WEB SECURITY

Year 2 semester 2

Submitted to

Sri Lanka Institute of Information Technology

In partial fulfillment of the requirements for the Bachelor of Science Special Honors Degree in Information Technology (Cyber Security)

Student ID : IT19208268

Student Name: Poobalan A.V

Acknowledgement and Special thanks.

I certify that this report does not incorporate without acknowledgement, any material previously submitted for a degree or diploma in any university, and to the best of my knowledge believe it does not contain any material previously published or written by another person, except where due reference is made in text.

Table of Contents

Topic

https://www.koho.ca

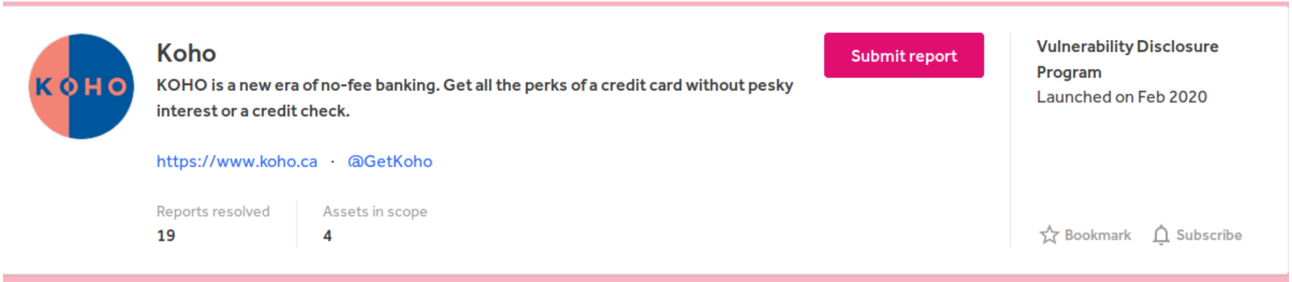
Objective

“Web audit and vulnerability assessment report of the domain”

EXECUTIVE SUMMARY

The web audit presented in this document is with relation to koho.ca which was available at Hackerone. This particular domain is registered in Canada as understood by the URL. This website allows a prepaid debit/credit card service once registered with the system. The assessment of the report was completed on 23rd of October 2020.

Link to Hackerone portal: https://hackerone.com/koho?type=team



Koho

KOHO is a new era of no-fee banking. Get all the perks of a credit card without pesky interest or a credit check.

https://www.koho.ca · @GetKoho

Reports resolved 19

Assets in scope 4

Vulnerability Disclosure Program

Launched on Feb 2020

INTENDED AUDIENCE

This assessment report was done only for educational purposes as part of the degree programme. As per the mentioned purpose the intended audience is restricted to the academics who view the report only.

ASSESSMENT OBJECTIVES

The objective of the assessment is to audit the mentioned domain "www.koho.ca" for any available vulnerabilities in its domain and its subdomains. The intended purpose was not to cause any damage to its resource or cause denial of service to interrupt the service provided as strongly emphasized and mentioned by the respective organization in the portal.

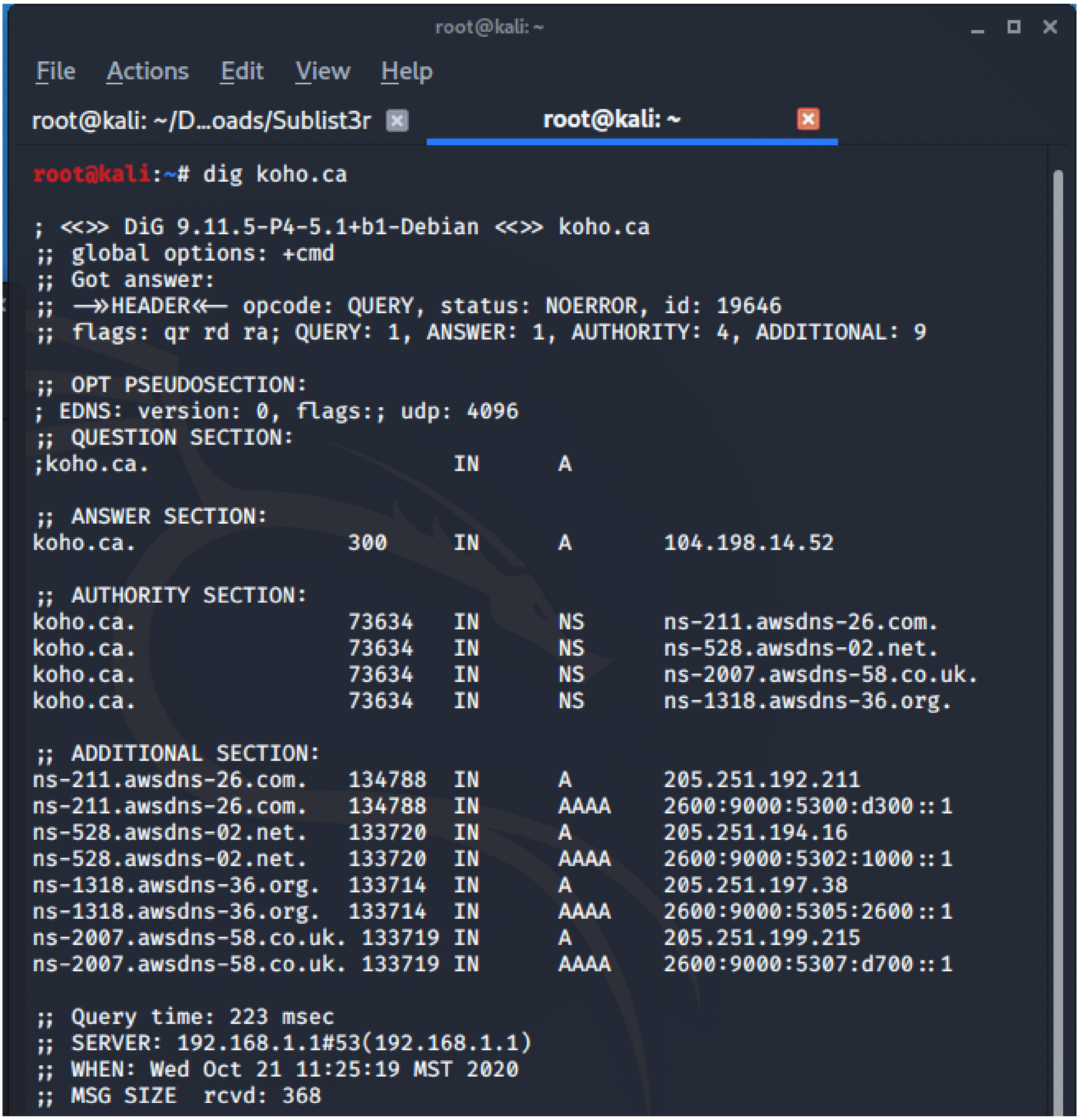
APPLICATION URL

The assessment is done on the following URL.

