Sri Lanka Institute of Information Technology

Web audit Assignment

Individual Assignment

Web.com VDP

IE2062 - Web Security

Submitted by:

Student Registration Number	Student Name
IT20128272	Rathnayake.R.M.K.G

19/10/2021

Date of submission

Disclaimer:

All the information provided on this report are for educational purposes only. The author or the content is no-way responsible for any misuse of the information. Hence please refrain from using any illegal activities!!!

Table of Contents

Acknowledgement

Learning is always not about sitting and listening when the lecturer teaches and writes notes about them, memorizing those things, attending to the exam, and getting passed. We need to understand how we can use what we learn in real-life scenarios outside the classroom. We must understand and practice those things because those things will be essential when we are exposed to the professional industry, especially in the Cybersecurity field.

When we talk about Web Security, we can consider it a crucial part of the Cybersecurity industry. Because of that, we already have a complete module about Web security, and we learn many things that could help us in real-life scenarios. This assignment mainly focuses on using what we learned from Web Security for accurate web application penetration testing. This assignment focuses on using the theories that we learned and gathering more experience and practical knowledge.

I want to thank Dr. Lakmal Rupasinghe and Ms. Chethana Liyanapathirana, who helped and guided me throughout this assignment help to get maximum profit from it and complete it successfully.

Objectives of the Audit

This vulnerability assessment on https://www.web.com for the Web Security module (IE2062) in the second year second semester. The main focus of this assessment is to discover vulnerabilities in the target scope of our selected web application. All the tools and techniques that I am using during the assessment will be summarily described.

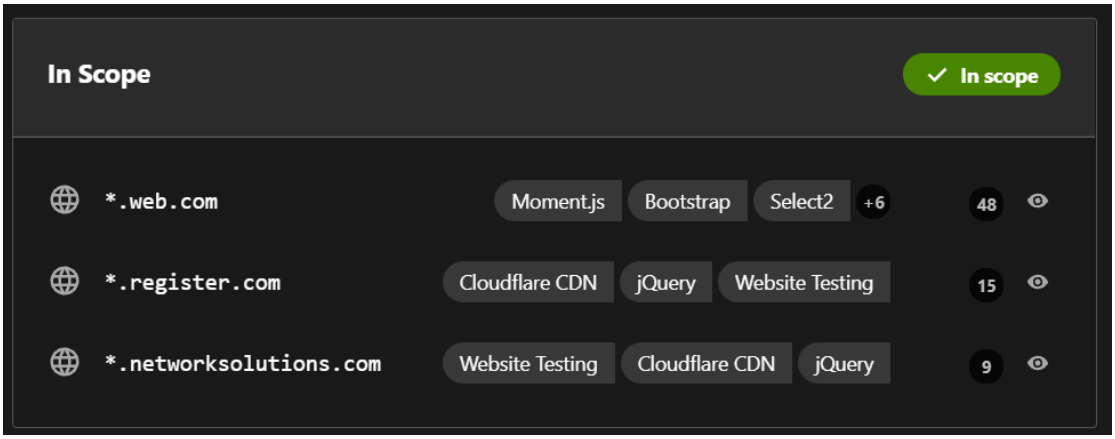
About the target

Web.com (https://www.web.com/) is a platform that is used for Web designing. Customers can easily design their professional web application by themselves or add a store and begin selling products online or have a website professionally designed for them. It is very user-friendly and not much complicated, and it is very time-saving. If users want their website, they have built it from scratch or download a template and altered it but, it does require proper knowledge about web designing languages such as HTML, CSS, JavaScript, PHP, and many other techniques and tools. However, Web.com provides flexible Drag & Drop Website Design, so the user does not need to worry about it. Because of that, many users use Web.com to build their websites for their individual or business purposes, which causes the significant security risk increment of this website. That is the main reason they co-operate with Bugcrowd (https://bugcrowd.com/webdotcom-vdp) and conduct a Bug Bounty Program all most from four years to find vulnerabilities that will cause their safety and reputation and resolve them.

