Srilanka Institute of Information Technology

Web Application Security Audit

Assessment Report

IE2062 – Web Security

Submitted by:

Student Registration Number	Student Name
Sulaxshayan.N	IT20660116

Date of submission: 05/06/2022

Bug Bounty

We are instructed to perform a bug bounty program as our web security assignment at legal websites such as Hackerone, Bugcrowd, and initigrity. Initially, to succeed in these programs, professionals used to go into the approach called 'Hackers Methodology'. Therefore, we also decided to go like a professional; how he/she approaches the target.

1. Reconnaissance.
2. Enumeration.
3. Gaining access.
4. Privilege escalation (or privacy exploration).
5. Covering tracks.

These are the tools that we going to use for this bug bounty program.

Reconnaissance.

This is the initial phase of this Methodology; People also call it **Recon**. In this phase, we are not interacting with the target overall but, we collect information about the target.

1. Who. is
2. Crt.h
3. Maltego.
4. Knock
5. Sudomy.
6. Nmap.

Enumeration

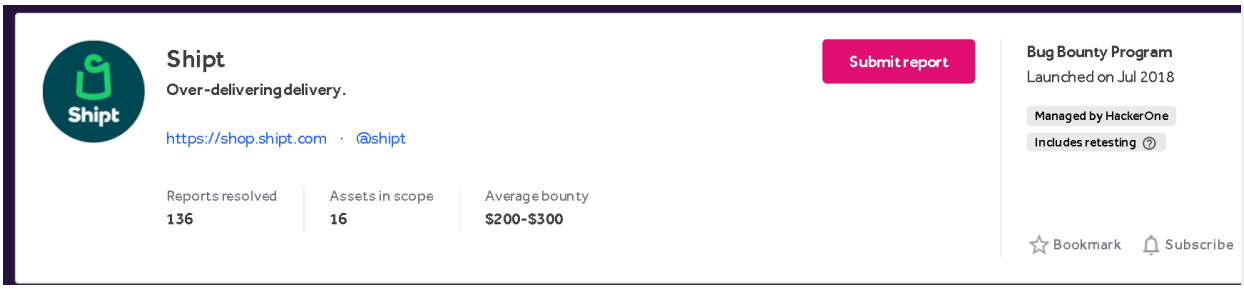
In this phase, attackers try to interact with the target and start attempting to find vulnerabilities related to that target. Attackers started to interact with the target from all of their attack surfaces. If we simply say, this is the phase where we find the target system is vulnerable to which kind of attack.

1. Burp Suite.
2. PWN XSS.

3. XSpear.
4. Nikto.
5. OWASP ZAP.
6. Skipfish

BOUNTY PROGRAM INFORMATION

I choose the shipt program from the Hackerone platform to do my bug bounty.



These are the program rules:

Program Rules

Do:

- Abide by these Shipt Bug Bounty Program Terms & Conditions.
- Comply with all applicable laws.
- Be respectful when interacting with Shipt Staff/Team members and HackerOne Triage members.
- Only conduct testing with accounts you own.
  - Test accounts must comply with testing conditions (Researchers must use @wearehackerone accounts.)
- Immediately stop conducting research/testing when unsure. If you think you may have caused damage with testing a vulnerability, please report your initial findings and request authorization to continue testing.

Items in the following section are also Out of Scope of our Bug Bounty Program.

Do NOT:

- Access, process or destroy Shipt employee data, customer data or Shipt intellectual property.
- Mass create accounts to perform testing against our applications or services.
- Publicly disclose vulnerabilities.
- Make any attempts to extort us.
- Run automated destructive testing.
- Leave any of our systems in a more vulnerable state than you found them.
- Upload shells, scripts, or create a backdoor of any kind.
- Conduct Brute Force attacks or guess credentials against user accounts to gain access to our Systems.
- Interact with accounts you do not own.
- Make changes to/or save any Shipt data that is not yours or that you do not have explicit written permission to manipulate.
  - *If prompted to change a password of an account not registered by yourself or provided to you, stop conducting research and report the finding immediately.*
- Conduct activities to be considered a privacy violation or destruction of data, such as exfiltrating or destroying data for any account that is not yours or that you do not have explicit written permission to manipulate.
- Conduct tests that disrupt service to Shipt users or employees.
  - *Examples include, but are not limited to: Denial of service, Distributed Denial of Service, DNS Spoofing, Buffer Overflow, Ping of Death, Syn Flood, Teardrop.*
- Make attempts in any form of social engineering.
  - *Examples include, but are not limited to: Phishing (Angler, Spear), Whaling, Diversion Theft, Baiting, Honey Trap, Pretexting, SMS Phishing, Scareware, Watering Hole, etc.*
- Engage in any form of physical penetration testing against Shipt properties or offices.

And these are the program scopes and out-of-scope,

In Scope

While we expect to expand our scope in the future, we are currently focused on our primary and critical public facing sites and applications. However, if you enumerate other Shipt assets and identify vulnerabilities against those, your reports may still be eligible for a bounty, so please responsibly disclose those to us as well.
If you identify a vulnerability within a third party that Shipt uses and it is not explicitly in scope, please report it to the third party's responsible disclosure program, bug bounty program, or security team. If you report it to us, we will be happy to assist you in reporting it to the third party, however, these reports usually will not be eligible for bounty from Shipt (unless special conditions are met such as the root cause being a misconfiguration by Shipt and not under the control of the 3rd party).

Out of Scope

In addition to our “Do Not” rules section, items below are out of scope.

- API key disclosure or leak without POC of exploitability.
- Internal IP Address disclosures without any proof of exploitation.
- Stack traces or error messages in staging environments without proof of exploitation.
- Any activity that could lead to the denial of service (DoS).
- Clickjacking on pages with no sensitive actions.
- Content spoofing and text injection issues without showing an attack vector (DO NOT actively engage in social engineering or phishing attempts to our employees or support staff via chat, email, or phone for POCs).
- Invalid or missing SPF (Sender Policy Framework), DMARC, or DKIM records/configurations.
- Attacks requiring MiTM or physical access to a users device
- Missing best practices in SSL/TLS configuration.
- Password and account recovery policies, such as reset link expiration or password complexity.
- Rate limit attacks or spamming.
- Unauthenticated or logout CSRF.
- User enumeration.
- "Information Disclosure" involving a userID or email address in a query parameter on a POST request
- Vulnerabilities only affecting users of outdated or unpatched browsers and platforms.
- Previously known vulnerable libraries or software versions without a working Proof of Concept.
- Any CORS-related vulnerabilities without working POC showing sensitive actions or data being affected.
- Self XSS or XSS that affects only out-of-date browsers or devices that require the user to configure them in an insecure state (allowing installation of applications from untrusted sources).
- Attacks predicated on a user's non-Shipt account being compromised (e.g. Attackers compromise the users Google/Gmail first before the vulnerability can be exploited).

These are the domain in the scope list,

Type	Identifier	Max. Severity	In Scope	Bounty
Domain	app.shipt.com	Critical	Yes	Yes
Domain	api.shipt.com	Critical	Yes	Yes
Domain	admin.shipt.com	Critical	Yes	Yes
Domain	shop.shipt.com	Critical	Yes	Yes
Domain	staging-api.shipt.com	Critical	Yes	Yes
Domain	staging-shop.shipt.com	Critical	Yes	Yes
Domain	staging-app.shipt.com	Critical	Yes	Yes
Domain	staging-admin.shipt.com	Critical	Yes	Yes
Domain	staging-shoppingcart.shipt.com	Critical	Yes	Yes
Domain	shoppingcart.shipt.com	Critical	Yes	Yes
Domain	www.shipt.com	Critical	Yes	Yes
Domain	*.shipt.com	Critical	Yes	Yes
Android: Play Store	com.shipt.groceries	Critical	Yes	Yes
Android: Play Store	com.shipt.shopper	Critical	Yes	Yes
iOS: App Store	971888874	Critical	Yes	Yes
iOS: App Store	976353472	Critical	Yes	Yes

These are the vulnerability rewards that are given by the Shipt.

Critical severity bug examples	$3000 - $5000
Remote Code Execution	
Vertical Authentication bypass	
SQL Injection that leaks targeted data	

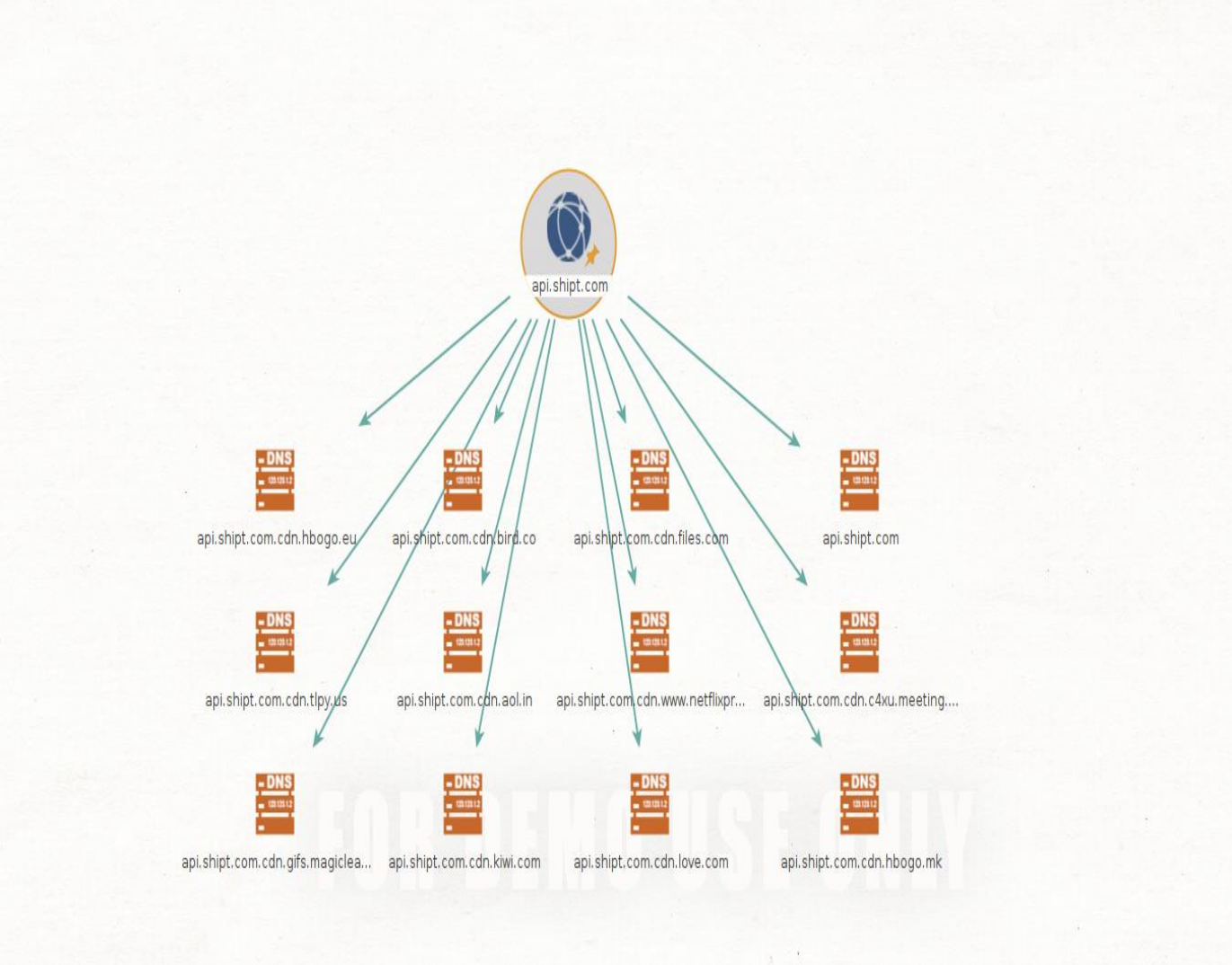
High severity bug examples	$1000 - $3000
Lateral authentication bypass	
Stored XSS	
Insecure handling of authentication cookies	

Medium severity bug examples	$500 - $1000
Reflected XSS	
Insecure Direct Object References	
CSRF on sensitive actions and functions	
Information leaks	
Subdomain Takeover	

Reconnaissance.

I started my Recon phase with the **maltego** tool. This is the most effective tool to help us to think smartly. It has been used by professionals for collecting hidden information as well as, it reduces the time when they do their security assessments.

Initially, I started to gather information by searching DNS names in DNSDB, related to the domain **api.shipt.com** then I found **twelve** DNS.



Then I searched for IP addresses related to each and every DNS. It could be possible to find IP addresses of these domain with the help of the tool. Those are ,

- api.shipt.com.cdn.hbogo.eu - 185.60.68.10 – Hungary
- api.shipt.com.cdn.bird.co - 13.32.153.87
  - 13.32.153.111
  - 13.32.153.126
  - 13.32.153.20

  } united states
- api.shipt.com.cdn.files.com - 34.204.145.250
  - 34.205.137.182
  - 34.204.153.236
  - 34.204.150.23

  } united states

- api.shipt.com - 162.159.135.84
  - 162.159.134.84
- api.shipt.com.tipy.us - could not get IP address of DNS

- api.shipt.com.cdn.aol.in - 74.6.136.150
  - when I check this domain for IPQS, it had a fraud score of 100.

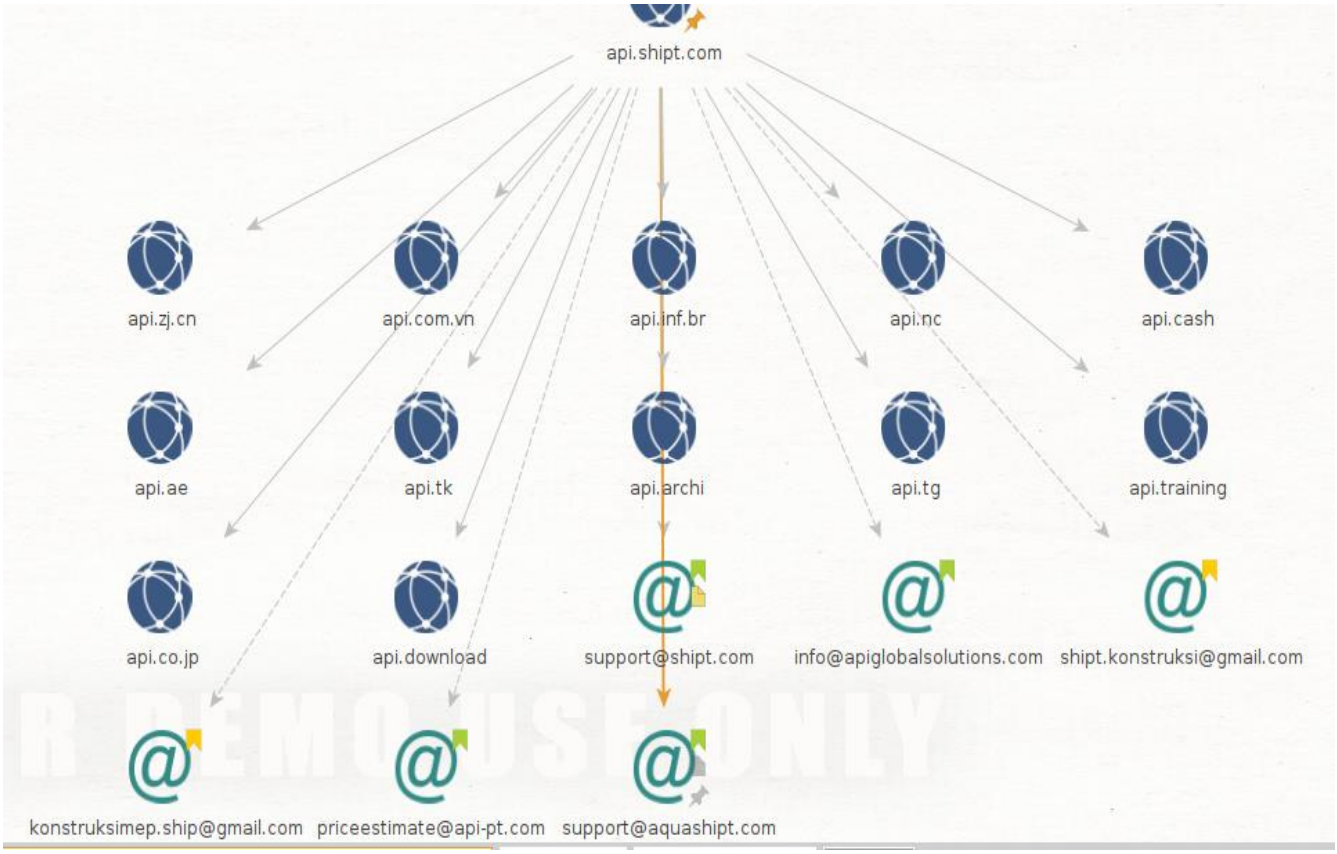
- api.shipt.com.cdn.www.netflixpremium.com - 50.17.247.9
  - 204.236.236.127

  united states
  - 107.20.175.192

- api.shipt.com.cdn.c4xu.meeting.hbogo.rs – 13.81.126.187

- api.shipt.com.cdn.gifs.magicleap.io.bridgeapp.com – 18.217.79.116
  - 18.217.209.204
  - 3.140.47.227