- https://www.koho.ca

RECONNAISSANCE

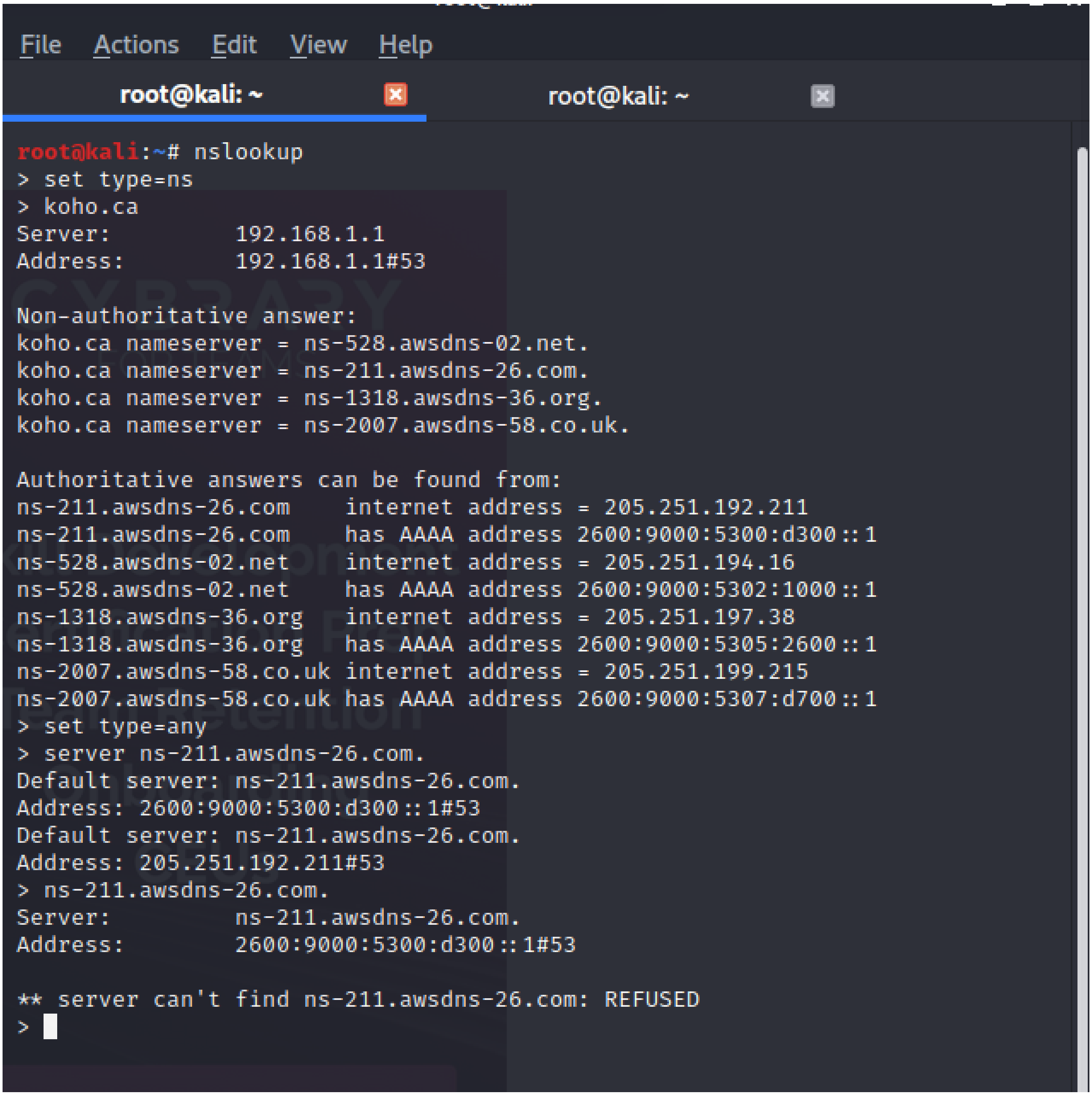
To start the assessment basic recon was needed to proceed ahead as only the domain name was given. The IP address where all the processes will be based on was easily obtained by running a dig on the domain name. Figure 2 shows results of dig.



```
root@kali:~# dig koho.ca

; <<>> DiG 9.11.5-P4-5.1+b1-Debian <<>> koho.ca
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19646
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;koho.ca.                       IN      A

;; ANSWER SECTION:
koho.ca.                300     IN      A       104.198.14.52

;; AUTHORITY SECTION:
koho.ca.                73634   IN      NS      ns-211.awsdns-26.com.
koho.ca.                73634   IN      NS      ns-528.awsdns-02.net.
koho.ca.                73634   IN      NS      ns-2007.awsdns-58.co.uk.
koho.ca.                73634   IN      NS      ns-1318.awsdns-36.org.

;; ADDITIONAL SECTION:
ns-211.awsdns-26.com.   134788  IN      A       205.251.192.211
ns-211.awsdns-26.com.   134788  IN      AAAA    2600:9000:5300:d300::1
ns-528.awsdns-02.net.   133720  IN      A       205.251.194.16
ns-528.awsdns-02.net.   133720  IN      AAAA    2600:9000:5302:1000::1
ns-1318.awsdns-36.org.  133714  IN      A       205.251.197.38
ns-1318.awsdns-36.org.  133714  IN      AAAA    2600:9000:5305:2600::1
ns-2007.awsdns-58.co.uk. 133719 IN      A       205.251.199.215
ns-2007.awsdns-58.co.uk. 133719 IN      AAAA    2600:9000:5307:d700::1

;; Query time: 223 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Wed Oct 21 11:25:19 MST 2020
;; MSG SIZE  rcvd: 368
```

[*Figure 2*]

As the IP was found proceeding ahead with it. A deep check on the domain was done with "nslookup" to look into any DNS zone transfer misconfiguration with the alias domains of the main domain. Figure 3 depicts the results obtained.

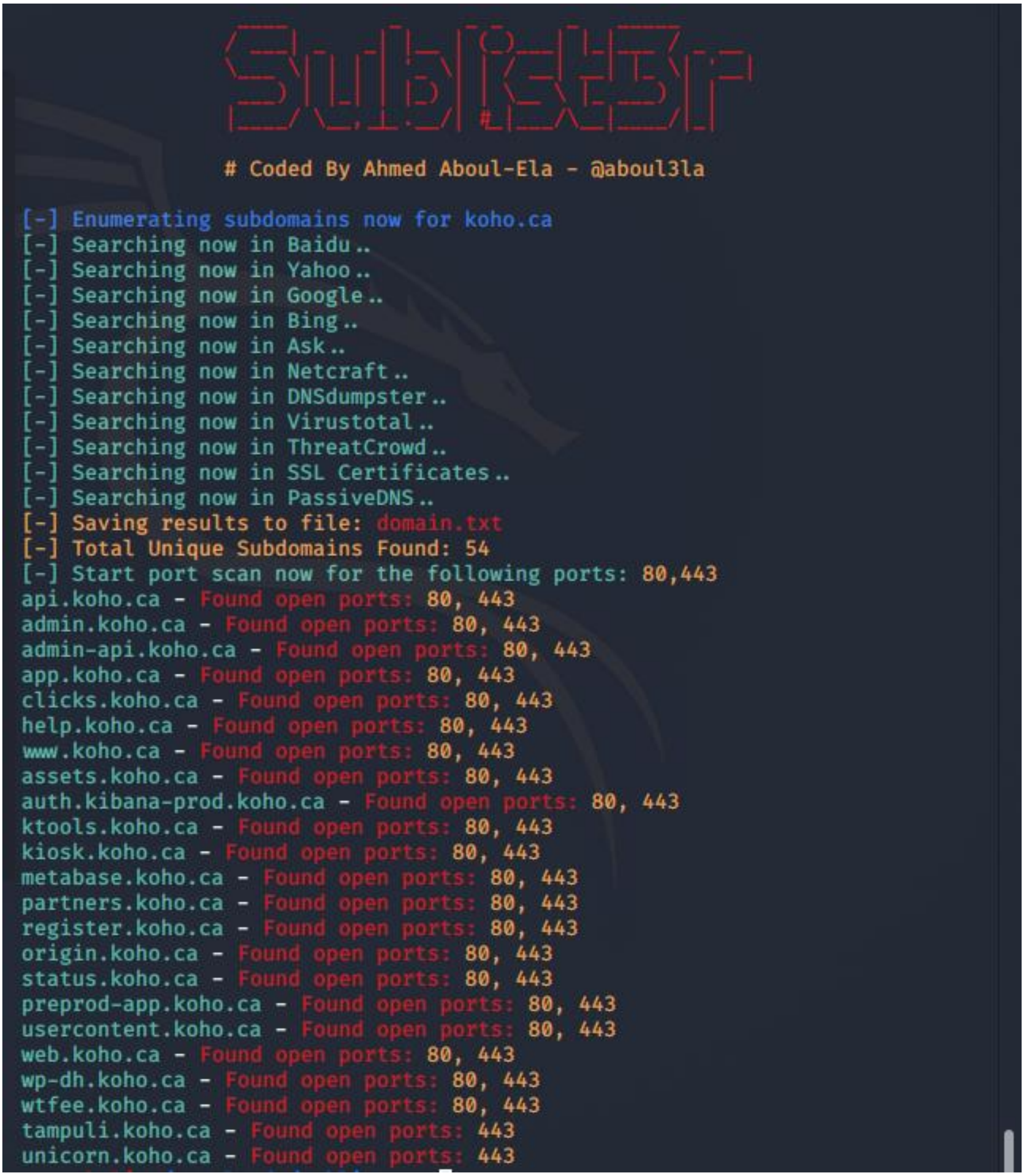


```
root@kali:~# nslookup
> set type=ns
> koho.ca
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
koho.ca nameserver = ns-528.awsdns-02.net.
koho.ca nameserver = ns-211.awsdns-26.com.
koho.ca nameserver = ns-1318.awsdns-36.org.
koho.ca nameserver = ns-2007.awsdns-58.co.uk.

Authoritative answers can be found from:
ns-211.awsdns-26.com    internet address = 205.251.192.211
ns-211.awsdns-26.com    has AAAA address 2600:9000:5300:d300::1
ns-528.awsdns-02.net    internet address = 205.251.194.16
ns-528.awsdns-02.net    has AAAA address 2600:9000:5302:1000::1
ns-1318.awsdns-36.org   internet address = 205.251.197.38
ns-1318.awsdns-36.org   has AAAA address 2600:9000:5305:2600::1
ns-2007.awsdns-58.co.uk internet address = 205.251.199.215
ns-2007.awsdns-58.co.uk has AAAA address 2600:9000:5307:d700::1
> set type=any
> server ns-211.awsdns-26.com.
Default server: ns-211.awsdns-26.com.
Address: 2600:9000:5300:d300::1#53
Default server: ns-211.awsdns-26.com.
Address: 205.251.192.211#53
> ns-211.awsdns-26.com.
Server:         ns-211.awsdns-26.com.
Address:        2600:9000:5300:d300::1#53

** server can't find ns-211.awsdns-26.com: REFUSED
> 
```

