303211237/http://www.iana.org/



- Tribute to Jon Postel

IANA CCTLD Database (updated December 17, 1999)

- About IANA
- Application Forms
- IP Address Services
- Domain Name Services
- Protocol Numbers and Assignment Services
- Contact Information
- Important Links
- Public Comments

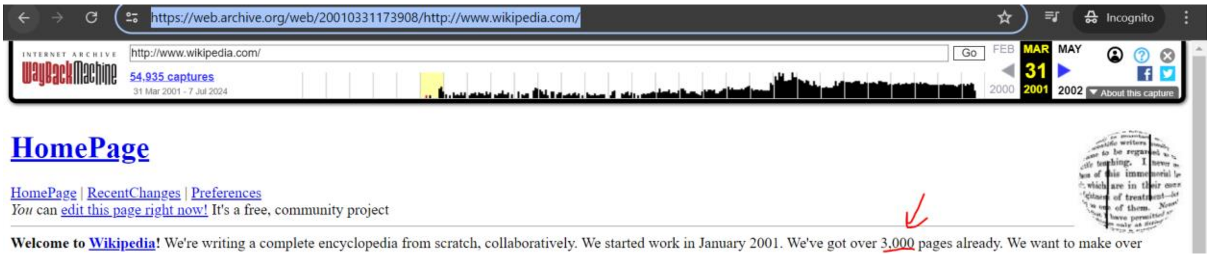
This site is mirrored at http://iana.netnod.se with the generous assistance of the Royal Institute of Technology's Network Operations Centre, and Netnod AB, operators of the D-GIX, Stockholm, Sweden.

Please send comments on this web site to: webmaster@iana.org
Page Updated 17-December-99.

**Question:** According to the wikipedia.com snapshot taken in March 2001, how many pages did they have over? Answer with the number they state without any commas, eg 2000 not 2,000

**Answer:** 3000

**Method:**
https://web.archive.org/web/20010331173908/http://www.wikipedia.com/



HomePage

HomePage | RecentChanges | Preferences
*You* can edit this page right now! It's a free, community project

**Welcome to Wikipedia!** We're writing a complete encyclopedia from scratch, collaboratively. We started work in January 2001. We've got over 3,000 pages already. We want to make over

# Skills Assessment

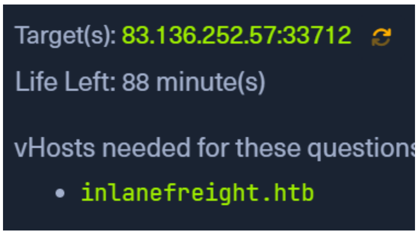
**Web Recon - Skills Assessment:**

*note – as the completion of the section took several session, the target IP and port will occasionally change. *

**Question:** What is the IANA ID of the registrar of the inlanefreight.com domain?

**Answer:** 468

**Method:** in this skill assessment section our target machine also comes with a porta as well as IP (just like 'Virtual Hosts' section):



Target(s): 83.136.252.57:33712

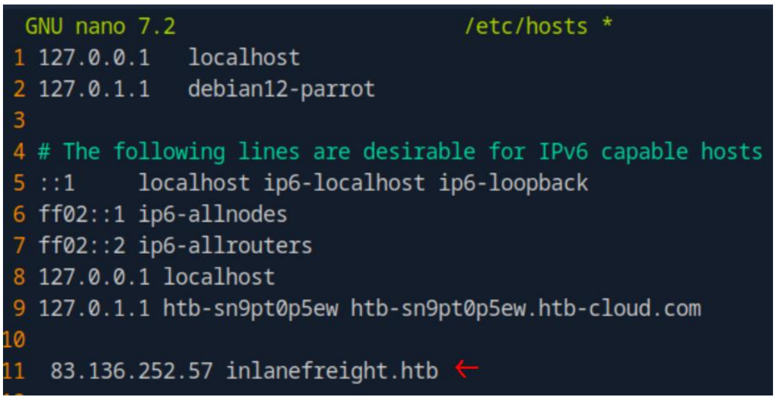
Life Left: 88 minute(s)