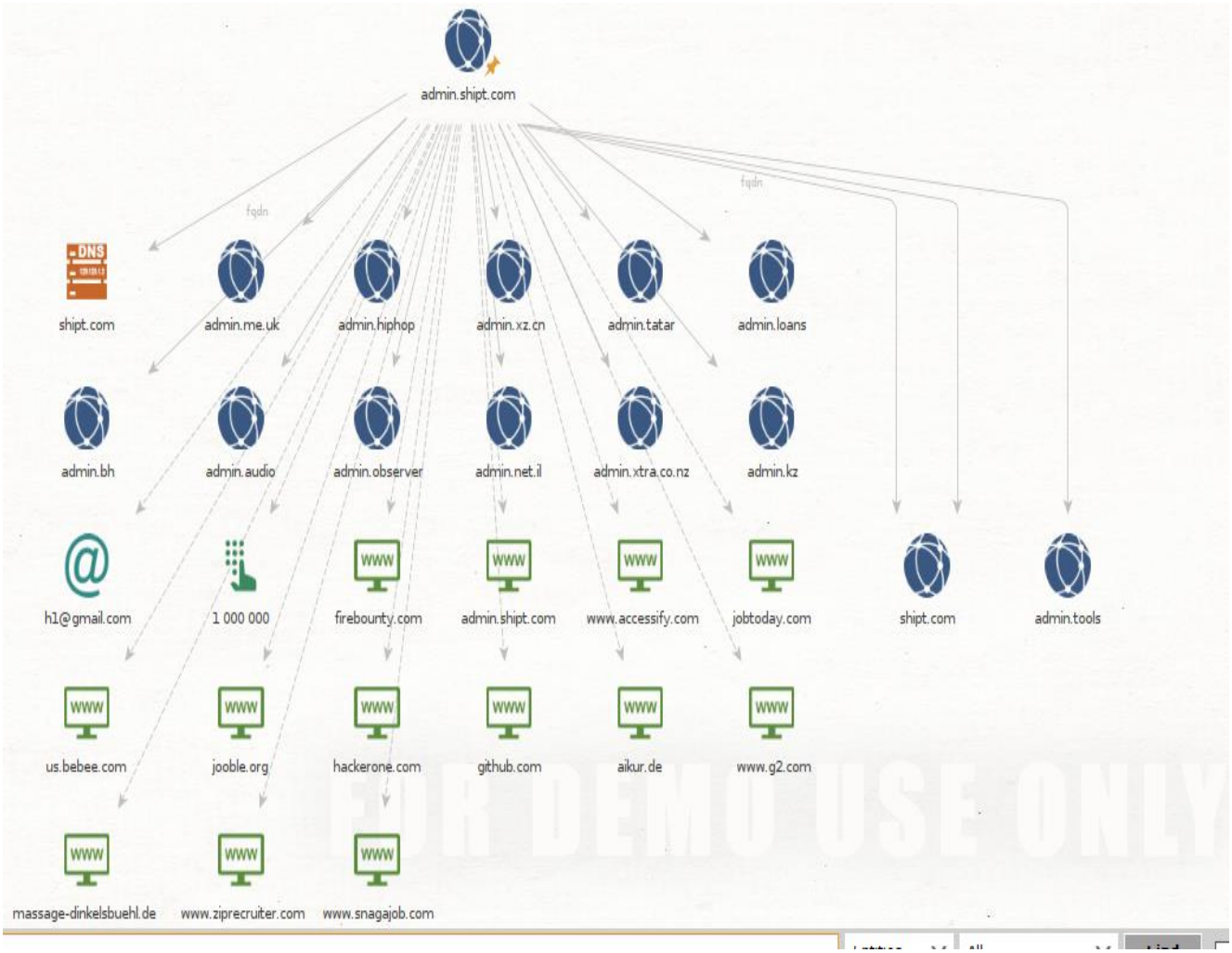
- api.shipt.com.cdn.hbogo.mk – 13.81.126.64
- api.shipt.com.cdn.kiwi.com – could not find the IP address of DNS
- api.shipt.com.cdn.love.com - could not get the IP address of DNS



I searched for the top-level domains related to **api.shipt.com,** I found **twelve** top-level domain that is currently used by that site. Moreover, I found **six** email addresses on that site. I error-checked those email addresses and the result was relatively less than 75 (fraud score should be less than or equal to 75).

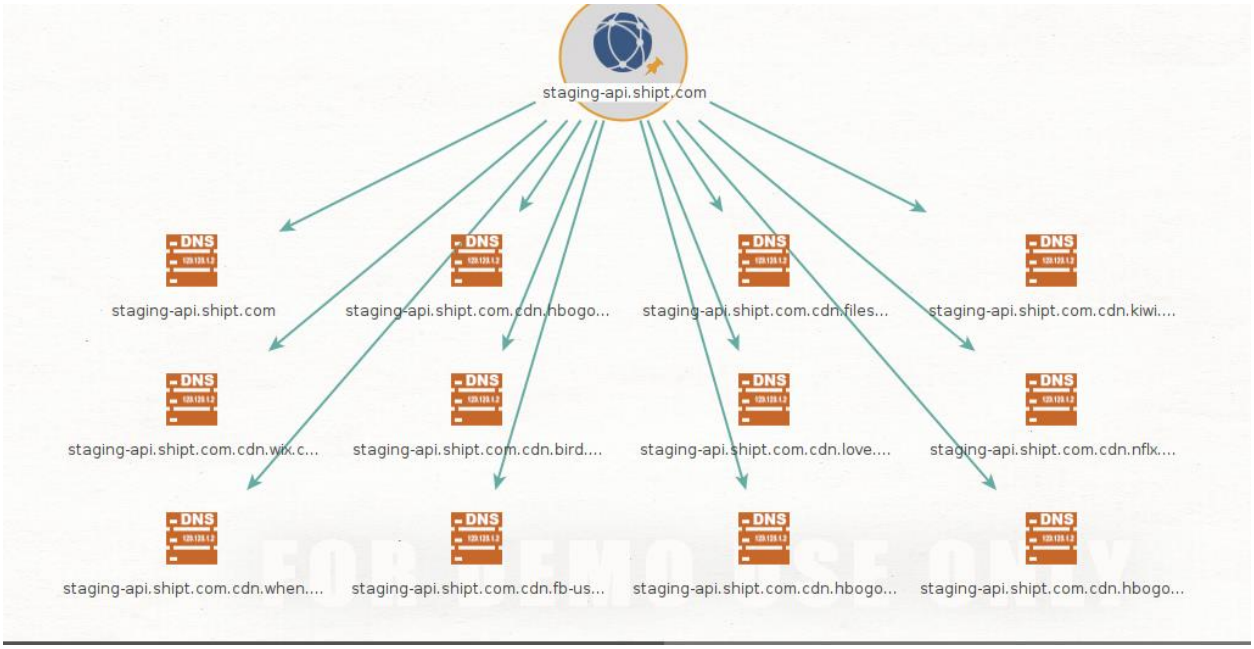
In search, I found some top-level domains, DNS, Email addresses, and websites related to **app.shipt.com**

These are the records related to the **admin.shipt.com**

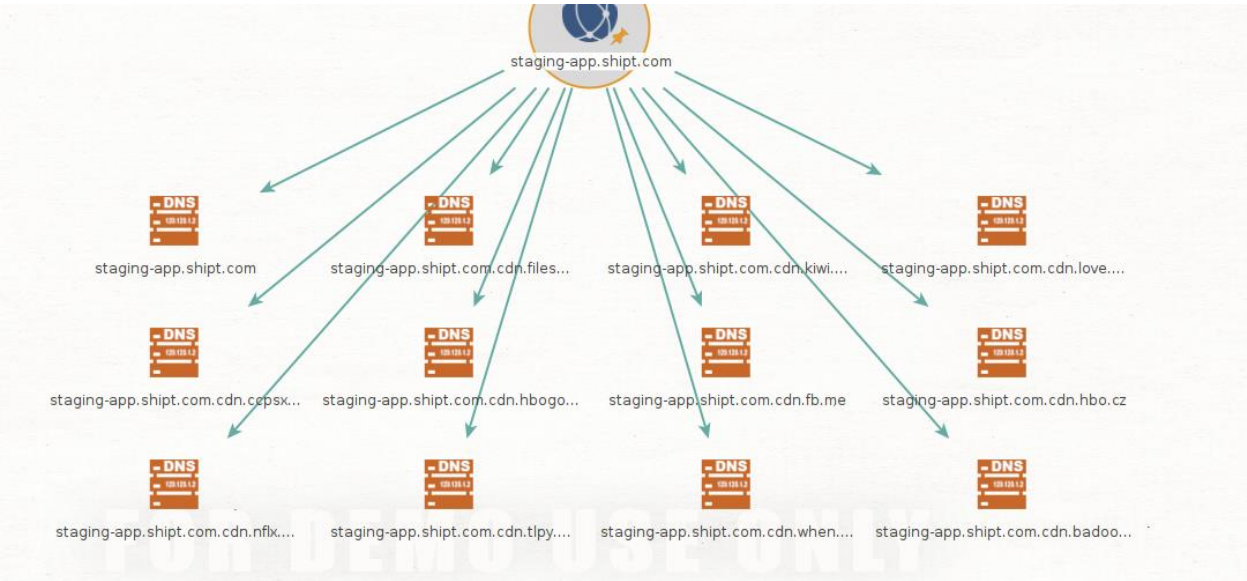


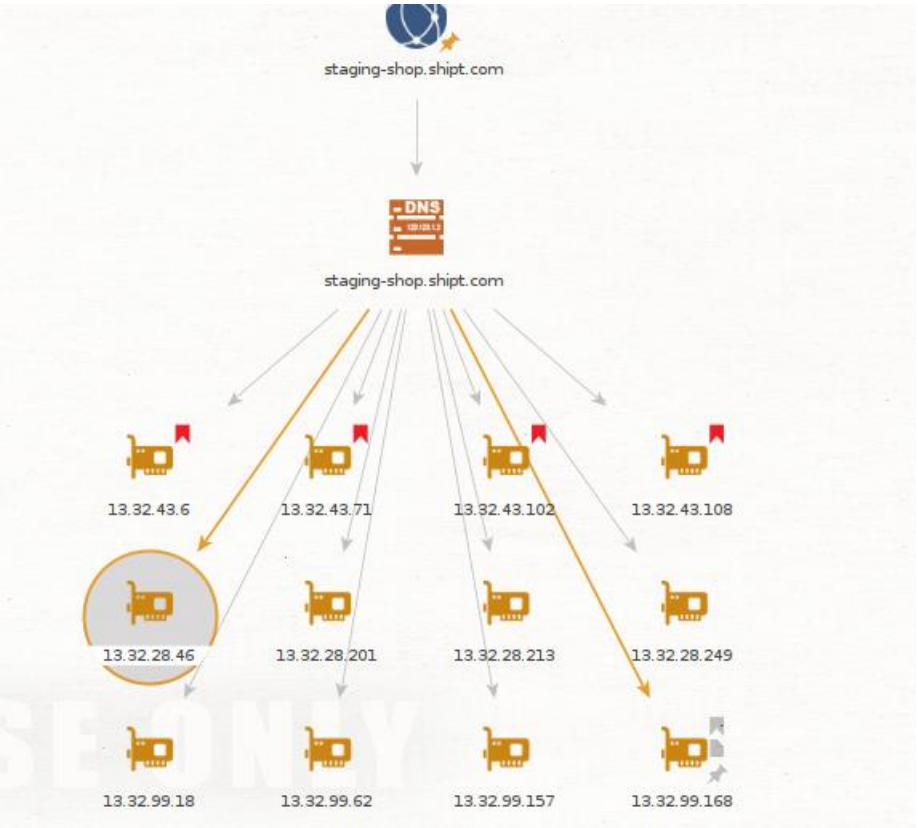
This is the DNS records information of **staging-shoppingcart.shipt.com**

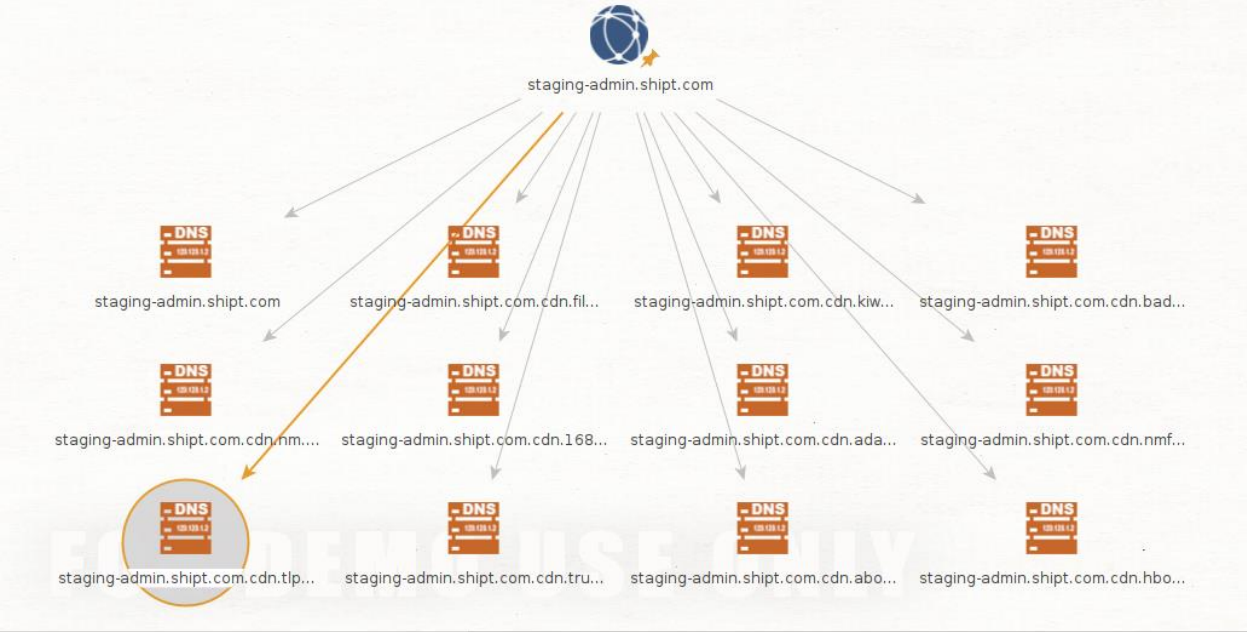
Then I found a DNS record related to **staging-api.shipt.com**