[*Figure 3*]

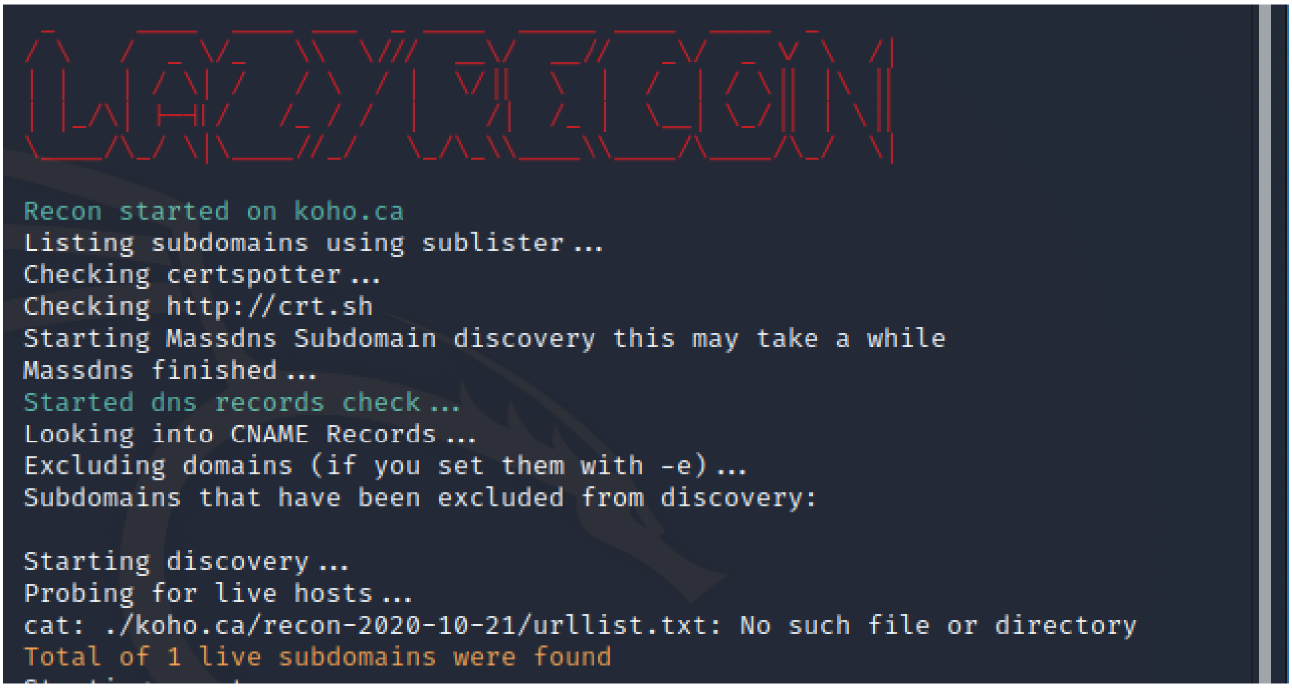
As it is very clear that the results revealed that "AWS" is managing the DNS of koho.ca further proceedings are a lost cause against the tech giant. Anyways, proceedings were carried out and the connection was refused immediately leaking NO further information.

To further examine the domain, it was necessary to know the subdomains in the system. For the particular purpose “Sublis3r” was used to list out the subdomains and it resulted with 54 domains. Figure 4 depicts results from “Sublist3r”. Although there were 54 subdomains there was only one active domain which was revealed by “lazy recon”. Figure 5 depicts results from “lazyrecon”.



```
# Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for koho.ca
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[-] Saving results to file: domain.txt
[-] Total Unique Subdomains Found: 54
[-] Start port scan now for the following ports: 80,443
api.koho.ca - Found open ports: 80, 443
admin.koho.ca - Found open ports: 80, 443
admin-api.koho.ca - Found open ports: 80, 443
app.koho.ca - Found open ports: 80, 443
clicks.koho.ca - Found open ports: 80, 443
help.koho.ca - Found open ports: 80, 443
www.koho.ca - Found open ports: 80, 443
assets.koho.ca - Found open ports: 80, 443
auth.kibana-prod.koho.ca - Found open ports: 80, 443
ktools.koho.ca - Found open ports: 80, 443
kiosk.koho.ca - Found open ports: 80, 443
metabase.koho.ca - Found open ports: 80, 443
partners.koho.ca - Found open ports: 80, 443
register.koho.ca - Found open ports: 80, 443
origin.koho.ca - Found open ports: 80, 443
status.koho.ca - Found open ports: 80, 443
preprod-app.koho.ca - Found open ports: 80, 443
usercontent.koho.ca - Found open ports: 80, 443
web.koho.ca - Found open ports: 80, 443
wp-dh.koho.ca - Found open ports: 80, 443
wtfee.koho.ca - Found open ports: 80, 443
tampuli.koho.ca - Found open ports: 443
unicorn.koho.ca - Found open ports: 443
```

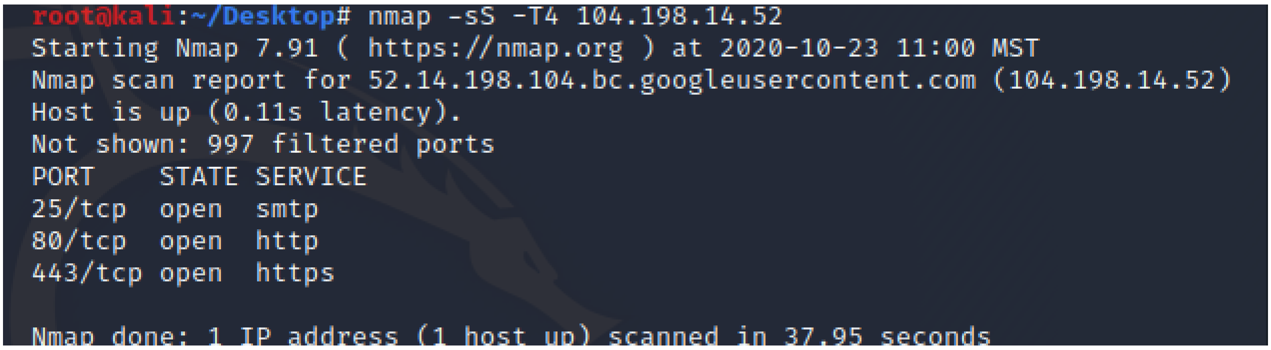
[*Figure 4*]



```
Recon started on koho.ca
Listing subdomains using sublister...
Checking certspotter...
Checking http://crt.sh
Starting Massdns Subdomain discovery this may take a while
Massdns finished...
Started dns records check...
Looking into CNAME Records...
Excluding domains (if you set them with -e)...
Subdomains that have been excluded from discovery:

Starting discovery...
Probing for live hosts...
cat: ./koho.ca/recon-2020-10-21/urllist.txt: No such file or directory
Total of 1 live subdomains were found
```

[*Figure 5*]