In Scope Targets

(According to the https://bugcrowd.com/webdotcom-vdp)

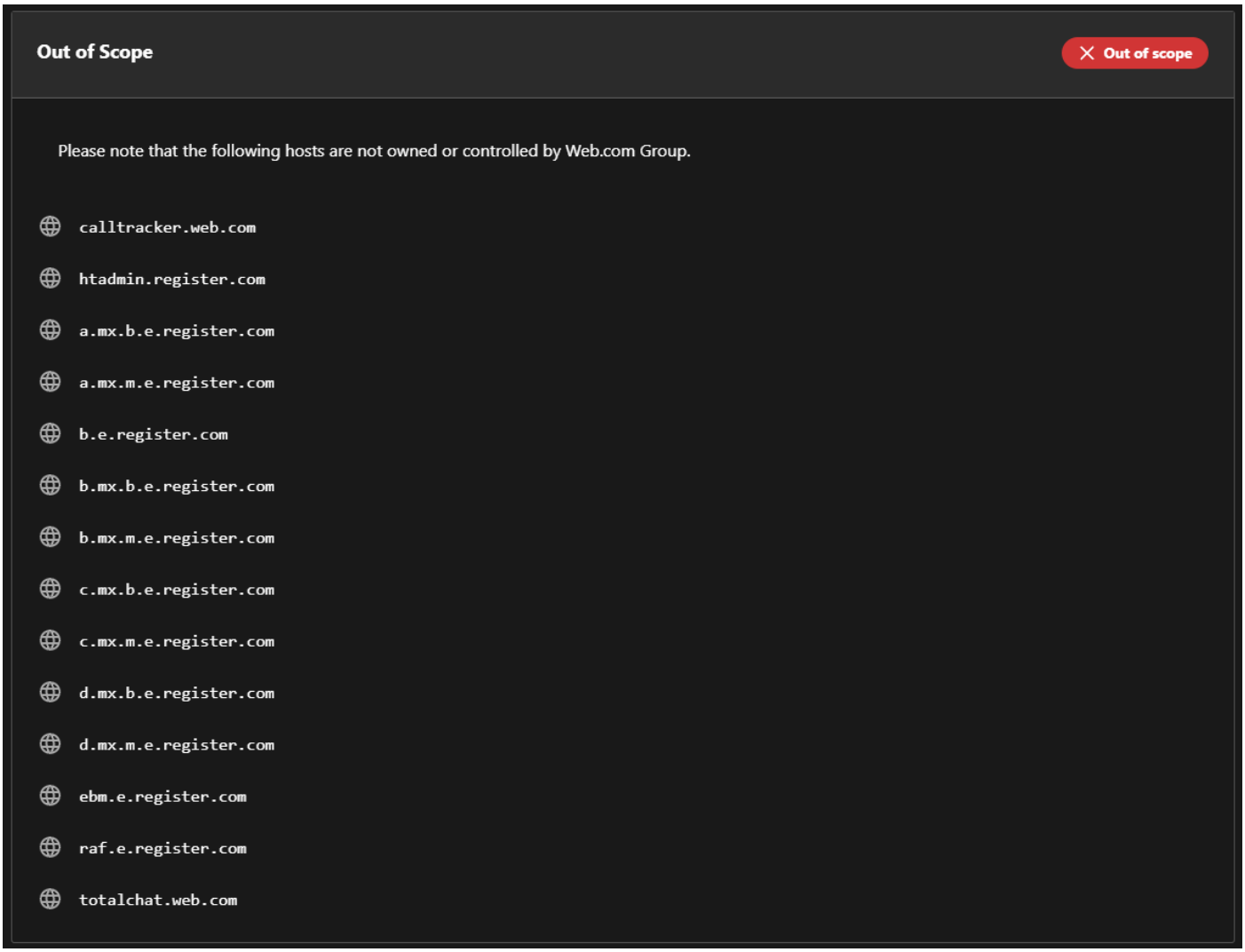
- *.web.com
- *.register.com
- *.networksolutions.com



Out of Scope Targets

(According to the https://bugcrowd.com/webdotcom-vdp)

- calltracker.web.com
- htadmin.register.com
- websitea.mx.b.e.register.com
- websitea.mx.m.e.register.com
- websiteb.e.register.com
- websiteb.mx.b.e.register.com
- websiteb.mx.m.e.register.com
- websitec.mx.b.e.register.com
- websitec.mx.m.e.register.com
- websited.mx.b.e.register.com
- websited.mx.m.e.register.com
- websiteebm.e.register.com
- websiteraf.e.register.com
- websitetotalchat.web.com



Out of Scope

(According to the https://bugcrowd.com/webdotcom/updates)

- Any domain/property of web.com not listed in the targets section is out of scope. (including any/all subdomains not listed above)
- DDos and Application DoS are not permitted.
- Test Chatboxes on the applications is prohibited.
- Test email spoofing is prohibited.

Risk level rate

1. High → This level will represent the highest risk connected with a very specific vulnerability. These vulnerabilities will cause successful exploitation on the targeted website, and the attacker can alter or delete the information of that particular website successfully.

2. Medium → This level will represent the considerable risk connected with a specific vulnerability. An attacker can obtain low-level information about that particular web application when the attacker is exploiting the medium-level risk vulnerability. These kinds of risks should be mitigated after mitigating the high-level risk vulnerabilities.