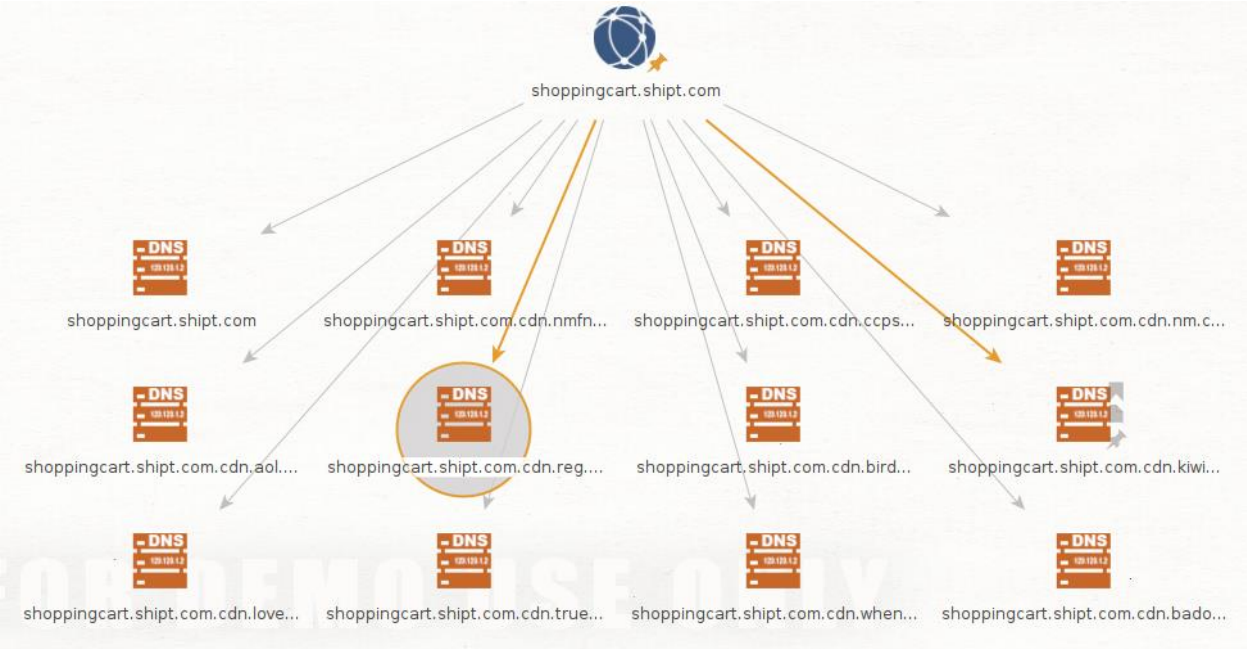


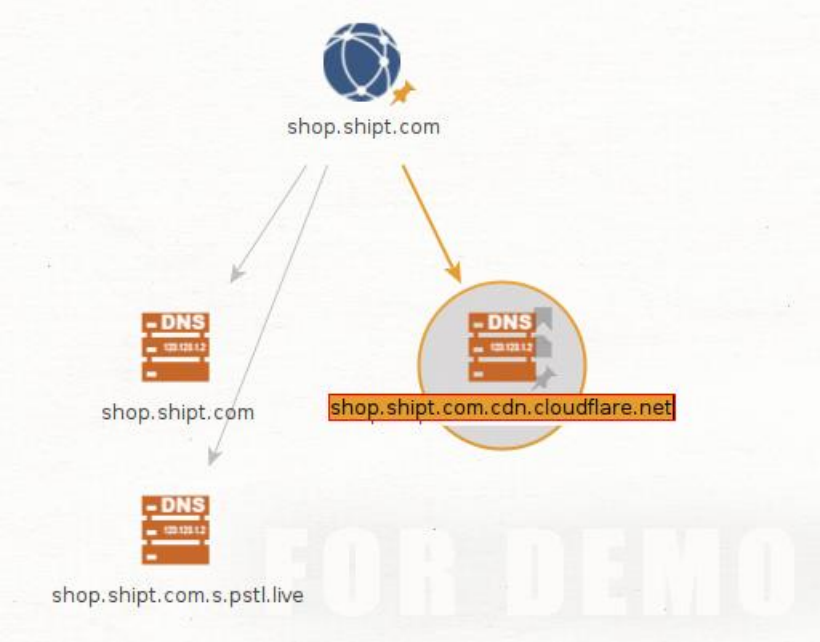
I searched for DNS name-related **staging-app.shipt.com** as well.






staging-admin.shipt.com
staging-admin.shipt.com
staging-admin.shipt.com.cdn.fil...
staging-admin.shipt.com.cdn.kiw...
staging-admin.shipt.com.cdn.bad...
staging-admin.shipt.com.cdn.nm....
staging-admin.shipt.com.cdn.168...
staging-admin.shipt.com.cdn.ada...
staging-admin.shipt.com.cdn.nmf...
staging-admin.shipt.com.cdn.tlp...
staging-admin.shipt.com.cdn.tru...
staging-admin.shipt.com.cdn.abo...
staging-admin.shipt.com.cdn.hbo...

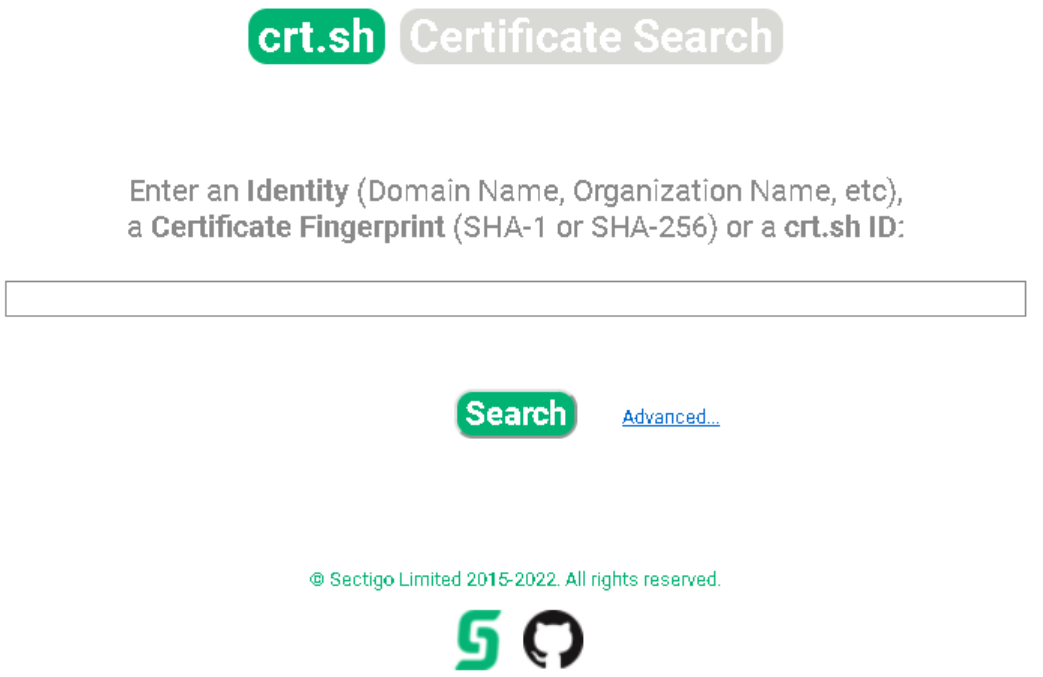

shoppingcart.shipt.com
shoppingcart.shipt.com
shoppingcart.shipt.com.cdn.nmfn...
shoppingcart.shipt.com.cdn.ccps...
shoppingcart.shipt.com.cdn.nm.c...
shoppingcart.shipt.com.cdn.aol....
shoppingcart.shipt.com.cdn.reg....
shoppingcart.shipt.com.cdn.bird...
shoppingcart.shipt.com.cdn.kiwi...
shoppingcart.shipt.com.cdn.love...
shoppingcart.shipt.com.cdn.true...
shoppingcart.shipt.com.cdn.when...
shoppingcart.shipt.com.cdn.bado...



I had to take a ‘ToDoList’ to avoid unwanted loops in the testing process. Overall, maltego is an awesome tool that illustrated organizational structure as a rooted tree structure and it helps us to identify complex structures in one keystroke.

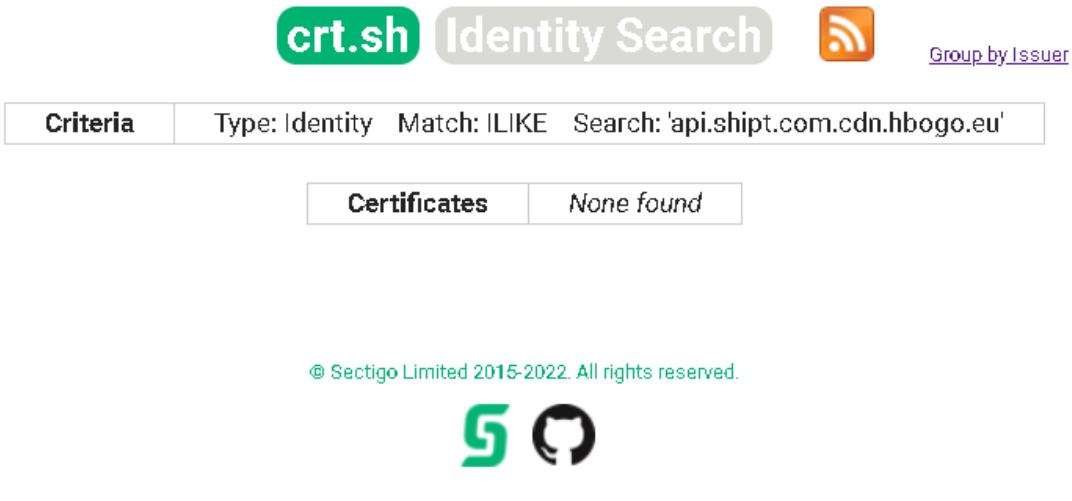
Crt.h

'Crt.h' is a website. That is used to find all TLS or SSL certificates of targeted domains. It is a completely free and open-source site. The idea behind this search is, that I have found that **api.shipt.com** uses **twelve** DNS names (CNAME records). So, we can inspect each one's certificates with the help of this tool whether the DNS name has been registered under the shipt.com; registered period of date is expired or not; verify the SSL is owned by the targeted domain. If there any possibilities in the DNS configuration, we can be able to do a **subdomain takeover**.



Above this image indicates the interface of the **crt.sh** site.

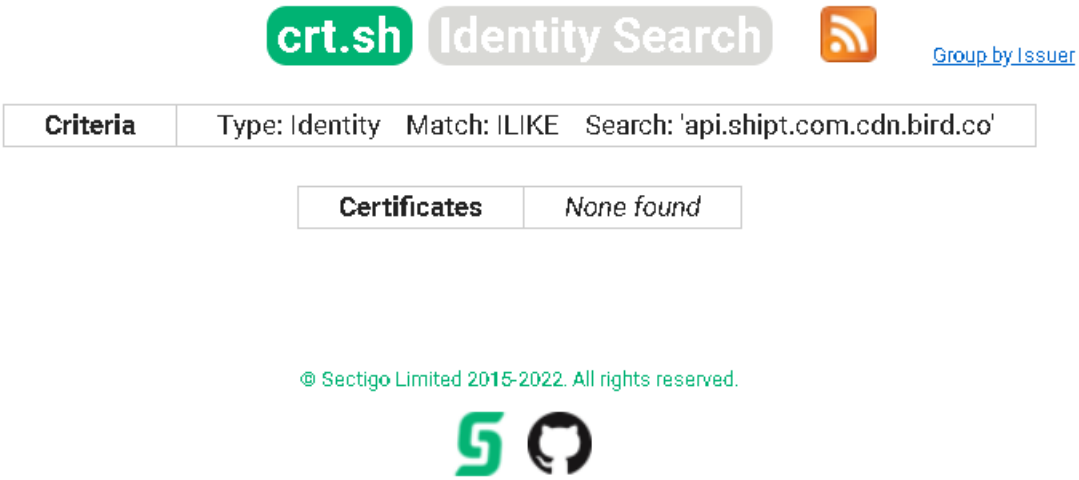
I searched for the CA certificate of this **api.shipt.com.cdn.hbogo.eu** subdomain, But I could not get it.



-

then I checked each one of them accordingly,

-



crt.sh Identity Search Group by Issuer

Criteria	Type: Identity Match: ILIKE Search: 'api.shipt.com.cdn.files.com'

Certificates	None found

-

At last of this process, I was not able to find any of the CA-related api.shipt.com.*DNS names.

For an alternate, I searched CA for DNS names within the property **api.shipt.com.cdn.hbogo.eu** then I found **cdn.hbogo.cu** as a sub-DNS name within that domain. So, I had to decide to find CA for that DNS. Finally, **crt. sh** shows the output like this

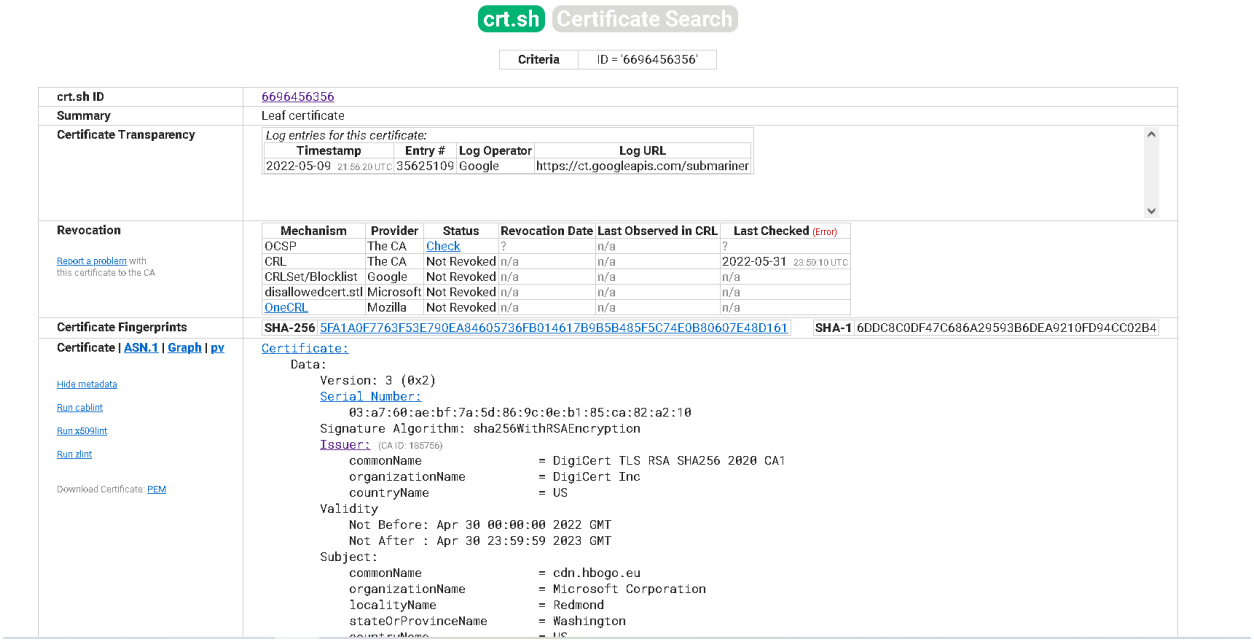
crt.sh Identity Search Group by Issuer

Criteria	Type: Identity Match: ILIKE Search: 'cdn.hbogo.eu'