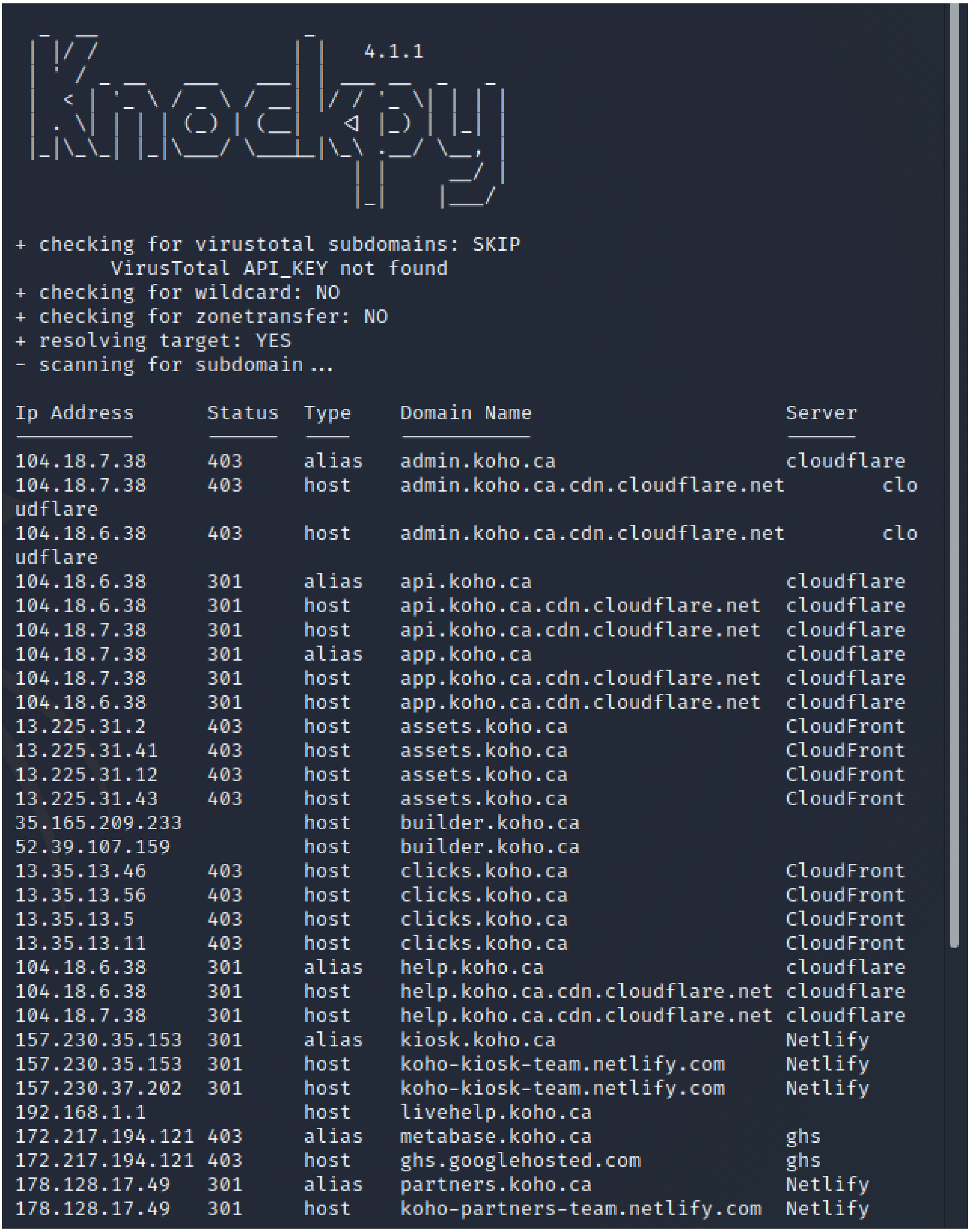
The open ports were also needed to be identified to know if there were any open ports that could be vulnerable to attacks. After seeing AWS in DNS management, nmap scan was run in stealth mode with “sS”. Figure 6 depicts results from nmap which revealed 3 open ports, http, https and smtp respectively.



```
root@kali:~/Desktop# nmap -sS -T4 104.198.14.52
Starting Nmap 7.91 ( https://nmap.org ) at 2020-10-23 11:00 MST
Nmap scan report for 52.14.198.104.bc.googleusercontent.com (104.198.14.52)
Host is up (0.11s latency).
Not shown: 997 filtered ports
PORT    STATE SERVICE
25/tcp  open  smtp
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 37.95 seconds
```

[*Figure 6*]

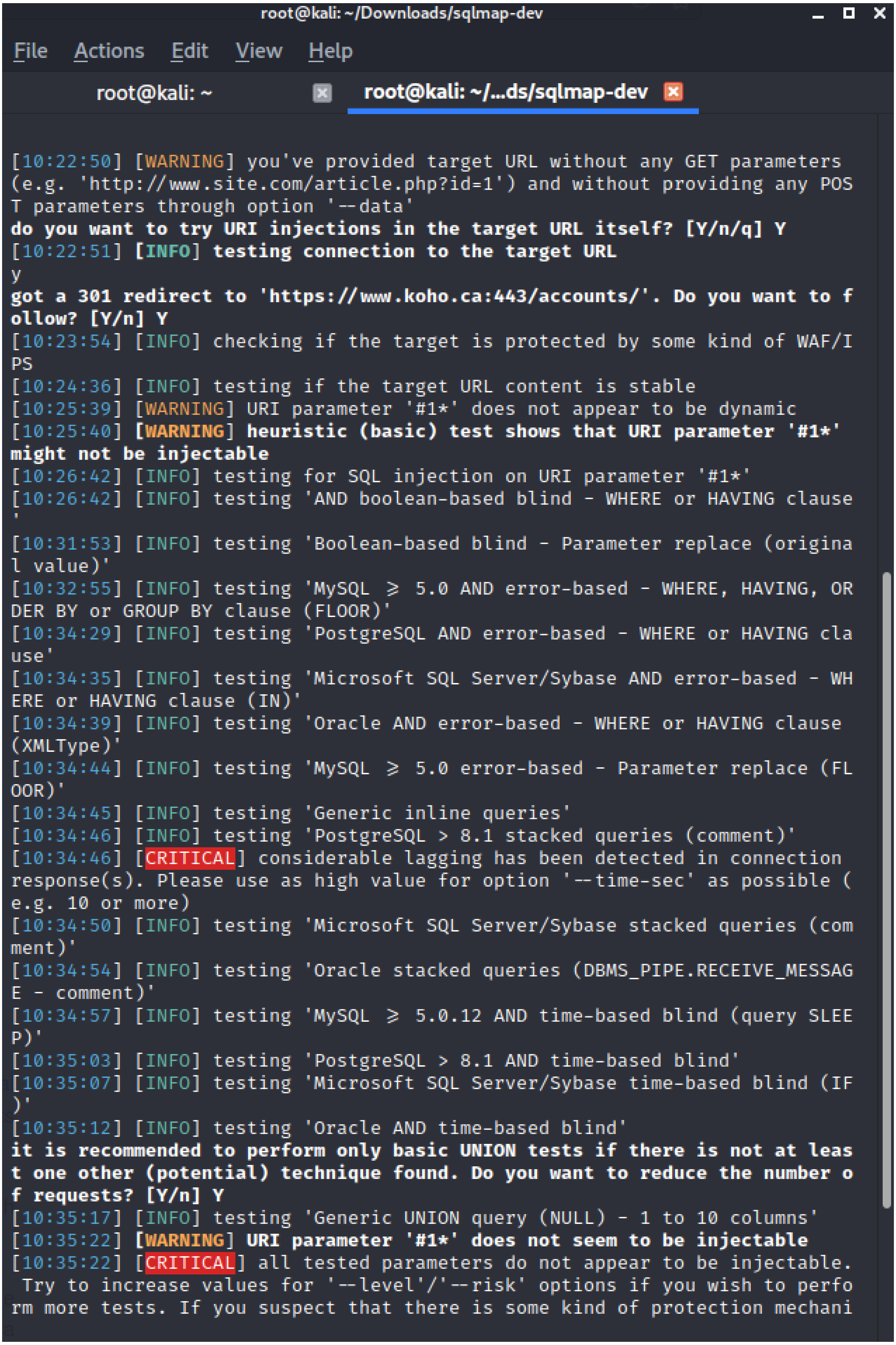
It was also necessary to find out in which cloud service were the domains deployed on and a deep search for DNS zone transfer misconfiguration and wildcards in the domains. For this purpose, “Knockpy” was used to reveal all the mentioned details. Figure 7 shows the results of Knockpy.



```
  _  __                 _
 | |/ /                | |   4.1.1
 | ' / _ __   ___   ___| | ___ __  _   _
 |  < | '_ \ / _ \ / __| |/ / '_ \| | | |
 | . \| | | | (_) | (__|   <| |_) | |_| |
 |_|\_\_| |_|\___/ \___|_|\_\ .__/ \__, |
                            | |     __/ |
                            |_|    |___/

+ checking for virustotal subdomains: SKIP
        VirusTotal API_KEY not found
+ checking for wildcard: NO
+ checking for zonetransfer: NO
+ resolving target: YES
- scanning for subdomain...

Ip Address      Status  Type    Domain Name                     Server
__________      ______  ____    ___________                     ______

104.18.7.38     403     alias   admin.koho.ca                   cloudflare
104.18.7.38     403     host    admin.koho.ca.cdn.cloudflare.net        clo
udflare
104.18.6.38     403     host    admin.koho.ca.cdn.cloudflare.net        clo
udflare
104.18.6.38     301     alias   api.koho.ca                     cloudflare
104.18.6.38     301     host    api.koho.ca.cdn.cloudflare.net  cloudflare
104.18.7.38     301     host    api.koho.ca.cdn.cloudflare.net  cloudflare
104.18.7.38     301     alias   app.koho.ca                     cloudflare
104.18.7.38     301     host    app.koho.ca.cdn.cloudflare.net  cloudflare
104.18.6.38     301     host    app.koho.ca.cdn.cloudflare.net  cloudflare
13.225.31.2     403     host    assets.koho.ca                  CloudFront
13.225.31.41    403     host    assets.koho.ca                  CloudFront
13.225.31.12    403     host    assets.koho.ca                  CloudFront
13.225.31.43    403     host    assets.koho.ca                  CloudFront
35.165.209.233          host    builder.koho.ca
52.39.107.159           host    builder.koho.ca
13.35.13.46     403     host    clicks.koho.ca                  CloudFront
13.35.13.56     403     host    clicks.koho.ca                  CloudFront
13.35.13.5      403     host    clicks.koho.ca                  CloudFront
13.35.13.11     403     host    clicks.koho.ca                  CloudFront
104.18.6.38     301     alias   help.koho.ca                    cloudflare
104.18.6.38     301     host    help.koho.ca.cdn.cloudflare.net cloudflare
104.18.7.38     301     host    help.koho.ca.cdn.cloudflare.net cloudflare
157.230.35.153  301     alias   kiosk.koho.ca                   Netlify
157.230.35.153  301     host    koho-kiosk-team.netlify.com     Netlify
157.230.37.202  301     host    koho-kiosk-team.netlify.com     Netlify
192.168.1.1             host    livehelp.koho.ca
172.217.194.121 403     alias   metabase.koho.ca                ghs
172.217.194.121 403     host    ghs.googlehosted.com            ghs
178.128.17.49   301     alias   partners.koho.ca                Netlify
178.128.17.49   301     host    koho-partners-team.netlify.com  Netlify
```