vHosts needed for these questions

- inlanefreight.htb

We will link in 'etc/hosts' the IP to the vHost:

```
sudo nano /etc/hosts
```

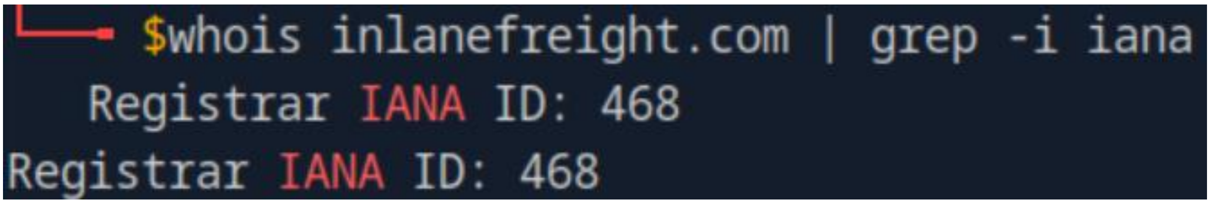


```
 GNU nano 7.2                         /etc/hosts *
127.0.0.1    localhost
127.0.1.1    debian12-parrot

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1 localhost
127.0.1.1 htb-sn9pt0p5ew htb-sn9pt0p5ew.htb-cloud.com

 83.136.252.57 inlanefreight.htb ←
```

Now when the vHost and IP are linked, lets run:

```
whois inlanefreight.com | grep -i iana
```

in similar way to what we did in 'Utilising WHOIS' section:



```
└──╼ $whois inlanefreight.com | grep -i iana
   Registrar IANA ID: 468
Registrar IANA ID: 468
```

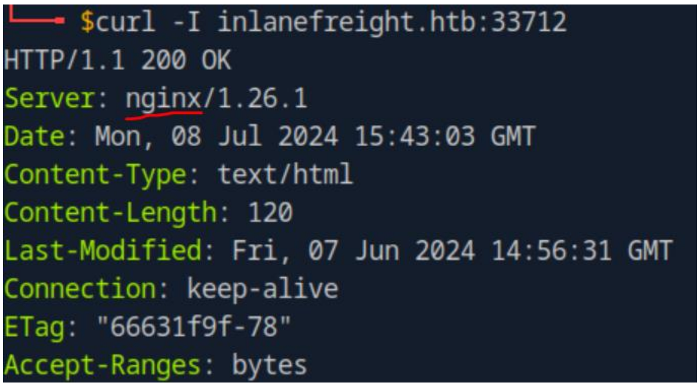
**Question:** What http server software is powering the inlanefreight.htb site on the target system? Respond with the name of the software, not the version, e.g., Apache.

**Answer:** nginx

**Method:** using the target port provided, we run:

```
curl -I inlanefreight.htb:33712
```

for the page metadata (-I flag is for ‘Show document info only’):



```
└──╼ $curl -I inlanefreight.htb:33712
HTTP/1.1 200 OK
Server: nginx/1.26.1
Date: Mon, 08 Jul 2024 15:43:03 GMT
Content-Type: text/html
Content-Length: 120
Last-Modified: Fri, 07 Jun 2024 14:56:31 GMT
Connection: keep-alive
ETag: "66631f9f-78"
Accept-Ranges: bytes
```

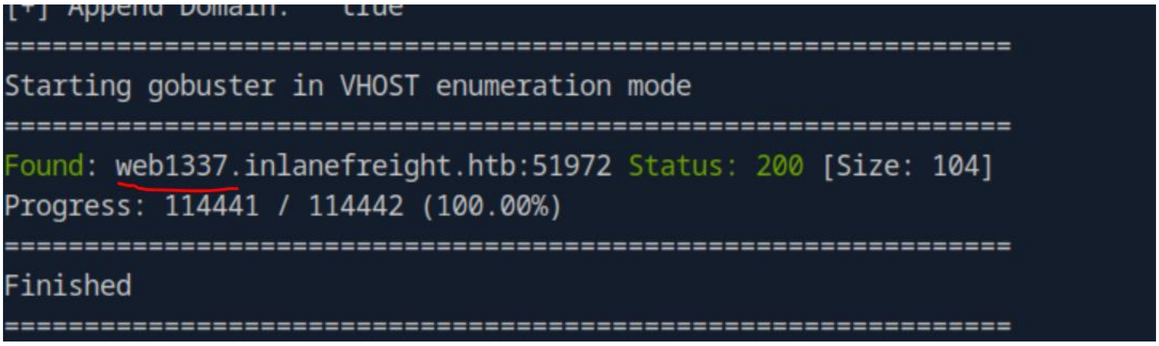
**Question:** What is the API key in the hidden admin directory that you have discovered on the target system?