Certificates

crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	Issuer Name
6696456356	2022-05-09	2022-04-30	2023-04-30	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert TLS RSA SHA256 2020 CA1
6638219077	2022-04-30	2022-04-30	2023-04-30	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert TLS RSA SHA256 2020 CA1
6545622588	2022-04-15	2022-04-15	2023-04-19	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert TLS Hybrid ECC SHA384 2020 CA1
6545617462	2022-04-15	2022-04-15	2023-04-19	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert TLS RSA SHA256 2020 CA1
6213285059	2022-02-20	2022-02-18	2023-03-21	sni11659gl.wpc.edgecastcdn.net	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert TLS RSA SHA256 2020 CA1
6199056201	2022-02-18	2022-02-18	2023-03-21	sni11659gl.wpc.edgecastcdn.net	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert TLS RSA SHA256 2020 CA1
5066273613	2021-08-18	2021-08-02	2022-08-02	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert TLS RSA SHA256 2020 CA1
4973760354	2021-08-02	2021-08-02	2022-08-02	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert TLS RSA SHA256 2020 CA1
4669387687	2021-06-08	2021-06-08	2022-06-13	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert SHA2 Secure Server CA
4667587526	2021-06-08	2021-06-08	2022-06-13	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert SHA2 Secure Server CA
4272524679	2021-03-25	2021-03-18	2022-04-18	sni11659gl.wpc.edgecastcdn.net	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert TLS RSA SHA256 2020 CA1
4234418199	2021-03-18	2021-03-18	2022-04-18	sni11659gl.wpc.edgecastcdn.net	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert TLS RSA SHA256 2020 CA1
3646264499	2020-11-14	2020-11-05	2021-11-04	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert TLS RSA SHA256 2020 CA1
3606398149	2020-11-05	2020-11-05	2021-11-04	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert TLS RSA SHA256 2020 CA1
2777696836	2020-05-07	2020-05-07	2021-08-06	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Secure Site ECC CA-1
2777653074	2020-05-07	2020-05-07	2021-08-06	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert SHA2 Secure Server CA
2451819851	2020-02-12	2020-02-08	2021-02-08	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert SHA2 Secure Server CA
2430523854	2020-02-08	2020-02-08	2021-02-08	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert SHA2 Secure Server CA
1881344790	2019-09-13	2019-05-13	2020-05-13	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert SHA2 Secure Server CA
1881344802	2019-09-13	2019-05-14	2021-05-18	sni11659gl.wpc.edgecastcdn.net	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert SHA2 Secure Server CA
1598560203	2019-06-21	2019-06-21	2020-06-20	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert SHA2 Secure Server CA
1469964964	2019-05-14	2019-05-14	2021-05-18	sni11659gl.wpc.edgecastcdn.net	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert SHA2 Secure Server CA
1467485097	2019-05-13	2019-05-13	2020-05-13	cdn.hbogo.eu	cdn.hbogo.eu	C=US, O=DigiCert Inc, CN=DigiCert SHA2 Secure Server CA

According to its last activity, that domain has not expired.



crt.sh Certificate Search

Criteria | ID = '6696456356'

crt.sh ID	6696456356
Summary	Leaf certificate
Certificate Transparency	*Log entries for this certificate:*

Timestamp	Entry #	Log Operator	Log URL
2022-05-09 21:56:20 UTC	35625109	Google	https://ct.googleapis.com/submariner

Revocation

Report a problem with this certificate to the CA

Mechanism	Provider	Status	Revocation Date	Last Observed in CRL	Last Checked (Error)
OCSP	The CA	Check	?	n/a	?
CRL	The CA	Not Revoked	n/a	n/a	2022-05-31 23:50:10 UTC
CRLSet/Blocklist	Google	Not Revoked	n/a	n/a	n/a
disallowedcert.stl	Microsoft	Not Revoked	n/a	n/a	n/a
OneCRL	Mozilla	Not Revoked	n/a	n/a	n/a

Certificate Fingerprints SHA-256 5FA1A0F7763F53E790EA84605736FB014617B9B5B485F5C74E0B80607E48D161 SHA-1 6DDC8C0DF47C686A29593B6DEA9210FD94CC02B4

Certificate | ASN.1 | Graph | pv

Hide metadata

Run cablint

Run x509lint

Run zlint

Download Certificate: PEM

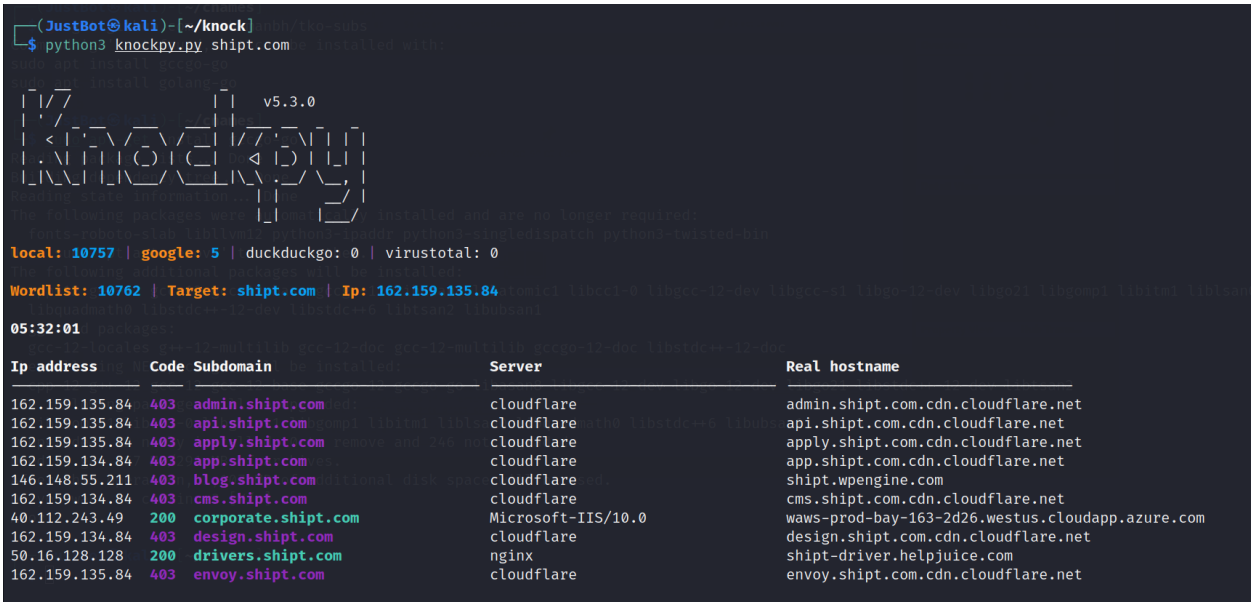
```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            03:a7:60:ae:bf:7a:5d:86:9c:0e:b1:85:ca:82:a2:10
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: (CA ID: 185756)
            commonName               = DigiCert TLS RSA SHA256 2020 CA1
            organizationName         = DigiCert Inc
            countryName              = US
        Validity
            Not Before: Apr 30 00:00:00 2022 GMT
            Not After : Apr 30 23:59:59 2023 GMT
        Subject:
            commonName               = cdn.hbogo.eu
            organizationName         = Microsoft Corporation
            localityName             = Redmond
            stateOrProvinceName      = Washington
            countryName              = US
```

According to the findings, I just started to turn my thoughts to a command-line tool to enumerate the list of subdomains. Because I was frustrated by checking each site in the crt.sh web site. Finally, in search of a tool, I found a subdomain enumeration tool called “**knock”** on GitHub.

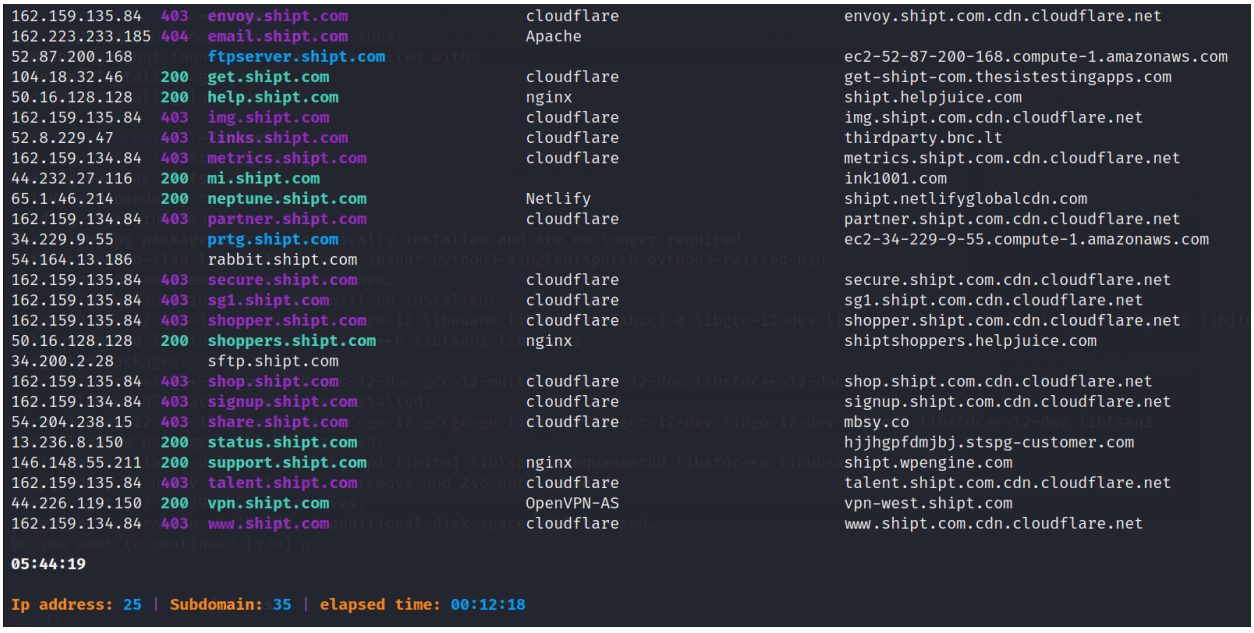
Knock

Knockpy is written in python. It is designed to enumerate subdomains on a target domain through a wordlist. It is intended to monitor for DNS zone transfer and if it is enabled, to attempt to circumvent the wildcard DNS record automatically. So, I enumerated subdomains to www.shipt.com website.

These are the outputs that I got.

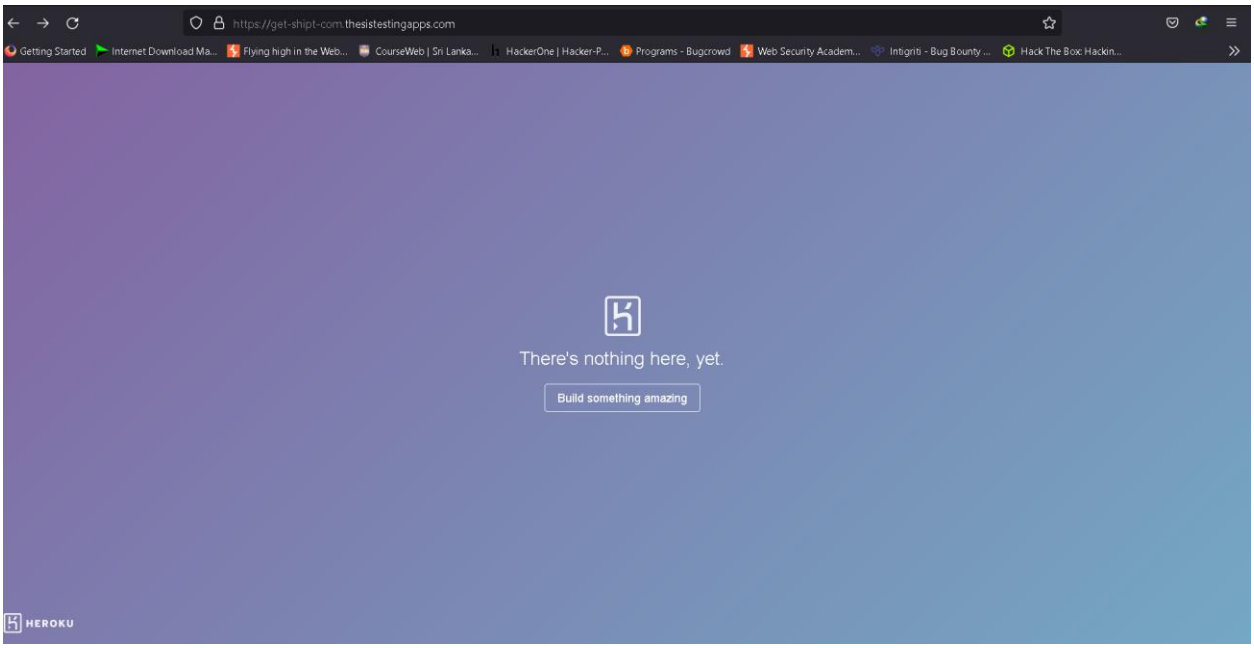


```
┌──(JustBot㉿kali)-[~/knock]
└─$ python3 knockpy.py shipt.com

 _ __
 | |/ /              | |   v5.3.0
 | ' / _ __   ___   ___| | __ _ _   _
 |  < | '_ \ / _ \ / __| |/ /| '_ \| | | |
 | . \| | | | (_) | (__|   < | |_) | |_| |
 |_|\_\_| |_|\___/ \___|_|\_\| .__/ \__, |
                              | |     __/ |
                              |_|    |___/

local: 10757 | google: 5 | duckduckgo: 0 | virustotal: 0

Wordlist: 10762 | Target: shipt.com | Ip: 162.159.135.84

05:32:01

Ip address      Code  Subdomain                 Server                  Real hostname
--------------  ----  ------------------------  ----------------------  ------------------------------------------
162.159.135.84  403   admin.shipt.com           cloudflare              admin.shipt.com.cdn.cloudflare.net
162.159.135.84  403   api.shipt.com             cloudflare              api.shipt.com.cdn.cloudflare.net
162.159.135.84  403   apply.shipt.com           cloudflare              apply.shipt.com.cdn.cloudflare.net
162.159.134.84  403   app.shipt.com             cloudflare              app.shipt.com.cdn.cloudflare.net
146.148.55.211  403   blog.shipt.com            cloudflare              shipt.wpengine.com
162.159.134.84  403   cms.shipt.com             cloudflare              cms.shipt.com.cdn.cloudflare.net
40.112.243.49   200   corporate.shipt.com       Microsoft-IIS/10.0      waws-prod-bay-163-2d26.westus.cloudapp.azure.com
162.159.134.84  403   design.shipt.com          cloudflare              design.shipt.com.cdn.cloudflare.net
50.16.128.128   200   drivers.shipt.com         nginx                   shipt-driver.helpjuice.com
162.159.135.84  403   envoy.shipt.com           cloudflare              envoy.shipt.com.cdn.cloudflare.net
```



```
162.159.135.84  403   envoy.shipt.com           cloudflare              envoy.shipt.com.cdn.cloudflare.net
162.223.233.185 404   email.shipt.com           Apache
52.87.200.168         ftpserver.shipt.com                               ec2-52-87-200-168.compute-1.amazonaws.com
104.18.32.46    200   get.shipt.com             cloudflare              get-shipt-com.thesistestingapps.com
50.16.128.128   200   help.shipt.com            nginx                   shipt.helpjuice.com
162.159.135.84  403   img.shipt.com             cloudflare              img.shipt.com.cdn.cloudflare.net
52.8.229.47     403   links.shipt.com           cloudflare              thirdparty.bnc.lt
162.159.134.84  403   metrics.shipt.com         cloudflare              metrics.shipt.com.cdn.cloudflare.net
44.232.27.116   200   mi.shipt.com                                      ink1001.com
65.1.46.214     200   neptune.shipt.com         Netlify                 shipt.netlifyglobalcdn.com
162.159.134.84  403   partner.shipt.com         cloudflare              partner.shipt.com.cdn.cloudflare.net
34.229.9.55           prtg.shipt.com                                    ec2-34-229-9-55.compute-1.amazonaws.com
54.164.13.186         rabbit.shipt.com
162.159.135.84  403   secure.shipt.com          cloudflare              secure.shipt.com.cdn.cloudflare.net
162.159.135.84  403   sg1.shipt.com             cloudflare              sg1.shipt.com.cdn.cloudflare.net
162.159.135.84  403   shopper.shipt.com         cloudflare              shopper.shipt.com.cdn.cloudflare.net
50.16.128.128   200   shoppers.shipt.com        nginx                   shiptshoppers.helpjuice.com
34.200.2.28           sftp.shipt.com
162.159.135.84  403   shop.shipt.com            cloudflare              shop.shipt.com.cdn.cloudflare.net
162.159.134.84  403   signup.shipt.com          cloudflare              signup.shipt.com.cdn.cloudflare.net
54.204.238.15   403   share.shipt.com           cloudflare              mbsy.co
13.236.8.150    200   status.shipt.com                                  hjjhgpfdmjbj.stspg-customer.com
146.148.55.211  200   support.shipt.com         nginx                   shipt.wpengine.com
162.159.135.84  403   talent.shipt.com          cloudflare              talent.shipt.com.cdn.cloudflare.net
44.226.119.150  200   vpn.shipt.com             OpenVPN-AS              vpn-west.shipt.com
162.159.134.84  403   www.shipt.com             cloudflare              www.shipt.com.cdn.cloudflare.net

05:44:19

Ip address: 25 | Subdomain: 35 | elapsed time: 00:12:18
```