[*Figure 7*]

The results immediately revealed that the domains were hosted on Cloudflare Netlify and ghs. Any attacks against it is highly unlikely to be successful because of the industrial standard maintained by the said companies.

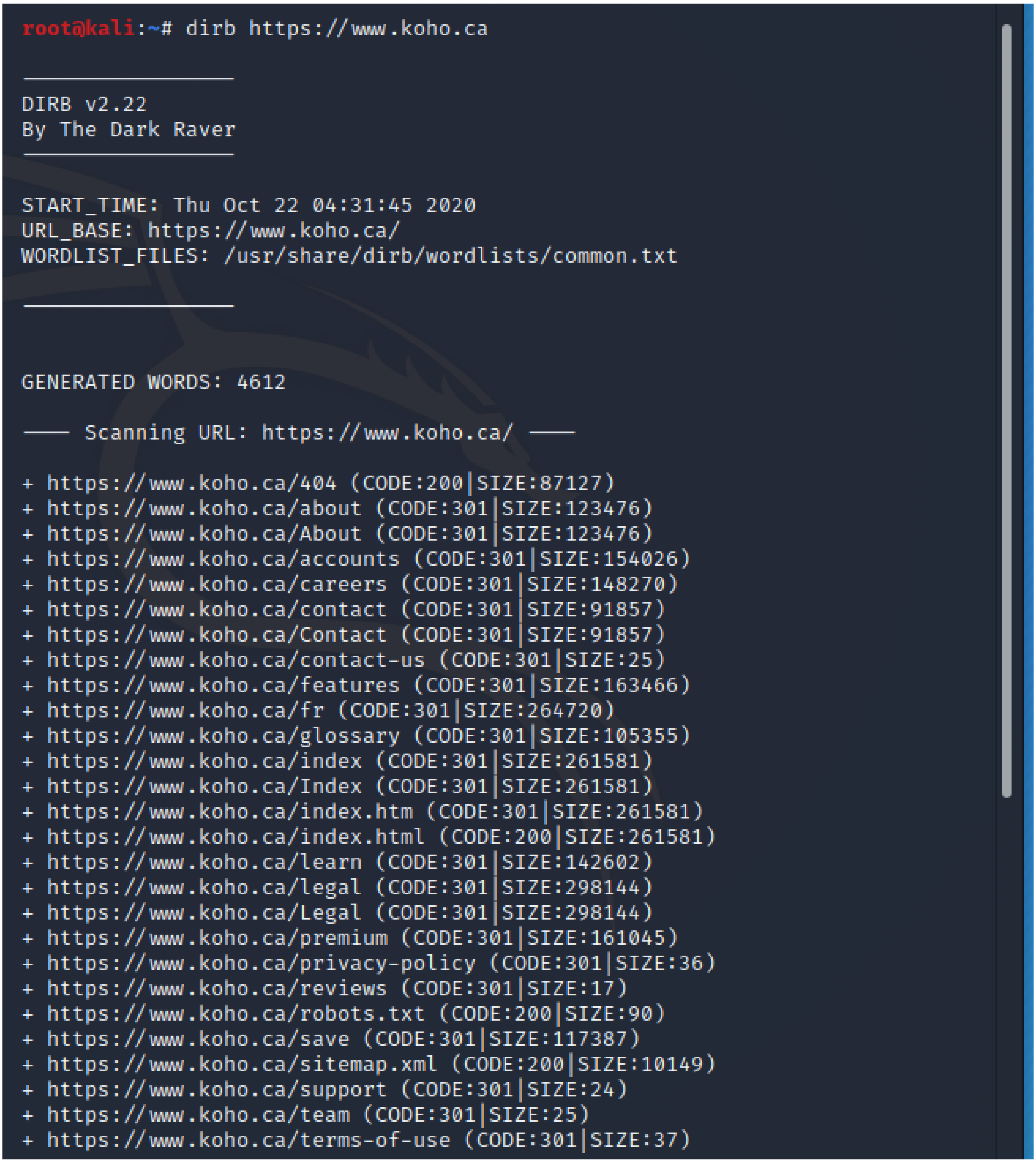

root@kali: ~/Downloads/sqlmap-dev
File
Actions
Edit
View
Help
root@kali: ~
root@kali: ~/...ds/sqlmap-dev

```
[10:22:50] [WARNING] you've provided target URL without any GET parameters
(e.g. 'http://www.site.com/article.php?id=1') and without providing any POS
T parameters through option '--data'
do you want to try URI injections in the target URL itself? [Y/n/q] Y
[10:22:51] [INFO] testing connection to the target URL
y
got a 301 redirect to 'https://www.koho.ca:443/accounts/'. Do you want to f
ollow? [Y/n] Y
[10:23:54] [INFO] checking if the target is protected by some kind of WAF/I
PS
[10:24:36] [INFO] testing if the target URL content is stable
[10:25:39] [WARNING] URI parameter '#1*' does not appear to be dynamic
[10:25:40] [WARNING] heuristic (basic) test shows that URI parameter '#1*'
might not be injectable
[10:26:42] [INFO] testing for SQL injection on URI parameter '#1*'
[10:26:42] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause
'
[10:31:53] [INFO] testing 'Boolean-based blind - Parameter replace (origina
l value)'
[10:32:55] [INFO] testing 'MySQL ≥ 5.0 AND error-based - WHERE, HAVING, OR
DER BY or GROUP BY clause (FLOOR)'
[10:34:29] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING cla
use'
[10:34:35] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WH
ERE or HAVING clause (IN)'
[10:34:39] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause
(XMLType)'
[10:34:44] [INFO] testing 'MySQL ≥ 5.0 error-based - Parameter replace (FL
OOR)'
[10:34:45] [INFO] testing 'Generic inline queries'
[10:34:46] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[10:34:46] [CRITICAL] considerable lagging has been detected in connection
response(s). Please use as high value for option '--time-sec' as possible (
e.g. 10 or more)
[10:34:50] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (com
ment)'
[10:34:54] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAG
E - comment)'
[10:34:57] [INFO] testing 'MySQL ≥ 5.0.12 AND time-based blind (query SLEE
P)'
[10:35:03] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[10:35:07] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF
)'
[10:35:12] [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at leas
t one other (potential) technique found. Do you want to reduce the number o
f requests? [Y/n] Y
[10:35:17] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[10:35:22] [WARNING] URI parameter '#1*' does not seem to be injectable
[10:35:22] [CRITICAL] all tested parameters do not appear to be injectable.
 Try to increase values for '--level'/'--risk' options if you wish to perfo
rm more tests. If you suspect that there is some kind of protection mechani
```

[*Figure 8*]

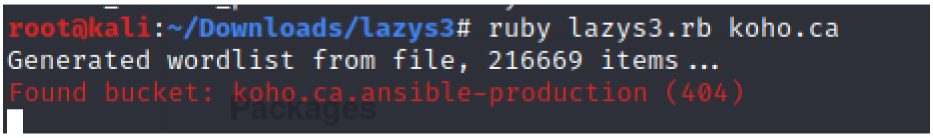
As shown above in figure 8 in the sqlmap-dev tool which checks for sql injection too returned no positive results so its highly unlikely to run a sql injection on the domain.