**Answer:** e963d863ee0e82ba7080fbf558ca0d3f

**Method:** we will start by looking for subdomains, we will use the command:

```
gobuster vhost -u http://inlanefreight.htb:<target-port> -w
/usr/share/seclists/Discovery/DNS/subdomains-top1million-
110000.txt --append-domain
```

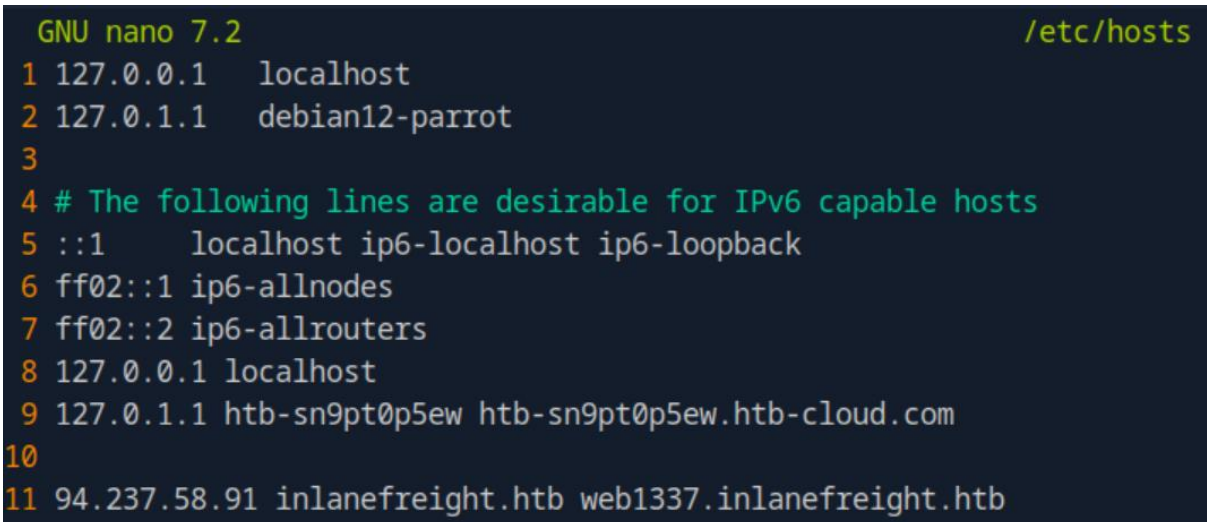
to bruteforce for subdomains within ‘inlanefreight.htb’ using provided wordlist (‘top1million’):



```
[+] Append Domain:    true
===============================================================
Starting gobuster in VHOST enumeration mode
===============================================================
Found: web1337.inlanefreight.htb:51972 Status: 200 [Size: 104]
Progress: 114441 / 114442 (100.00%)
===============================================================
Finished
===============================================================
```

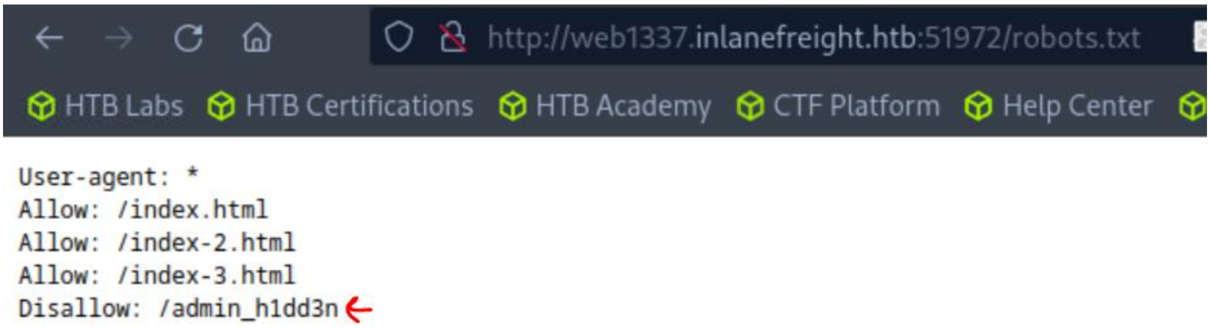
Within the results, we will observe the bruteforce found the subdomain ‘web1337’.