Then check each subdomain on the internet which has a status code of 200. Then I search for get.shipt.com it redirected me to a website. So on, it had a DNS entry pointing to **get-shipt-com.thesistestingapps.com** but when I checked nothing was hosting on that page. When I redirected it shows like this,



I assumed that something not going well on that website. So decided to check this CA on **crt.sh** website.

crt.sh Certificate Search

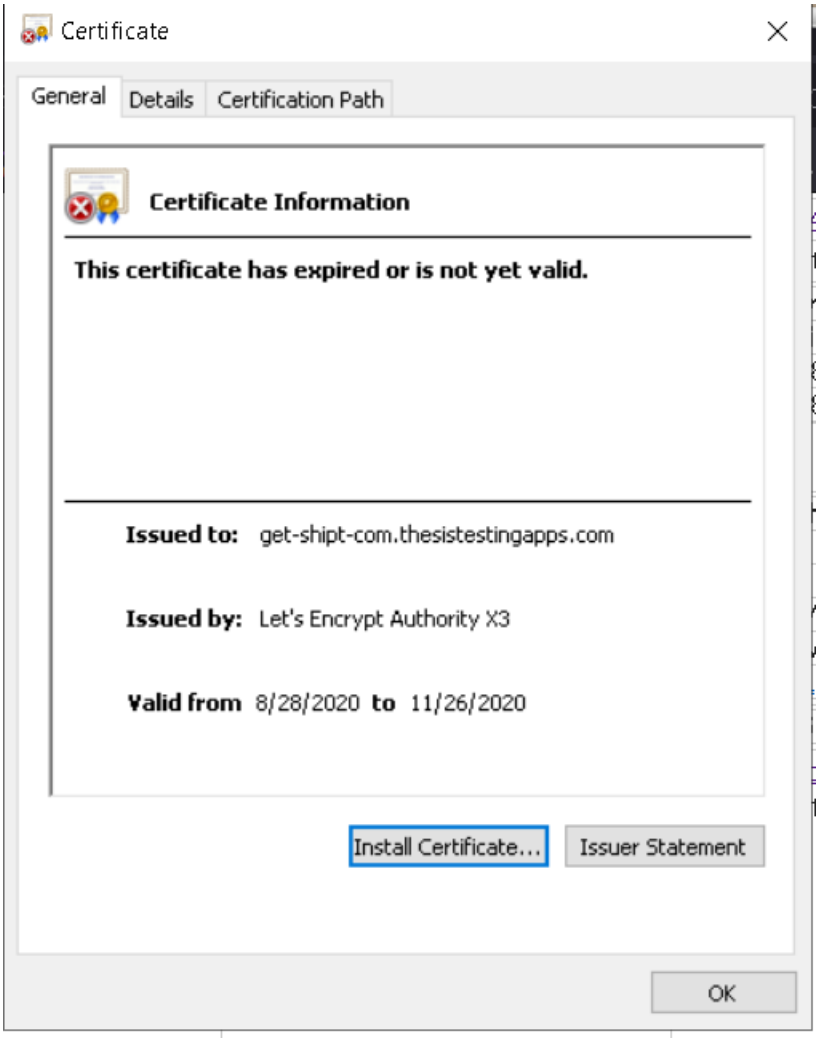
Criteria	ID = '3300014662'

crt.sh ID	3300014662
Summary	Leaf certificate
Certificate Transparency	*Log entries for this certificate:* Timestamp | Entry # | Log Operator | Log URL 2020-08-28 02:15:43 UTC | 859395812 | Google | https://ct.googleapis.com/logs/argon2020 2020-08-28 02:15:43 UTC | 890319429 | Google | https://ct.googleapis.com/logs/xenon2020
Revocation Report a problem with this certificate to the CA	Mechanism | Provider | Status | Revocation Date | Last Observed in CRL | Last Checked (Error) OCSP | The CA | Check | ? | n/a | ? CRL | The CA | Unknown (Expired) | n/a | n/a | CRLSet/Blocklist | Google | Not Revoked | n/a | n/a | n/a disallowedcert.stl | Microsoft | Not Revoked | n/a | n/a | n/a OneCRL | Mozilla | Not Revoked | n/a | n/a | n/a
Certificate Fingerprints	**SHA-256** D753D188A94C43BAA028BF7668EF4CC42515F0DA651880A80BC585A2E148C6A4 **SHA-1** 72336C34C4A0F98493B493852076F42C907EEBFC
Certificate | ASN.1 | Graph | pv Hide metadata Run cablint Run x509lint Run zlint Download Certificate: PEM	Certificate: Data: Version: 3 (0x2) Serial Number: 03:b0:33:81:7d:5f:1d:0d:f2:53:9a:24:f4:c1:cb:ca:cc:85 Signature Algorithm: sha256WithRSAEncryption Issuer: (CA ID: 16418) commonName = Let's Encrypt Authority X3 organizationName = Let's Encrypt countryName = US Validity (Expired) Not Before: Aug 28 01:15:43 2020 GMT Not After : Nov 26 01:15:43 2020 GMT Subject: commonName = get-shipt-com.thesistestingapps.com Subject Public Key Info:

crt.sh CA Search

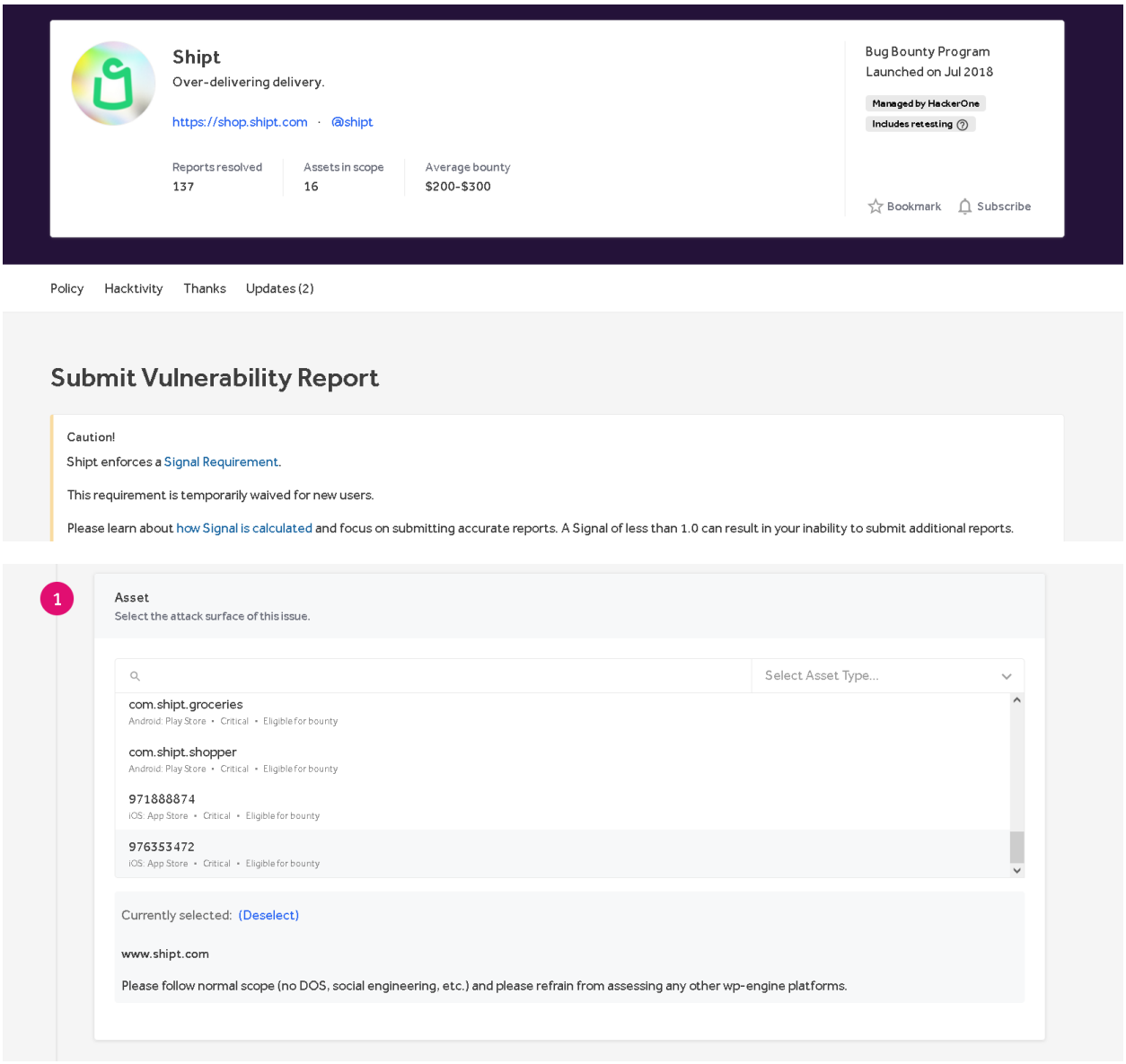
Criteria	Type: CA ID Match: = Search: '16418'

crt.sh CA ID	16418
CA Name/Key	Subject: commonName = Let's Encrypt Authority X3 organizationName = Let's Encrypt countryName = US Subject Public Key Info: Public Key Algorithm: rsaEncryption RSA Public-Key: (2048 bit) Modulus: 00:9c:d3:0c:f0:5a:e5:2e:47:b7:72:5d:37:83:b3: 68:63:30:ea:d7:35:26:19:25:e1:bd:be:35:f1:70: 92:2f:b7:b8:4b:41:05:ab:a9:9e:35:08:58:ec:b1: 2a:c4:68:87:0b:a3:e3:75:e4:e6:f3:a7:62:71:ba: 79:81:60:1f:d7:91:9a:9f:f3:d0:78:67:71:c8:69: 0e:95:91:cf:fe:e6:99:e9:60:3c:48:cc:7e:ca:4d: 77:12:24:9d:47:1b:5a:eb:b9:ec:1e:37:00:1c:9c: ac:7b:a7:05:ea:ce:4a:eb:bd:41:e5:36:98:b9:cb: fd:6d:3c:96:68:df:23:2a:42:90:0c:86:74:67:c8: 7f:a5:9a:b8:52:61:14:13:3f:65:e9:82:87:cb:db: fa:0e:56:f6:86:89:f3:85:3f:97:86:af:b0:dc:1a: ef:6b:0d:95:16:7d:c4:2b:a0:65:b2:99:04:36:75: 80:6b:ac:4a:f3:1b:90:49:78:2f:a2:96:4f:2a:20: 25:29:04:c6:74:c0:d0:31:cd:8f:31:38:95:16:ba: a8:33:b8:43:f1:b1:1f:c3:30:7f:a2:79:31:13:3d: 2d:36:f8:e3:fc:f2:33:6a:b9:39:31:c5:af:c4:8d: 0d:1d:64:16:33:aa:fa:84:29:b6:d4:0b:c0:d8:7d: c3:93 Exponent: 65537 (0x10001)
Certificates	crt.sh ID | Not Before | Not After | Issuer Name 47997543 | 2016-10-06 | 2021-10-06 | C=US, O=Internet Security Research Group, CN=ISRG Root X1 15706126 | 2016-03-17 | 2021-03-17 | O=Digital Signature Trust Co., CN=DST Root CA X3



So I double-checked it CA. finally, I found something interesting which is the CA certificate has Expired. This means any person who intended to purchase that domain can purchase and host a service or website. This leads to a security threat to a company. Therefore, I decided to submit a report to HackerOne.

These are the steps it showed me to enter in that submit report section.



2

Weakness

Select the type of the potential issue you have discovered. Can't pick just one? Select the best match or submit a separate report for each distinct weakness.

Subdomain Takeover | All clusters

Currently selected: None

3

Severity (optional)

Estimate the severity of this issue.

None | Low | Medium | High | Critical — Low

OR

CVSS v3.0 Calculator — No Rating ---

Attack Vector: Network | Adjacent | Local | Physical

Attack Complexity: Low | High

Privileges Required: None | Low | High

User Interaction: None | Required

Scope: Unchanged | Changed

Confidentiality: None | Low | High

Integrity: None | Low | High

Availability: None | Low | High

4

Proof of Concept

The proof of concept is the most important part of your report submission. Clear, reproducible steps will help us validate this issue as quickly as possible.

Title *

A clear and concise title includes the type of vulnerability and the impacted asset.

Subdomain Takeover at get.shipt.com 115

Description *

What is the vulnerability? In clear steps, how do you reproduce it?

initially, i was able to enumerate a list of subdomains of 'www.shipt.com' withe help of the tool called 'knockpy'. After then, i redirected to every subdomains which has status code as 200. So on, i redirected to a subdomain 'get.shipt.com'. In its record, it has a DNS entry pointing to 'get-shipt-com.thesistestingapps.com ' but there is nothing existing on that site. Moreover, i also checked CA for get.ship.com in the crt.sh site. there were CA record pointing to 'get-shipt-com.thesistestingapps.com' when i checked its CA, it was paired.

Write | Preview — Parsed with Markdown

Impact *

What security impact could an attacker achieve?

A malicious person can claim get-shipt-com.thesistestingapps.com and host a site there.

Write | Preview | Parsed with Markdown

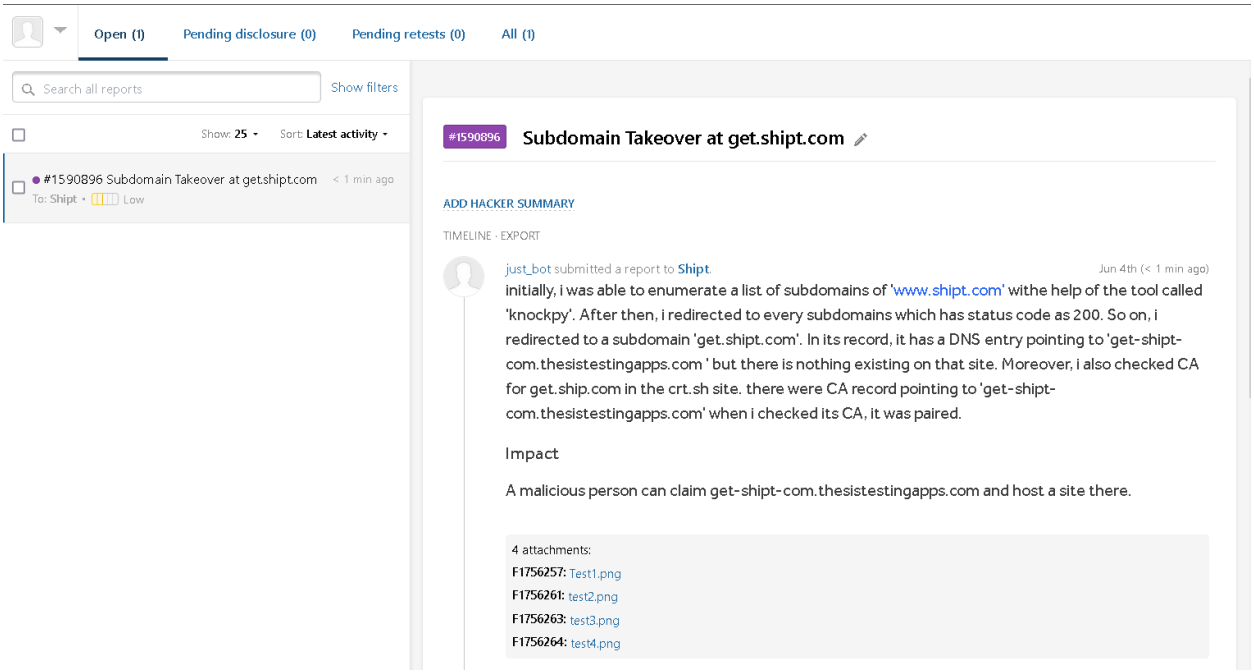
F1756257: Test1.png 42.8KB Download

F1756261: test2.png 0.1MB Download

F1756263: test3.png 44.7KB Download

F1756264: test4.png 43.1KB Download

I also attached screenshots of my finding.