Since it was managed by AWS. A tool specialized to brute-force AWS hosted web apps “lazys3” was used to determine where such word-lists can be used against. For the generation of word-lists “Dirb” was used which also revealed directories in the website.
Figure 9 depicts the results fetched by “Dirb” and figure 10 shows the results returned by “lazys3” using the word-lists generated by “Dirb”.



```
root@kali:~# dirb https://www.koho.ca

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Thu Oct 22 04:31:45 2020
URL_BASE: https://www.koho.ca/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------


GENERATED WORDS: 4612

---- Scanning URL: https://www.koho.ca/ ----

+ https://www.koho.ca/404 (CODE:200|SIZE:87127)
+ https://www.koho.ca/about (CODE:301|SIZE:123476)
+ https://www.koho.ca/About (CODE:301|SIZE:123476)
+ https://www.koho.ca/accounts (CODE:301|SIZE:154026)
+ https://www.koho.ca/careers (CODE:301|SIZE:148270)
+ https://www.koho.ca/contact (CODE:301|SIZE:91857)
+ https://www.koho.ca/Contact (CODE:301|SIZE:91857)
+ https://www.koho.ca/contact-us (CODE:301|SIZE:25)
+ https://www.koho.ca/features (CODE:301|SIZE:163466)
+ https://www.koho.ca/fr (CODE:301|SIZE:264720)
+ https://www.koho.ca/glossary (CODE:301|SIZE:105355)
+ https://www.koho.ca/index (CODE:301|SIZE:261581)
+ https://www.koho.ca/Index (CODE:301|SIZE:261581)
+ https://www.koho.ca/index.htm (CODE:301|SIZE:261581)
+ https://www.koho.ca/index.html (CODE:200|SIZE:261581)
+ https://www.koho.ca/learn (CODE:301|SIZE:142602)
+ https://www.koho.ca/legal (CODE:301|SIZE:298144)
+ https://www.koho.ca/Legal (CODE:301|SIZE:298144)
+ https://www.koho.ca/premium (CODE:301|SIZE:161045)
+ https://www.koho.ca/privacy-policy (CODE:301|SIZE:36)
+ https://www.koho.ca/reviews (CODE:301|SIZE:17)
+ https://www.koho.ca/robots.txt (CODE:200|SIZE:90)
+ https://www.koho.ca/save (CODE:301|SIZE:117387)
+ https://www.koho.ca/sitemap.xml (CODE:200|SIZE:10149)
+ https://www.koho.ca/support (CODE:301|SIZE:24)
+ https://www.koho.ca/team (CODE:301|SIZE:25)
+ https://www.koho.ca/terms-of-use (CODE:301|SIZE:37)
```

[*Figure 9*]

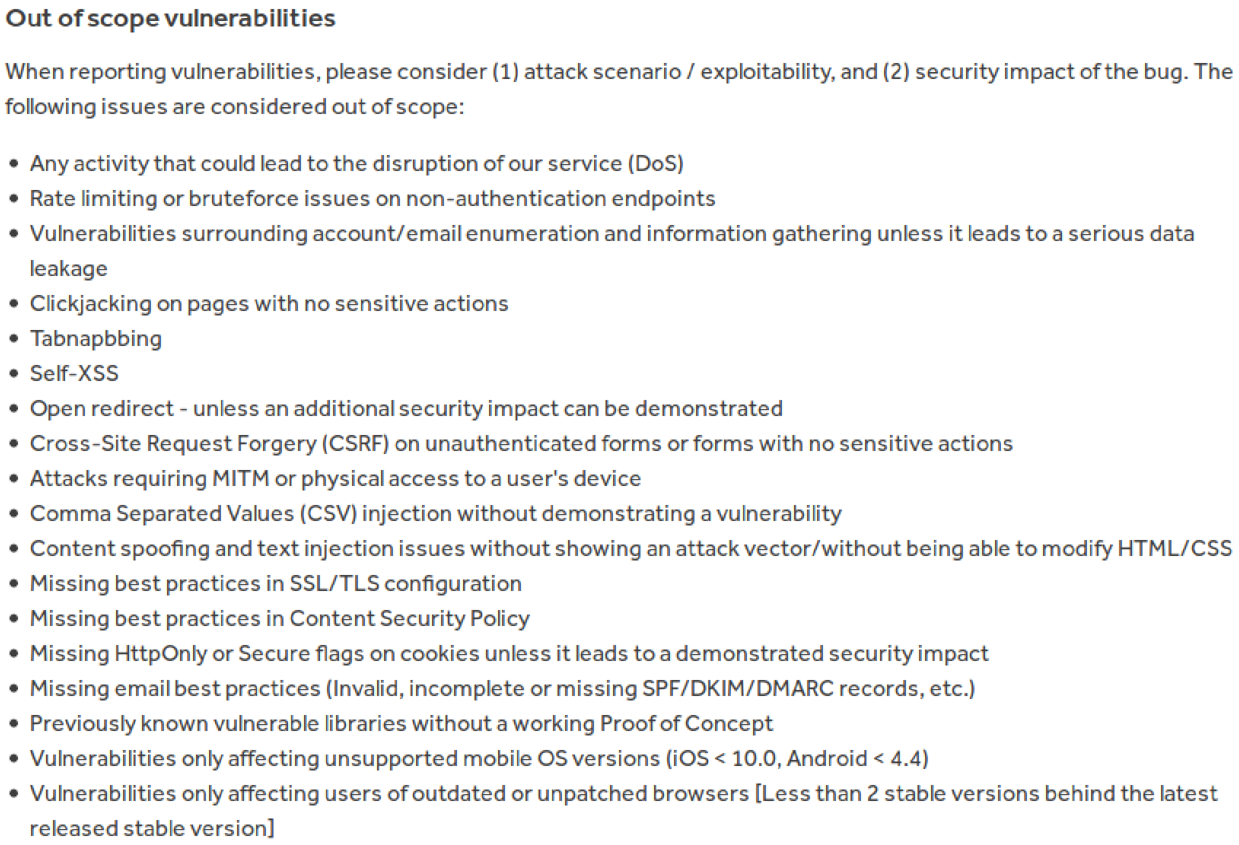


```
root@kali:~/Downloads/lazys3# ruby lazys3.rb koho.ca
Generated wordlist from file, 216669 items...
Found bucket: koho.ca.ansible-production (404)
```

[*Figure 10*]

As per the above results the wordlist mentioned found a bucket at the domain "koho.ca.ansible-production" although the error code shows a redirection to "NOT FOUND". The process can be further modified and attacked which would cause a denial of service in the system which was strongly opposed by the organization.

Figure 11 attached below shows the conditions set forth by the organization as out of scope in the Hackerone portal.