Now, before we proceed – its important to make sure to add the discovered subdomain to '/etc/hosts':



```
 GNU nano 7.2                                                  /etc/hosts
127.0.0.1   localhost
127.0.1.1   debian12-parrot

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1 localhost
127.0.1.1 htb-sn9pt0p5ew htb-sn9pt0p5ew.htb-cloud.com

94.237.58.91 inlanefreight.htb web1337.inlanefreight.htb
```

Once we have that, we can inspect that subdomain – lets see if it has robots.txt configuration file:

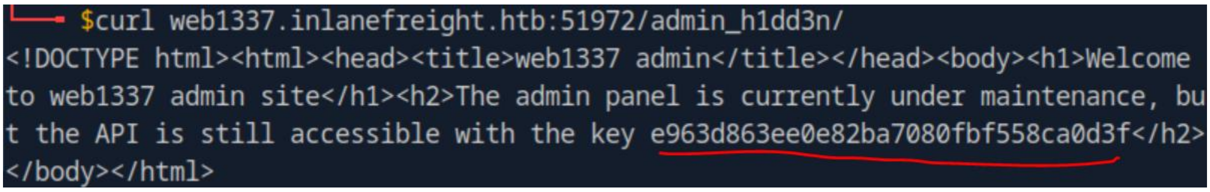


```
User-agent: *
Allow: /index.html
Allow: /index-2.html
Allow: /index-3.html
Disallow: /admin_h1dd3n
```

And there is – '/admin_h1dd3n'. crawlers are not permitted to inspects.

Now as it is directory, we wont find anything using the browser, we will use 'curl' instead:

```
curl web1337.inlanefreight.htb:<target-port>/admin_h1dd3n/
```



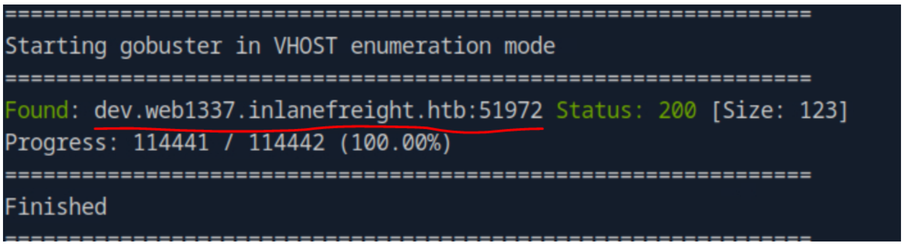
```
└──╼ $curl web1337.inlanefreight.htb:51972/admin_h1dd3n/
<!DOCTYPE html><html><head><title>web1337 admin</title></head><body><h1>Welcome
to web1337 admin site</h1><h2>The admin panel is currently under maintenance, bu
t the API is still accessible with the key e963d863ee0e82ba7080fbf558ca0d3f</h2>
</body></html>
```

**Question:** After crawling the inlanefreight.htb domain on the target system, what is the email address you have found? Respond with the full email, e.g., mail@inlanefreight.htb..