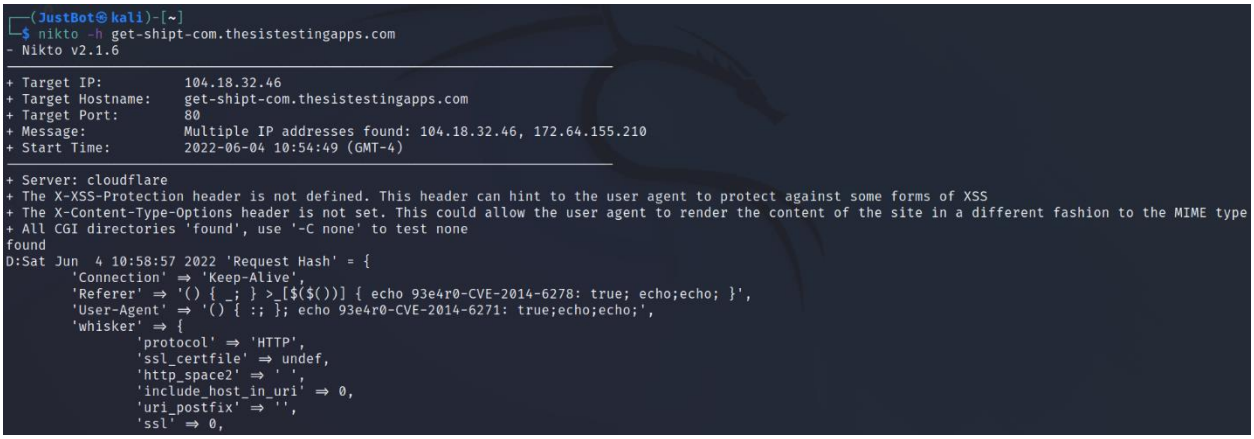


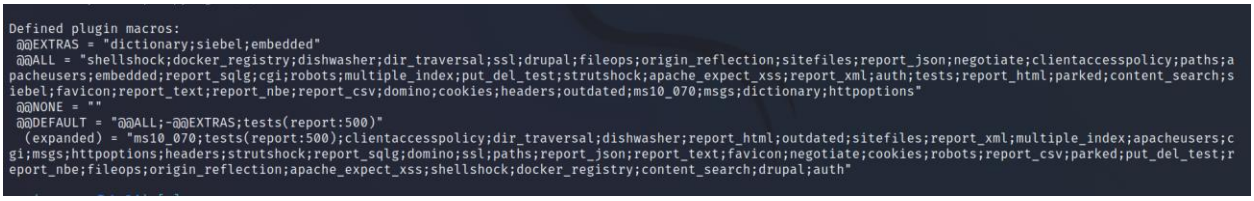
Then, I started to check different kinds of vulnerabilities on that site to identify whether the web application is vulnerable or not.

Nikto

Nikto is a web server and web application inspector that is free and open source. Nikto can test web servers for a wide range of security concerns, including around 6700 potentially harmful files and programs. Nikto may also check for out-of-date web server software and also version-specific problems.

So, I started to gather information from **get-shipt-com.thesistestingapps.com**



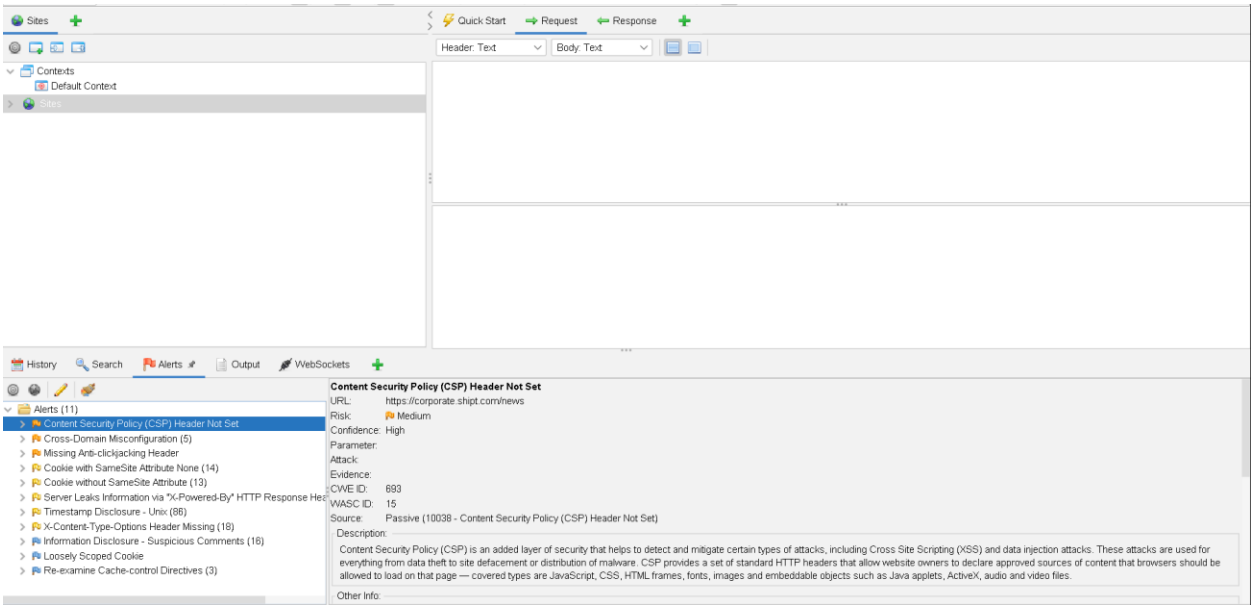


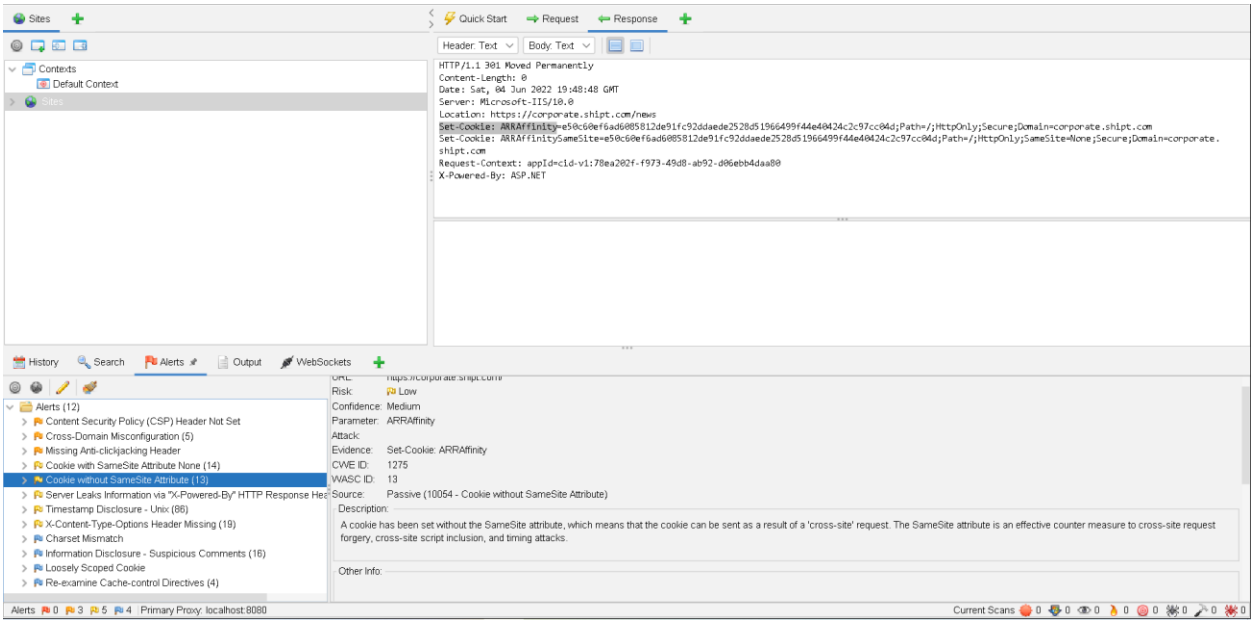
This is a plugin detail that is provided by the Nikto. In common, when checking for all subdomains it showed the same output as this.

OWASP ZAP

OWASAP Zed Attack Proxy is an open-source web application security scanner. With help of this tool, we can be able to identify web application issues without struggle. In pen-testing, this tool reduces the time. So, I started to check all websites with these tools.

When I check one of the sub-domains of **shipt.com.** it showed below output

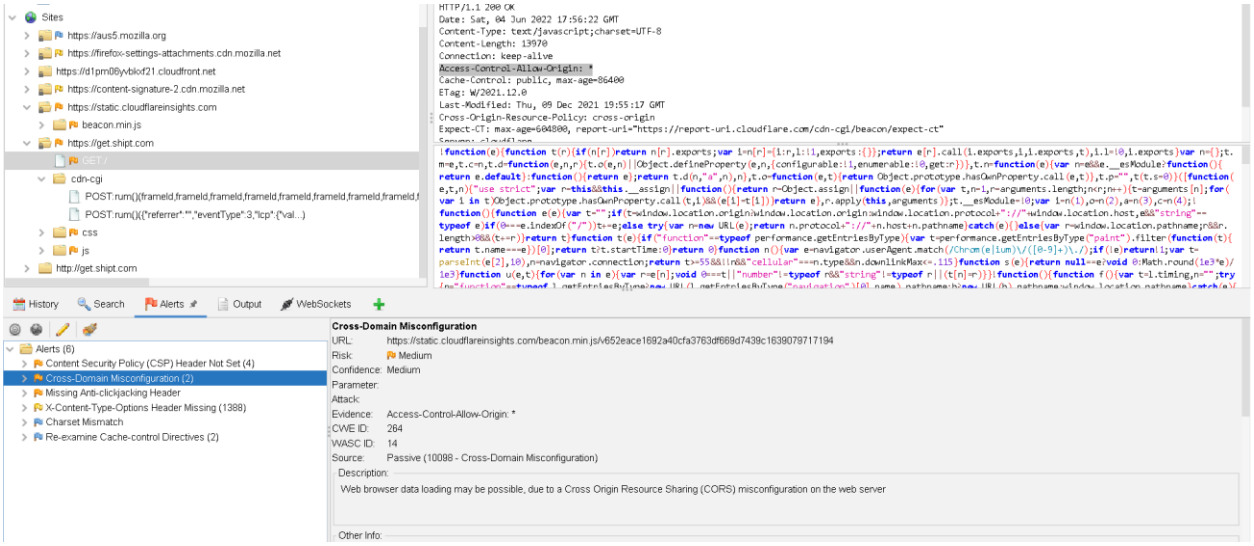


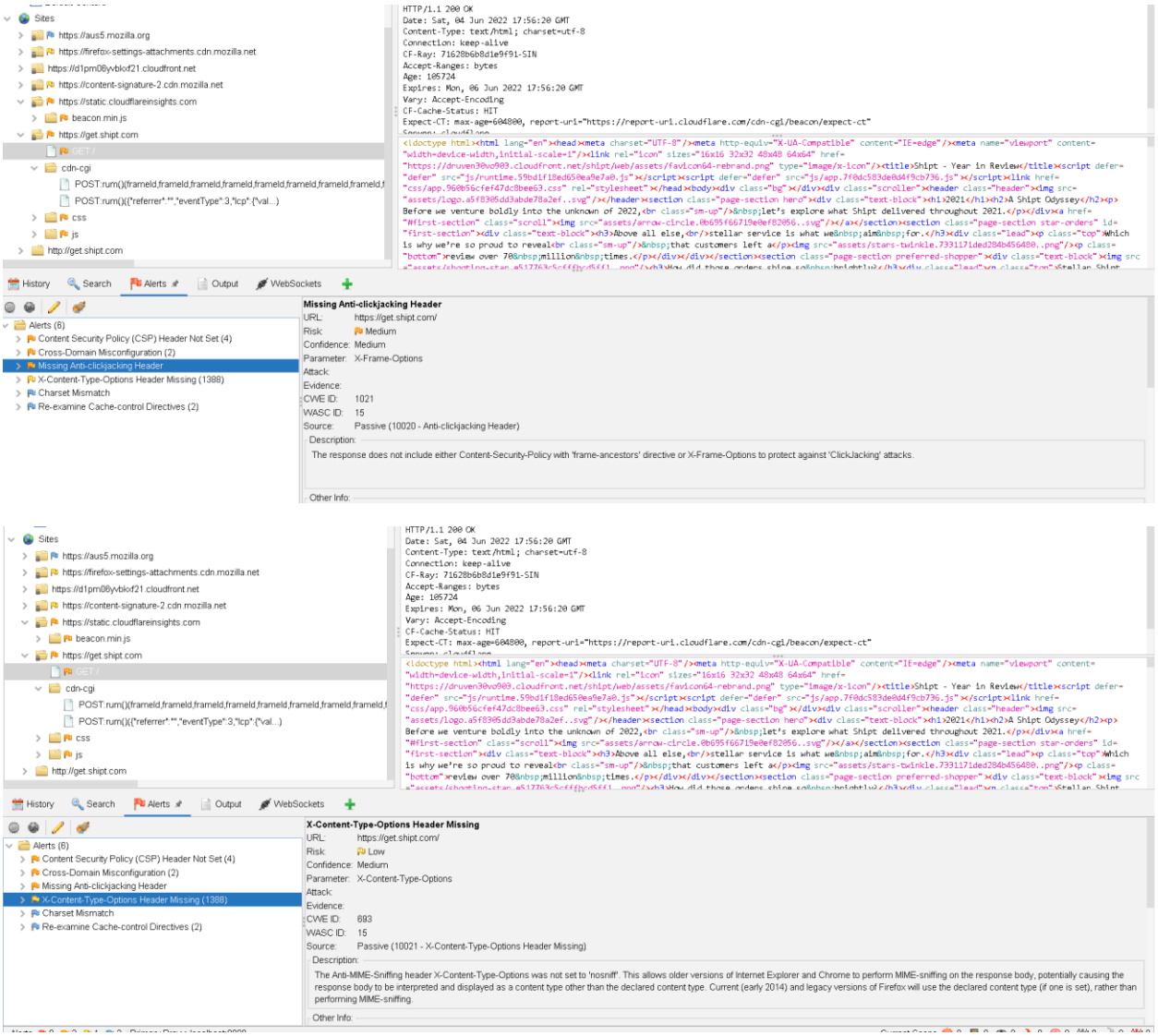


When I checked there were **eight** issues on that site. In those, CSP header not set, Cross-Domain misconfiguration, Missing anti-clickjacking header missing was

medium issues. So, an attacker can execute XSS, CSRF, and Clickjacking attacks on this site.