Out of scope vulnerabilities

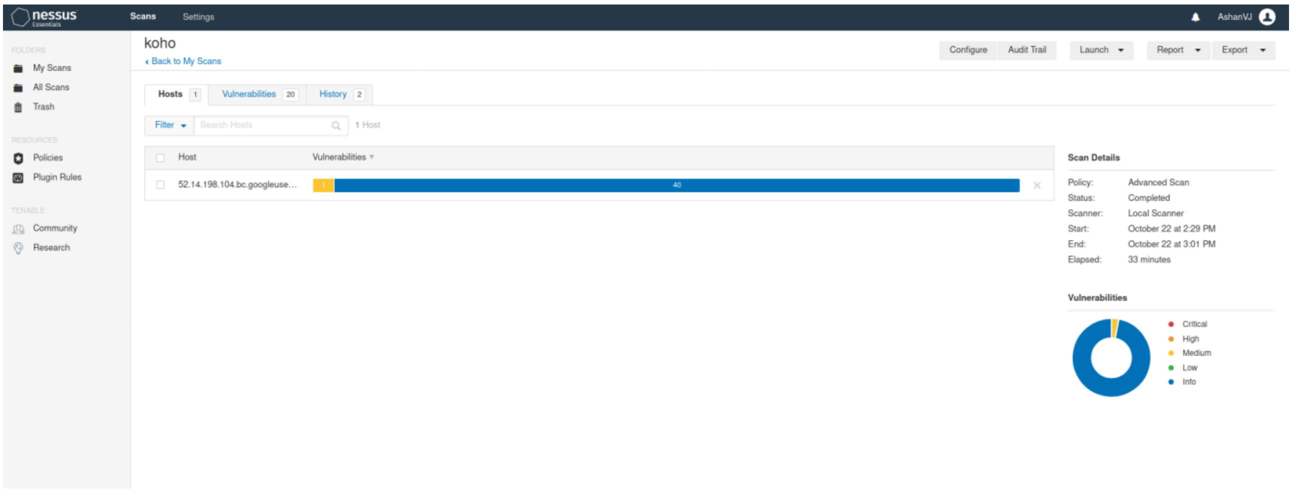
When reporting vulnerabilities, please consider (1) attack scenario / exploitability, and (2) security impact of the bug. The following issues are considered out of scope:

- Any activity that could lead to the disruption of our service (DoS)
- Rate limiting or bruteforce issues on non-authentication endpoints
- Vulnerabilities surrounding account/email enumeration and information gathering unless it leads to a serious data leakage
- Clickjacking on pages with no sensitive actions
- Tabnapbbing
- Self-XSS
- Open redirect - unless an additional security impact can be demonstrated
- Cross-Site Request Forgery (CSRF) on unauthenticated forms or forms with no sensitive actions
- Attacks requiring MITM or physical access to a user's device
- Comma Separated Values (CSV) injection without demonstrating a vulnerability
- Content spoofing and text injection issues without showing an attack vector/without being able to modify HTML/CSS
- Missing best practices in SSL/TLS configuration
- Missing best practices in Content Security Policy
- Missing HttpOnly or Secure flags on cookies unless it leads to a demonstrated security impact
- Missing email best practices (Invalid, incomplete or missing SPF/DKIM/DMARC records, etc.)
- Previously known vulnerable libraries without a working Proof of Concept
- Vulnerabilities only affecting unsupported mobile OS versions (iOS < 10.0, Android < 4.4)
- Vulnerabilities only affecting users of outdated or unpatched browsers [Less than 2 stable versions behind the latest released stable version]

[*Figure 11*]

SCAN

Two automated scanning tools were used for the scanning of the vulnerabilities for the particular domain. Figure 12 shows results from “Nessus” and figure 13 shows results from “Owasp ZAP”



koho

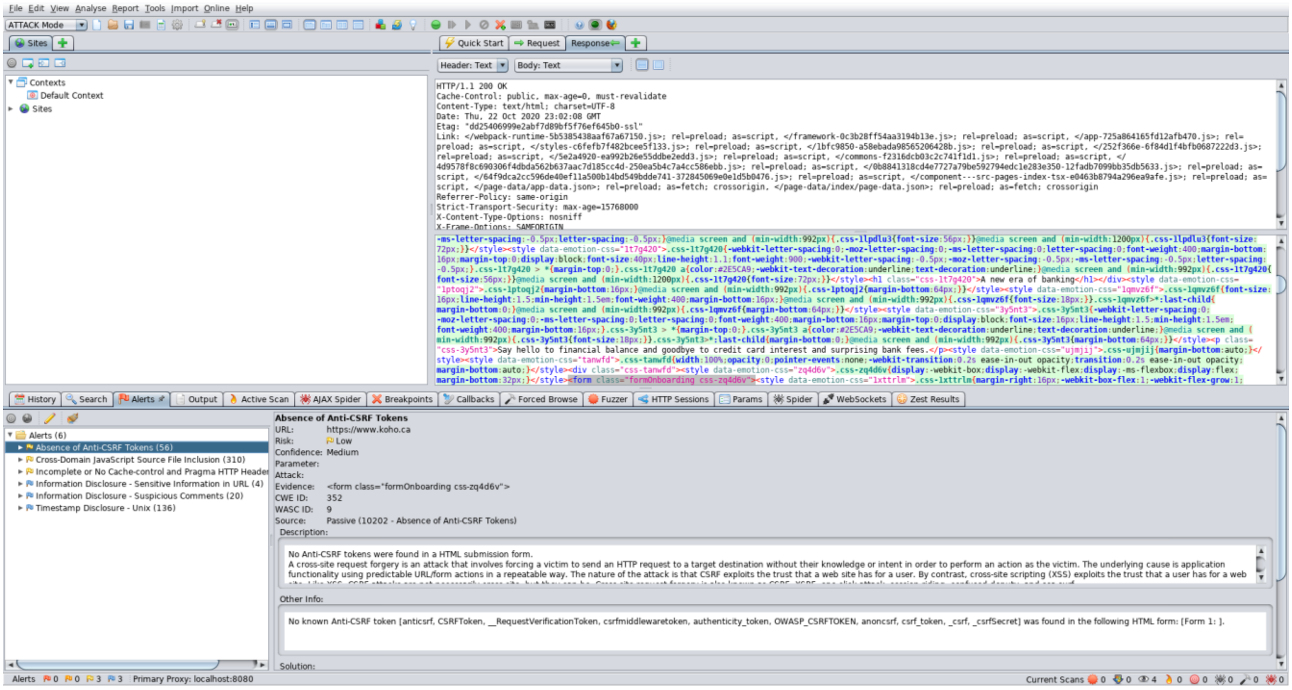
Host	Vulnerabilities
52.14.198.104.bc.googleuse...	1 40

Scan Details

Policy: Advanced Scan
Status: Completed
Scanner: Local Scanner
Start: October 22 at 2:29 PM
End: October 22 at 3:01 PM
Elapsed: 33 minutes