**Answer:** 1337testing@inlanefreight.htb

**Method:** let’s look for subdomains within the ‘web1337’ subdomain:

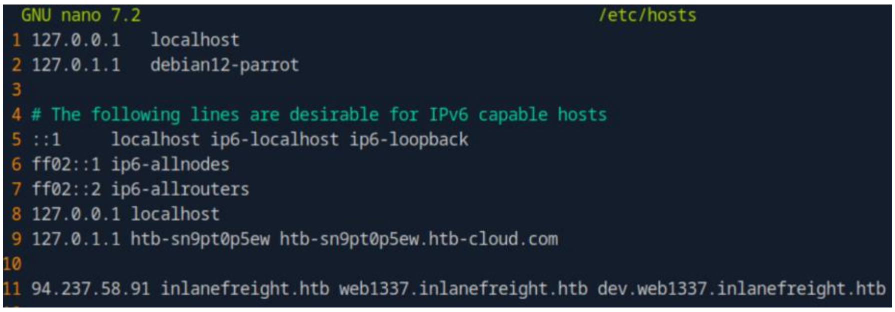
```
gobuster vhost -u http://web1337.inlanefreight.htb:<target-port> -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt --append-domain
```



```
===============================================================
Starting gobuster in VHOST enumeration mode
===============================================================
Found: dev.web1337.inlanefreight.htb:51972 Status: 200 [Size: 123]
Progress: 114441 / 114442 (100.00%)
===============================================================
Finished
===============================================================
```

We found a subdomain ‘dev.web1337.inlanefreight.htb’

Lets add it to ‘/etc/hosts’:



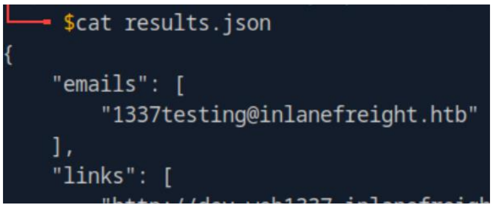
```
  GNU nano 7.2                                    /etc/hosts
127.0.0.1   localhost
127.0.1.1   debian12-parrot

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1 localhost
127.0.1.1 htb-sn9pt0p5ew htb-sn9pt0p5ew.htb-cloud.com

94.237.58.91 inlanefreight.htb web1337.inlanefreight.htb dev.web1337.inlanefreight.htb
```

When that is complete – lets run a crawler:

```
python3 ReconSpider3.py
http://dev.web1337.inlanefreight.htb:51972
```

using the same ReconSpider.py tool from ‘Creepy Crawlies’ section:

when done, an output file ‘results.json’ will be created, inspecting it will reveal the mail:

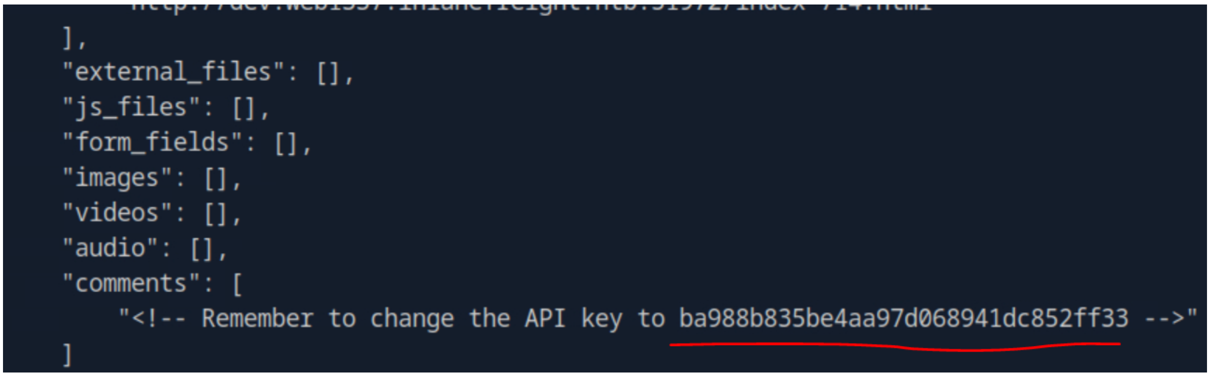


```
└──╼ $cat results.json
{
    "emails": [
        "1337testing@inlanefreight.htb"
    ],
    "links": [
        "http://dev.web1337.inlanefreigh
```

**Question:** What is the API key the inlanefreight.htb developers will be changing too?

**Answer:** ba988b835be4aa97d068941dc852ff33

**Method:** looking at the same 'results.json' from the previous question, scrolling down to the end, we will see this:



```
        http://dev.web1337.inlanefreight.htb:31972/index-714.html
    ],
    "external_files": [],
    "js_files": [],
    "form_fields": [],
    "images": [],
    "videos": [],
    "audio": [],
    "comments": [
        "<!-- Remember to change the API key to ba988b835be4aa97d068941dc852ff33 -->"
    ]
```