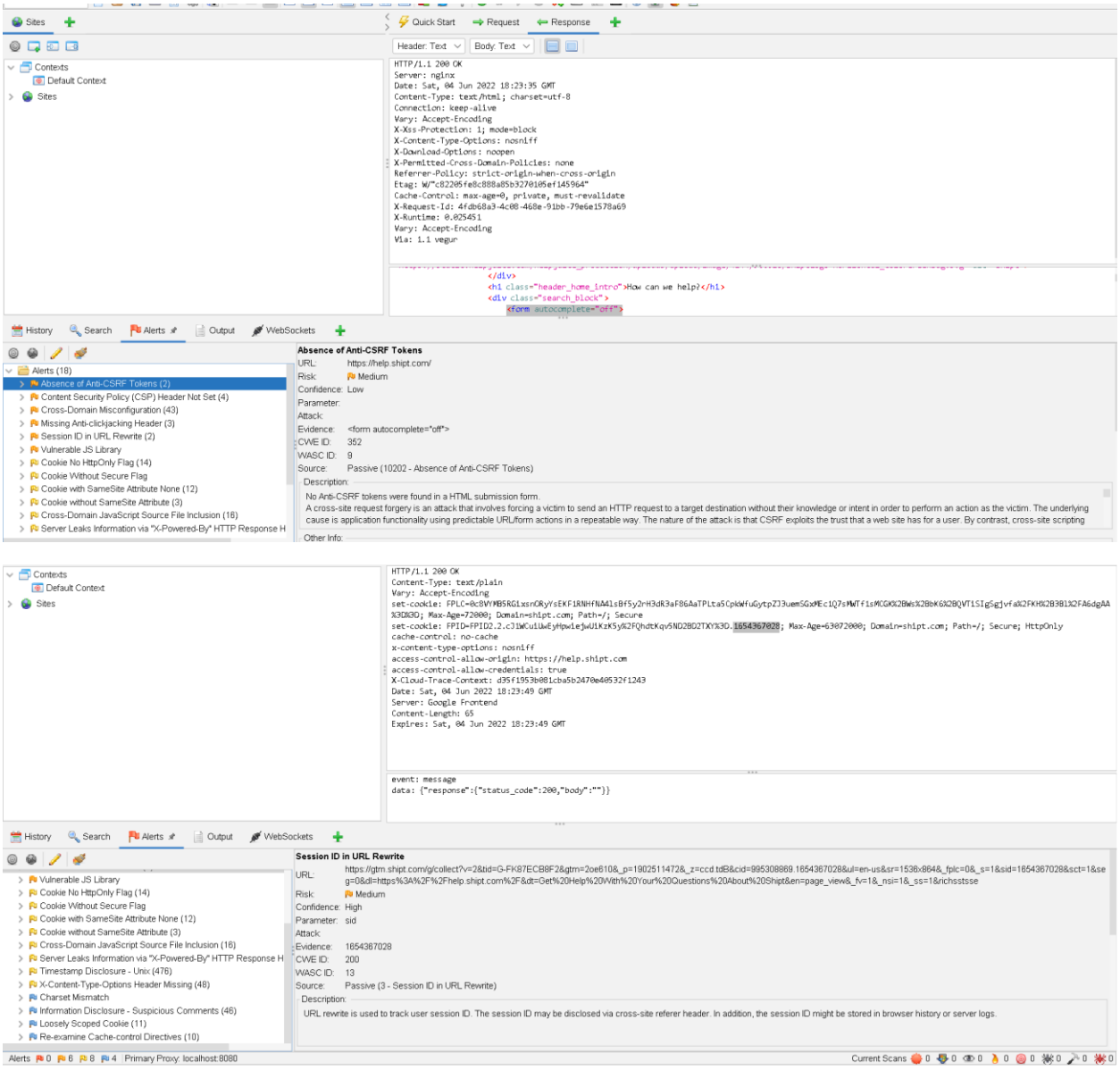
Then, I checked **get.shipt.com** ZAP also showed the same issues as before.





As an assumption, we can say that because of these missing headers, an attacker can able to perform XSS, CSRF, and, Clickjacking attacks on this site.

So on, I checked the **support.shipt.com**

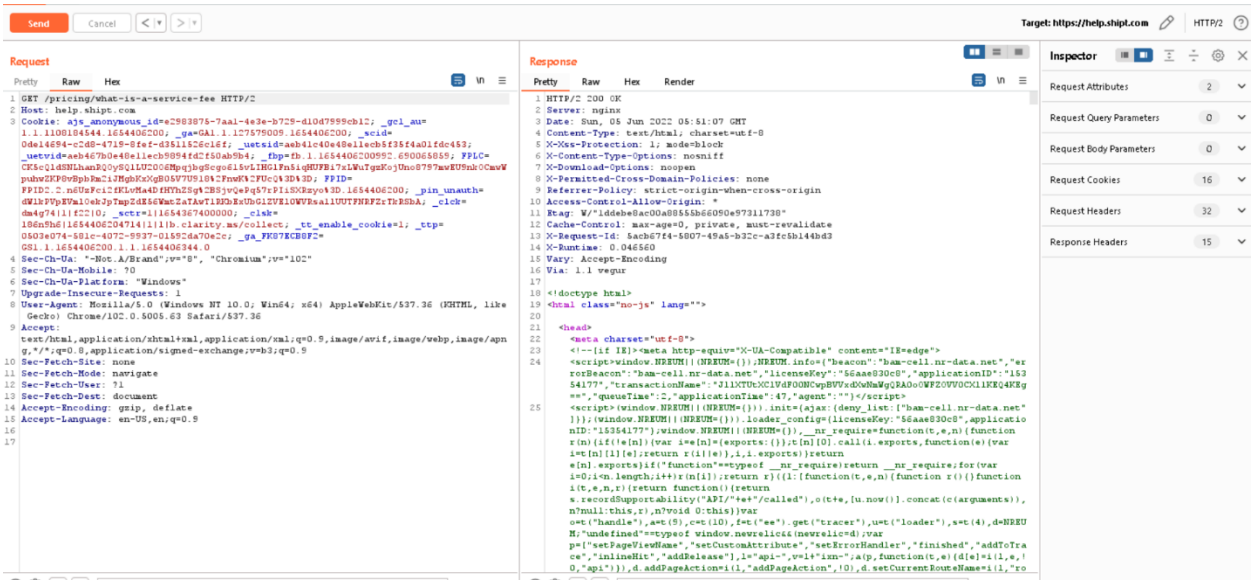


There were **six** medium issues found by the zap. The server leaks the information via the ‘X-Powered-By’ header and the absence of anti-CSRF tokens most considerable issue on this site.

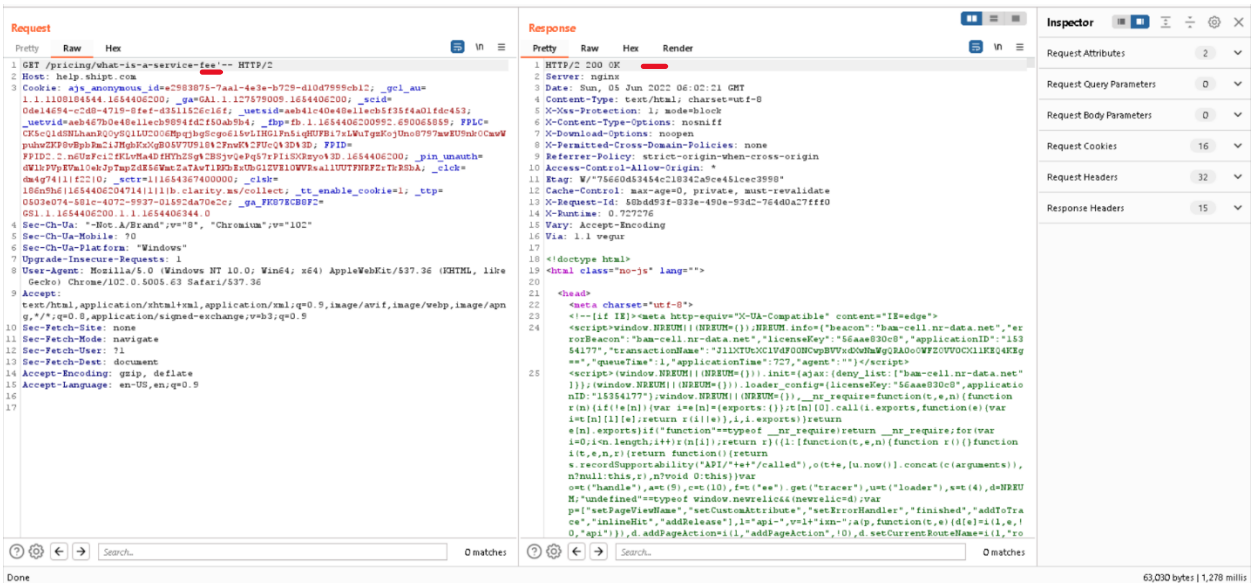
Burp Suite

This is a well-known web application pen-testing tool that is used by professionals all around the world. This is come-up with two editions. There are enterprise edition and community edition. The community edition is free.

When I intercepted the traffic of **help.shipt.com/pricing,** these are the output I got a response.

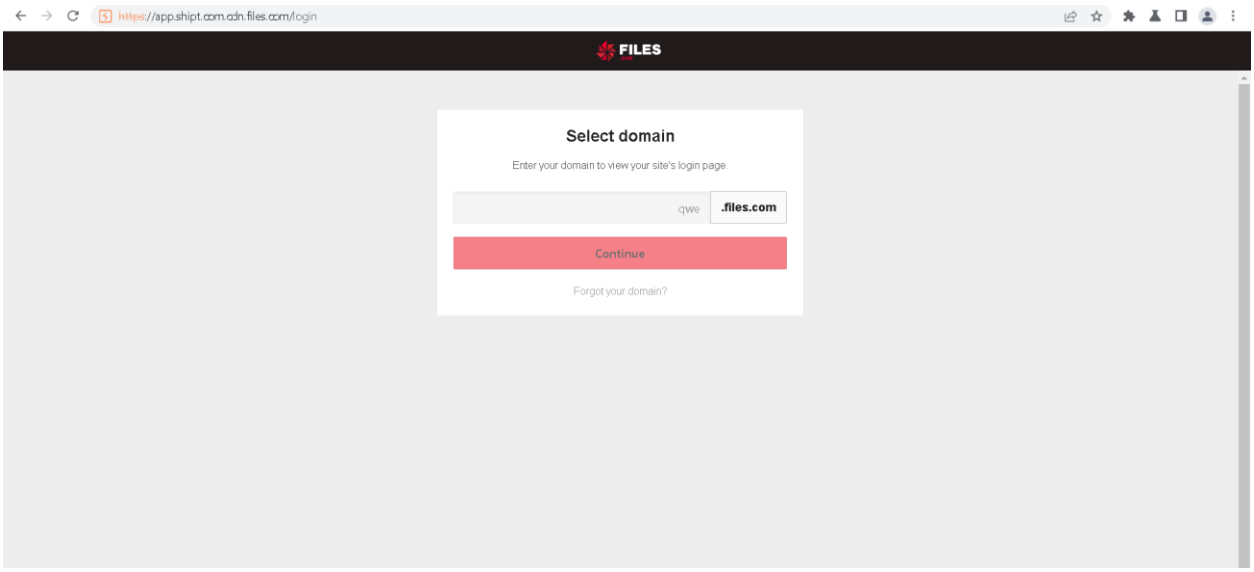


So I decided to inject a basic SQL injection into this website to understand whether the website is a response or not to a SQL injection.

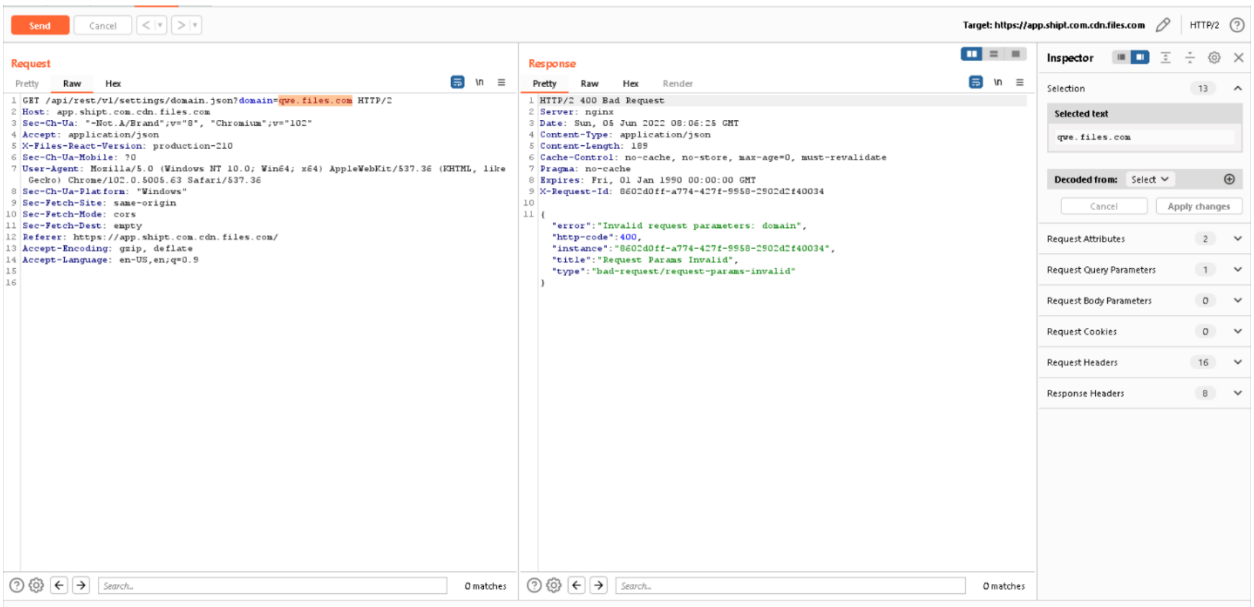


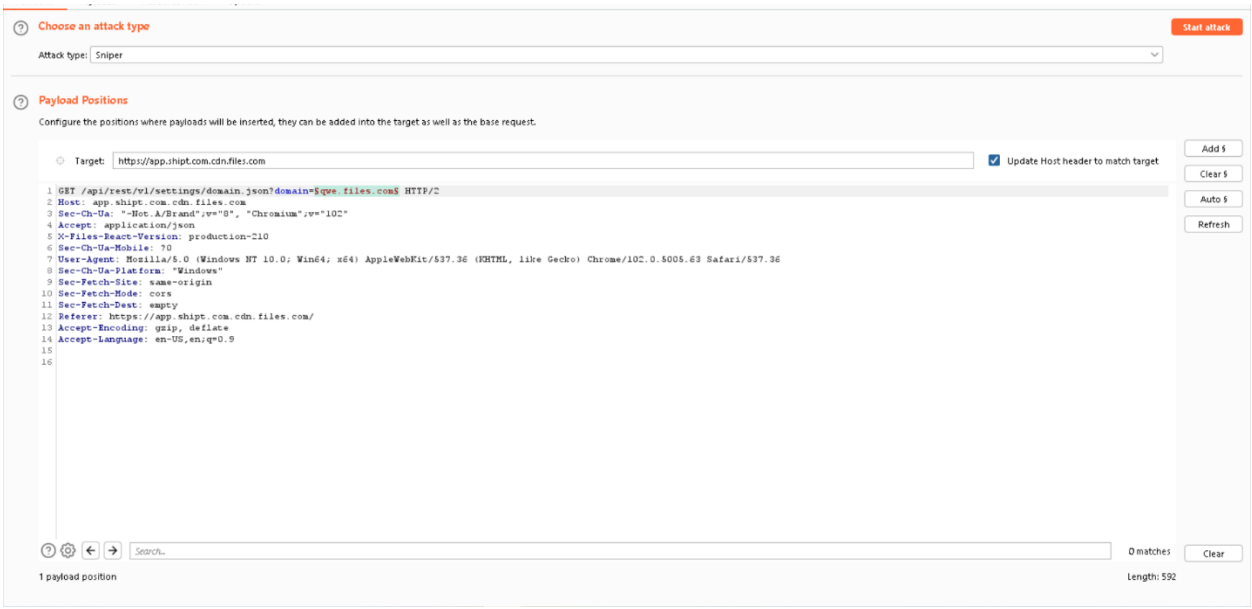
After SQL injection the response header responded with status code 200. From this instance, we can assume this site is vulnerable to SQL injection.