Vulnerabilities

[*Figure 12*]



```
HTTP/1.1 200 OK
Cache-Control: public, max-age=0, must-revalidate
Content-Type: text/html; charset=UTF-8
Date: Thu, 22 Oct 2020 23:02:08 GMT
Etag: "dd25406999e2abf7d89bf5f76ef645b0-ssl"
Link: </webpack-runtime-5b5385438aaf67a67150.js>; rel=preload; as=script, </framework-0c3b28ff54aa3194b13e.js>; rel=preload; as=script, </app-725a864165fd12afb470.js>; rel=
preload; as=script, </styles-c6fefb7f482bcee5f133.js>; rel=preload; as=script, </1bfc9850-a58ebada98565206428b.js>; rel=preload; as=script, </252f366e-6f84d1f4bfb0687222d3.js>;
rel=preload; as=script, </5e2a4920-ea992b26e55ddbe2edd3.js>; rel=preload; as=script, </commons-f2316dcb03c2c741f1d1.js>; rel=preload; as=script, </
4d9578f8c690306f4dbda562b637aac7d185cc4d-250ea5b4c7a4cc586ebb.js>; rel=preload; as=script, </0b8841318cd4e7727a79be592794edc1e283e350-12fadb7099bb35db5633.js>; rel=preload; as=
script, </64f9dca2cc596de40ef11a500b14bd549bdde741-372845069e0e1d5b0476.js>; rel=preload; as=script, </component---src-pages-index-tsx-e0463b8794a296ea9afe.js>; rel=preload; as=
script, </page-data/app-data.json>; rel=preload; as=fetch; crossorigin, </page-data/index/page-data.json>; rel=preload; as=fetch; crossorigin
Referrer-Policy: same-origin
Strict-Transport-Security: max-age=15768000
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
```

```
-ms-letter-spacing:-0.5px;letter-spacing:-0.5px;}@media screen and (min-width:992px){.css-1lpdlu3{font-size:56px;}}@media screen and (min-width:1200px){.css-1lpdlu3{font-size:
72px;}}</style><style data-emotion-css="1t7g420">.css-1t7g420{-webkit-letter-spacing:0;-moz-letter-spacing:0;-ms-letter-spacing:0;letter-spacing:0;font-weight:400;margin-bottom:
16px;margin-top:0;display:block;font-size:40px;line-height:1.1;font-weight:900;-webkit-letter-spacing:-0.5px;-moz-letter-spacing:-0.5px;-ms-letter-spacing:-0.5px;letter-spacing:
-0.5px;}.css-1t7g420 > *{margin-top:0;}.css-1t7g420 a{color:#2E5CA9;-webkit-text-decoration:underline;text-decoration:underline;}@media screen and (min-width:992px){.css-1t7g420{
font-size:56px;}}@media screen and (min-width:1200px){.css-1t7g420{font-size:72px;}}</style><h1 class="css-1t7g420">A new era of banking</h1></div><style data-emotion-css=
"1ptoqj2">.css-1ptoqj2{margin-bottom:16px;}@media screen and (min-width:992px){.css-1ptoqj2{margin-bottom:64px;}}</style><style data-emotion-css="1qmvz6f">.css-1qmvz6f{font-size:
16px;line-height:1.5;min-height:1.5em;font-weight:400;margin-bottom:16px;}@media screen and (min-width:992px){.css-1qmvz6f{font-size:18px;}}.css-1qmvz6f>*:last-child{
margin-bottom:0;}@media screen and (min-width:992px){.css-1qmvz6f{margin-bottom:64px;}}</style><style data-emotion-css="3y5nt3">.css-3y5nt3{-webkit-letter-spacing:0;
-moz-letter-spacing:0;-ms-letter-spacing:0;letter-spacing:0;font-weight:400;margin-bottom:16px;margin-top:0;display:block;font-size:16px;line-height:1.5;min-height:1.5em;
font-weight:400;margin-bottom:16px;}.css-3y5nt3 > *{margin-top:0;}.css-3y5nt3 a{color:#2E5CA9;-webkit-text-decoration:underline;text-decoration:underline;}@media screen and (
min-width:992px){.css-3y5nt3{font-size:18px;}}.css-3y5nt3>*:last-child{margin-bottom:0;}@media screen and (min-width:992px){.css-3y5nt3{margin-bottom:64px;}}</style><p class=
"css-3y5nt3">Say hello to financial balance and goodbye to credit card interest and surprising bank fees.</p><style data-emotion-css="ujmjij">.css-ujmjij{margin-bottom:auto;}</
style><style data-emotion-css="tanwfd">.css-tanwfd{width:100%;opacity:0;pointer-events:none;-webkit-transition:0.2s ease-in-out opacity;transition:0.2s ease-in-out opacity;
margin-bottom:auto;}</style><div class="css-tanwfd"><style data-emotion-css="zq4d6v">.css-zq4d6v{display:-webkit-box;display:-webkit-flex;display:-ms-flexbox;display:flex;
margin-bottom:32px;}</style><form class="formOnboarding css-zq4d6v"><style data-emotion-css="1xttrlm">.css-1xttrlm{margin-right:16px;-webkit-box-flex:1;-webkit-flex-grow:1;
```

Alerts (6)
- Absence of Anti-CSRF Tokens (56)
- Cross-Domain JavaScript Source File Inclusion (310)
- Incomplete or No Cache-control and Pragma HTTP Header
- Information Disclosure - Sensitive Information in URL (4)
- Information Disclosure - Suspicious Comments (20)
- Timestamp Disclosure - Unix (136)

Absence of Anti-CSRF Tokens

URL: https://www.koho.ca
Risk: Low
Confidence: Medium
Parameter:
Attack:
Evidence: <form class="formOnboarding css-zq4d6v">
CWE ID: 352
WASC ID: 9
Source: Passive (10202 - Absence of Anti-CSRF Tokens)

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

Other Info:

No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret] was found in the following HTML form: [Form 1:].

Solution:

Alerts 0 0 3 3 Primary Proxy: localhost:8080

Current Scans 0 0 4 0 0 0 0 0

[*Figure 13*]

VULNERABILITIES AND RECOMMENDATION

Vulnerability: HSTS Missing From HTTPS Server

Synopsis: The remote web server is not enforcing HSTS.

Description: The remote HTTPS server is not enforcing HTTP Strict Transport Security (HSTS). HSTS is an optional response header that can be configured on the server to instruct the browser to only communicate via HTTPS. The lack of HSTS allows downgrade attacks, SSL-stripping man-in-the-middle attacks, and weakens cookie hijacking protections

Severity: Medium

Recommendation: Configure the remote web server to use HSTS.

Vulnerability: Cross-Domain JavaScript Source File Inclusion

Description: The page includes one or more script files from a third-party domain

Severity: Medium

Evidence:

```
<script
src="https://cdn.polyfill.io/v3/polyfill.min.js?features=ResizeObserver,smoothscroll"></script>
<script
src="https://cdn.polyfill.io/v3/polyfill.min.js?features=ResizeObserver,smoothscroll"></script>
```

Recommendation: Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.

Vulnerability: Incomplete or No Cache-control and Pragma HTTP Header Set

Description: The cache-control and pragma HTTP header have not been set properly or are missing allowing the browser and proxies to cache content.

Severity: Medium

Evidence: public, max-age=0, must-revalidate

Recommendation: Whenever possible ensure the cache-control HTTP header is set with no-cache, no-store, must-revalidate; and that the pragma HTTP header is set with no-cache.

Vulnerability: Absence of Anti-CSRF Tokens

Synopsis: No Anti-CSRF tokens were found in a HTML submission form

Description: 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.

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.

Severity: Medium

Evidence: <form class="formOnboarding css-zq4d6v">

Recommendation:
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.

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).
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.
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.

Vulnerability: HTTP Methods Allowed (per directory)

Synopsis: Determines which HTTP methods are allowed on various CGI directories.

Description:
By calling the OPTIONS method, it is possible to determine which HTTP methods are allowed on each directory. The following HTTP methods are considered insecure: PUT, DELETE, CONNECT, TRACE, HEAD Many frameworks and languages treat 'HEAD' as a 'GET' request, albeit one without any body in the response. If a security constraint was set on 'GET' requests such that only 'authenticatedUsers' could access GET requests for a particular servlet or resource, it would be bypassed for the 'HEAD' version. This allowed unauthorized blind submission of any privileged GET request. As this list may be incomplete, the plugin also tests - if 'Thorough tests' are enabled or 'Enable web applications tests' is set to 'yes' in the scan policy - various known HTTP methods on each directory and considers them as unsupported if it receives a response code of 400, 403, 405, or 501.

Severity: INFO

Evidence: Based on tests of each method : -
HTTP methods DELETE GET HEAD OPTIONS PATCH POST PUT are allowed on :
/
/auth

Vulnerability: Web Server Directory Enumeration

Synopsis: Possibility to enumerate directories on the web server.

Description: Attempts to determine the presence of various common directories on the remote web server. By sending a request for a directory, the web server response code indicates if it is a valid directory or not.

Severity: INFO

Evidence: The following directories were discovered: /auth

Recommendation: Redirect to 404 error code

Vulnerability: OS Identification

Synopsis: Possible to guess the remote operating system

Description: Using a combination of remote probes (e.g., TCP/IP, SMB, HTTP, NTP, SNMP, etc.), it is possible to guess the name of the remote operating system in use. It is also possible sometimes to guess the version of the operating system.

Severity: INFO

Evidence:
Remote operating system : Linux Kernel 2.4
Linux Kernel 2.6 Confidence level : 54
Method : SinFP
The remote host is running one of these operating systems :
Linux Kernel 2.4
Linux Kernel 2.6

Recommendation: Use solutions like IP Personality to modify the stack behavior and let scanners make a wrong guess.

Vulnerability: SSL Certificate Signed Using Weak Hashing Algorithm (Known CA)

Synopsis: A known CA SSL certificate in the certificate chain has been signed using a weak hashing algorithm.

Description: The remote service uses a known CA certificate in the SSL certificate chain that has been signed using a cryptographically weak hashing algorithm (e.g., MD2, MD4, MD5, or SHA1). These signature algorithms are known to be vulnerable to collision attacks. An attacker can exploit this to generate another certificate with the same digital signature, allowing the attacker to masquerade as the affected service. Note that this plugin reports all SSL certificate chains signed with SHA-1 that expire after January 1, 2017 as vulnerable. This is in accordance with Google's gradual sunsetting of the SHA-1 cryptographic hash algorithm.

Severity: INFO

Evidence:
-Subject : C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert Global Root CA |-Signature Algorithm : SHA-1 With RSA Encryption
-Valid From : Nov 10 00:00:00 2006 GMT
-Valid To : Nov 10 00:00:00 2031 GMT

Recommendation: Contact the Certificate Authority to have the certificate reissued

Link to Video: https://mysliit-my.sharepoint.com/:f:/g/personal/it19208268_my_sliit_lk/Esr3h29TkS1Ph5AfcDzo2a4B8sUS8aHOVN4i4afvPwgqgg?e=DgO9km

Thank You!