When I searched for, the vulnerable getting user input function which is not showing user input without encoding it in the URL. So on, I found **api.shipt.com** has a 3rd party service pointing to **api.shipt.com.cdn.files.com.** So, re-directed to that site. This showed to me like this,

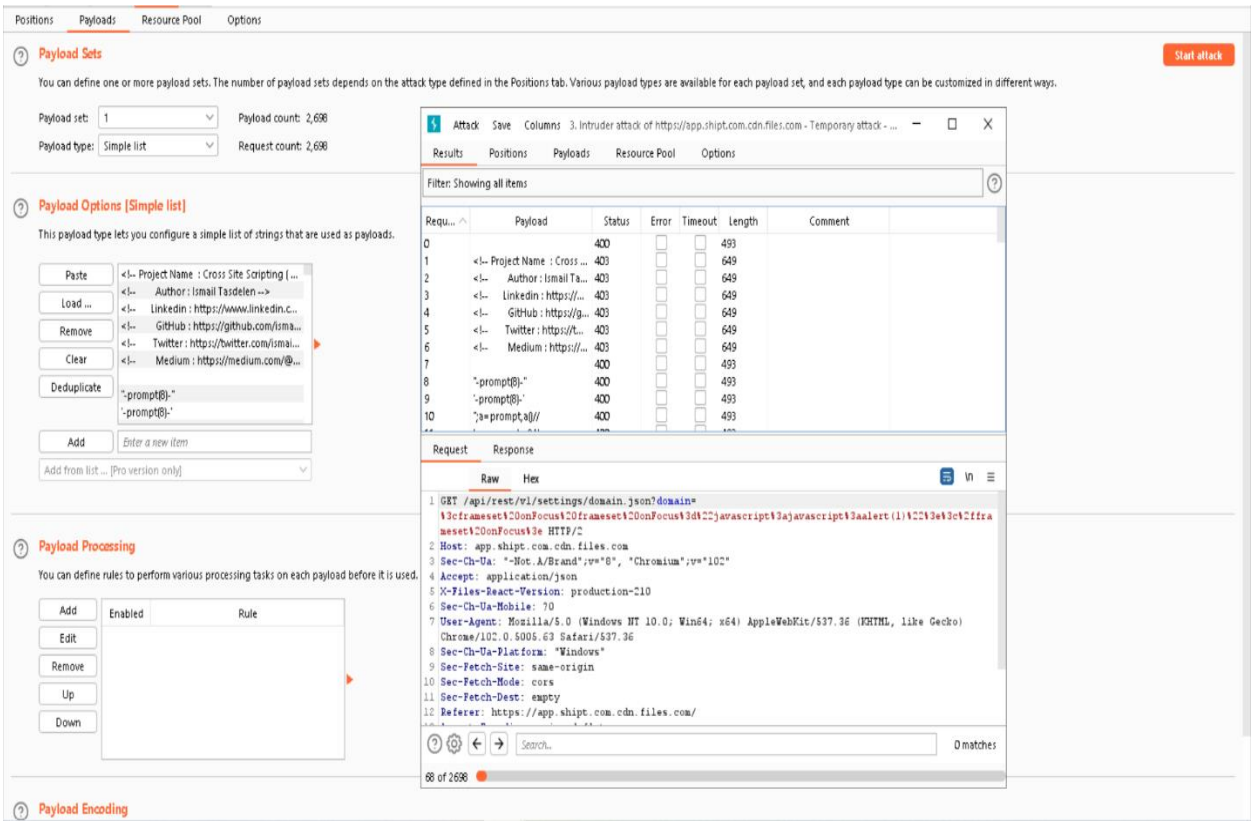


Then I intercepted this website traffic to the burp suite repeater tab.



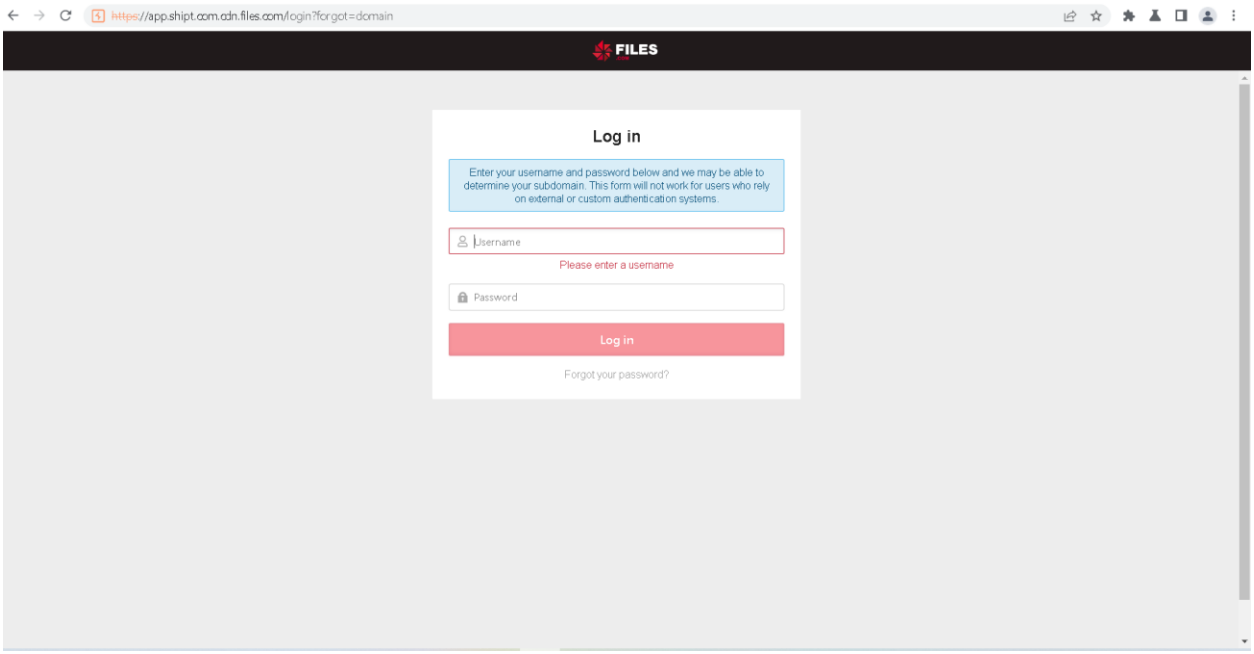


After that, I opened up the intruder tab then I added some XSS payloads to check whether that website is Vulnerable to the XSS.

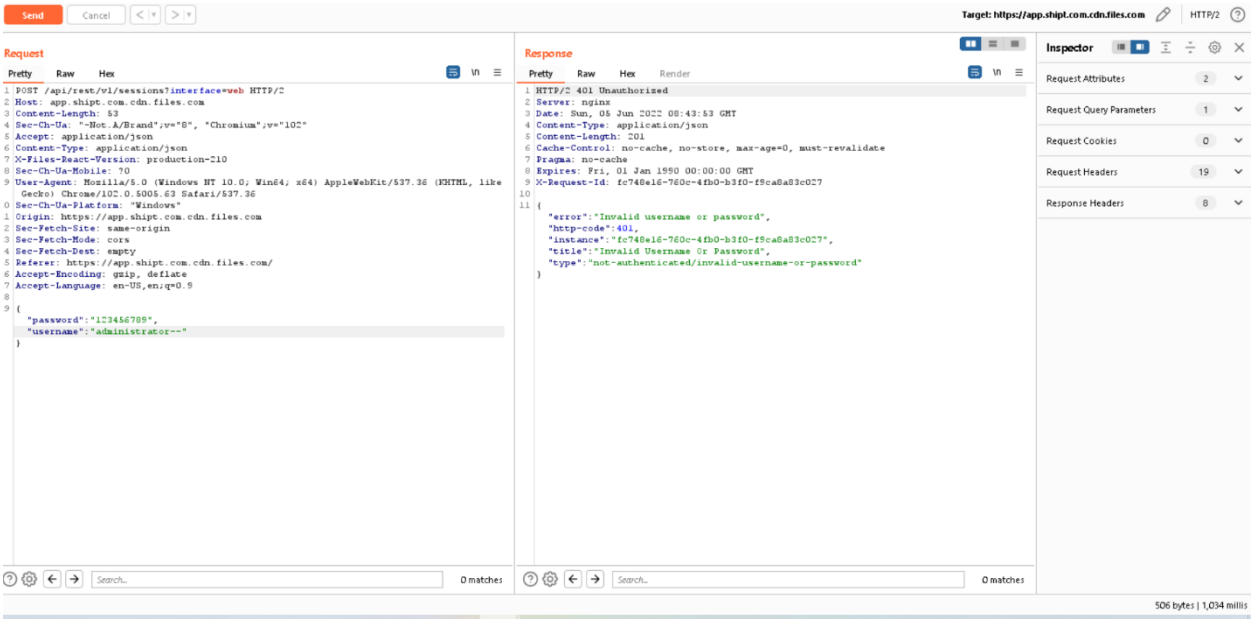


There was no impact by my payload. But According to the header information, XSS is still possible on that website. But, My lack of knowledge in the Scripting and Payload is the reason that I could not get any output.

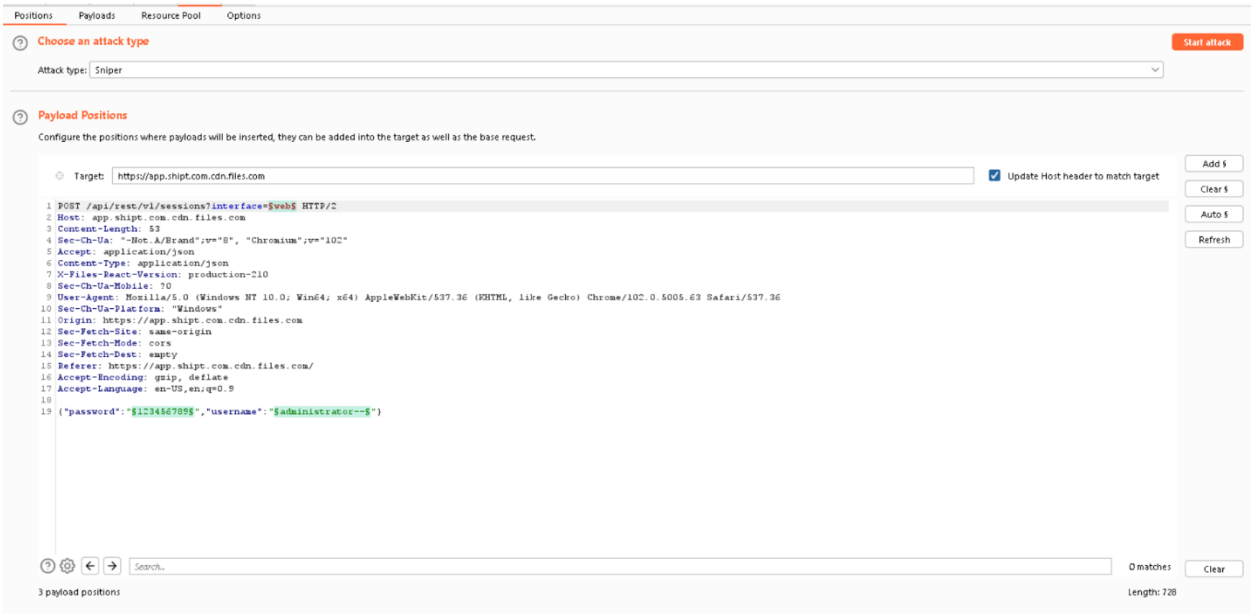
On that page, there was a link redirected to a login form. For an alternate, I thought to check whether it is vulnerable to a SQL injection or not.



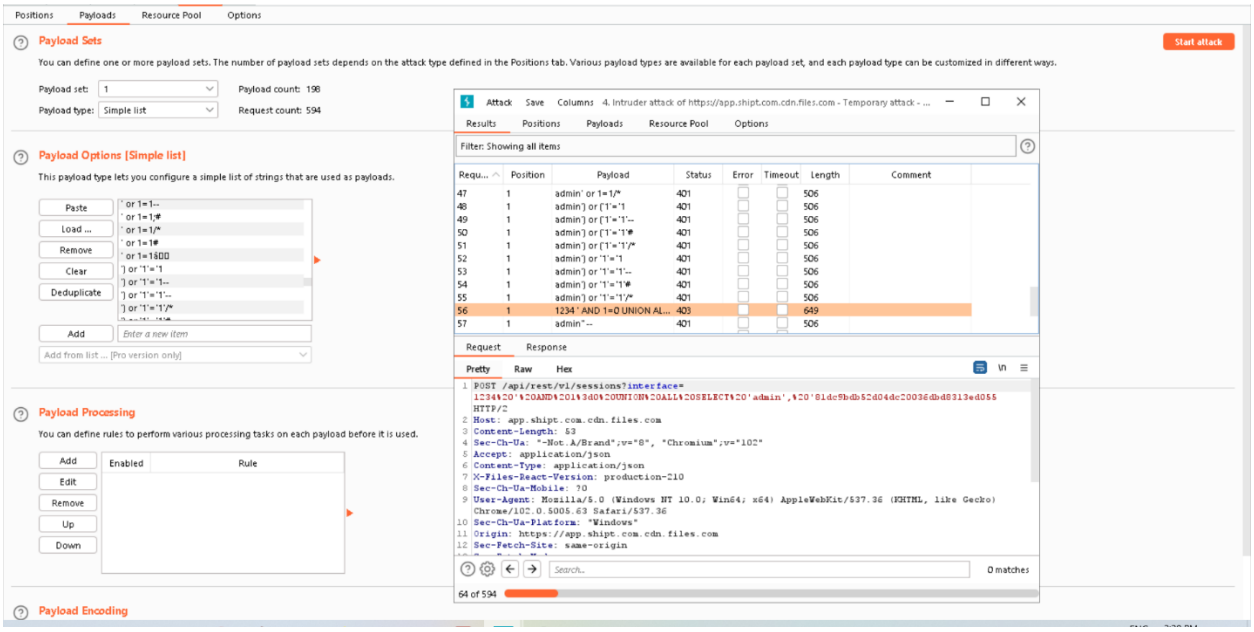
So decided to perform SQL injection to this site as well.



Intercepted the traffic to burp, then sent it back to the repeater.



Then I added some SQL injection payloads.



In final, unfortunately, I could not get any positive response.

Conclusion

The main reason for this web audit is to find security vulnerabilities and report back to the organizations. At first, I started to identify the DNS records and CNAMES of domains listed on a scope list. Then I moved forward to check each subdomain and their DNS entry pointing websites. Then I validated each DNS entry pointing to the website CA that they are expired or not. Then I submitted a Subdomain Takeover vulnerability to **HackerOne**. Then I inspected each website's Response header information. When I was doing it, I found some security issues related to service subdomains. That issues are,

- CSP header not set.
- Cross-domain misconfiguration.
- Missinng anti-clickjacking header.
- Cookie with same-site attribute.
- Server leaks information.
- X-Content-Type-Option header missing.
- Cookie without secure flag.
- Cross-domain JavaScript source file inclusion.

Expected attacks,

- XSS.
- CSRF.
- ClickJacking.
- Information Leak.
- Man in the Middle attack.