3. Low → This level will represent the very low risk associated with a very specific vulnerability. An attacker might cause these types of vulnerabilities to obtain information but not very critical level information.

Assessment Methodology

01. Information Gathering

The basic idea of information gathering is the process of collecting all information that you got interested in that particular party. This party could be a person, incident, or organization. In the Cybersecurity industry, we do information gathering to prepare for penetration testing or any hacking activity. It is also known as "Widening the attack surface." It is very crucial to gather information as much as possible about our target before we attempt any hacking. If you want to make a successful attack or deliver a successful vulnerability assessment to your client, you need to spend more time on this phase and gather more information instead of trying to do any exploitation cluelessly. Mainly, there are two objectives in the information-gathering phase.

1. **Collecting network data**

(Ex: - Public domain names, Private domain names, Associated domain names, Network hosts)

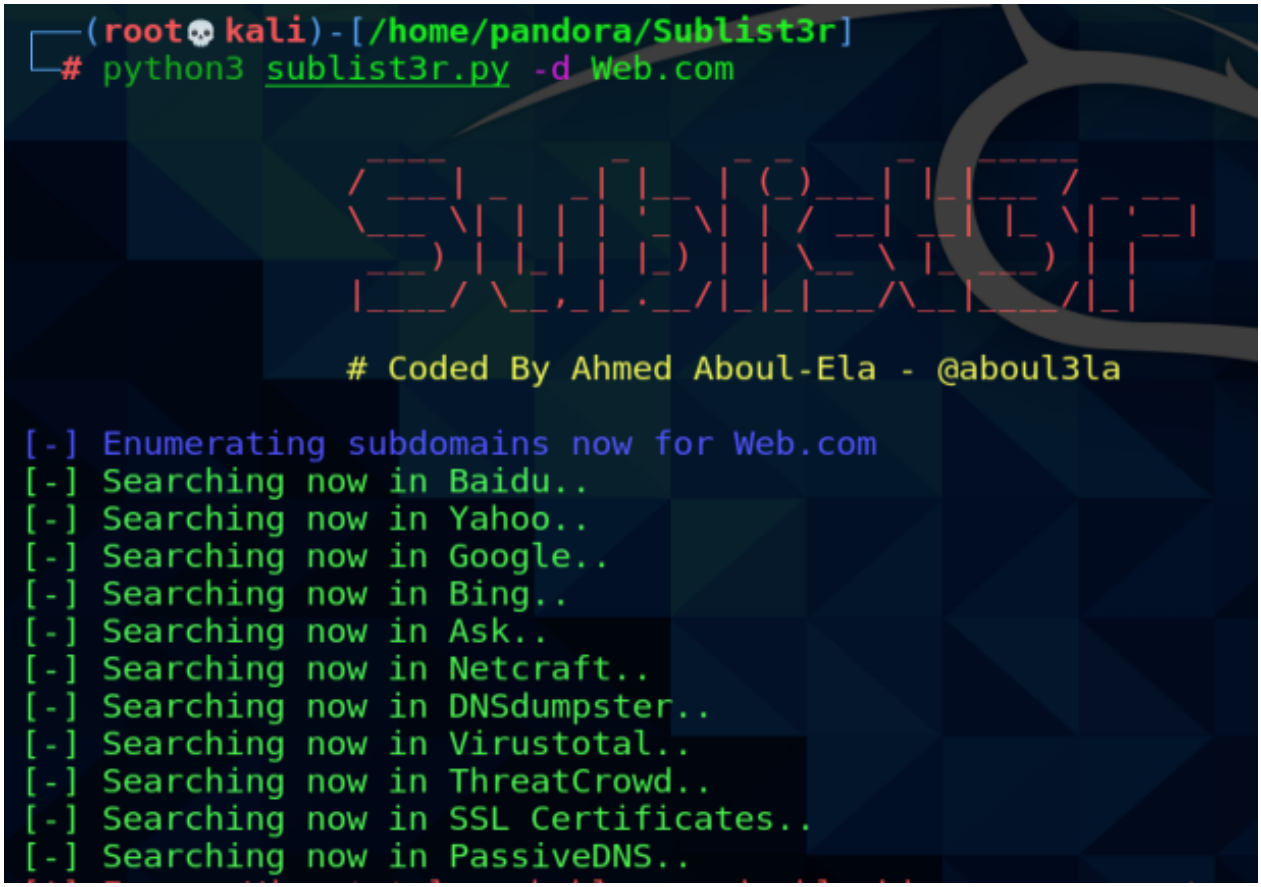
2. **Collecting system-related data**

(Ex: - OS host names, OS system type, System banners)

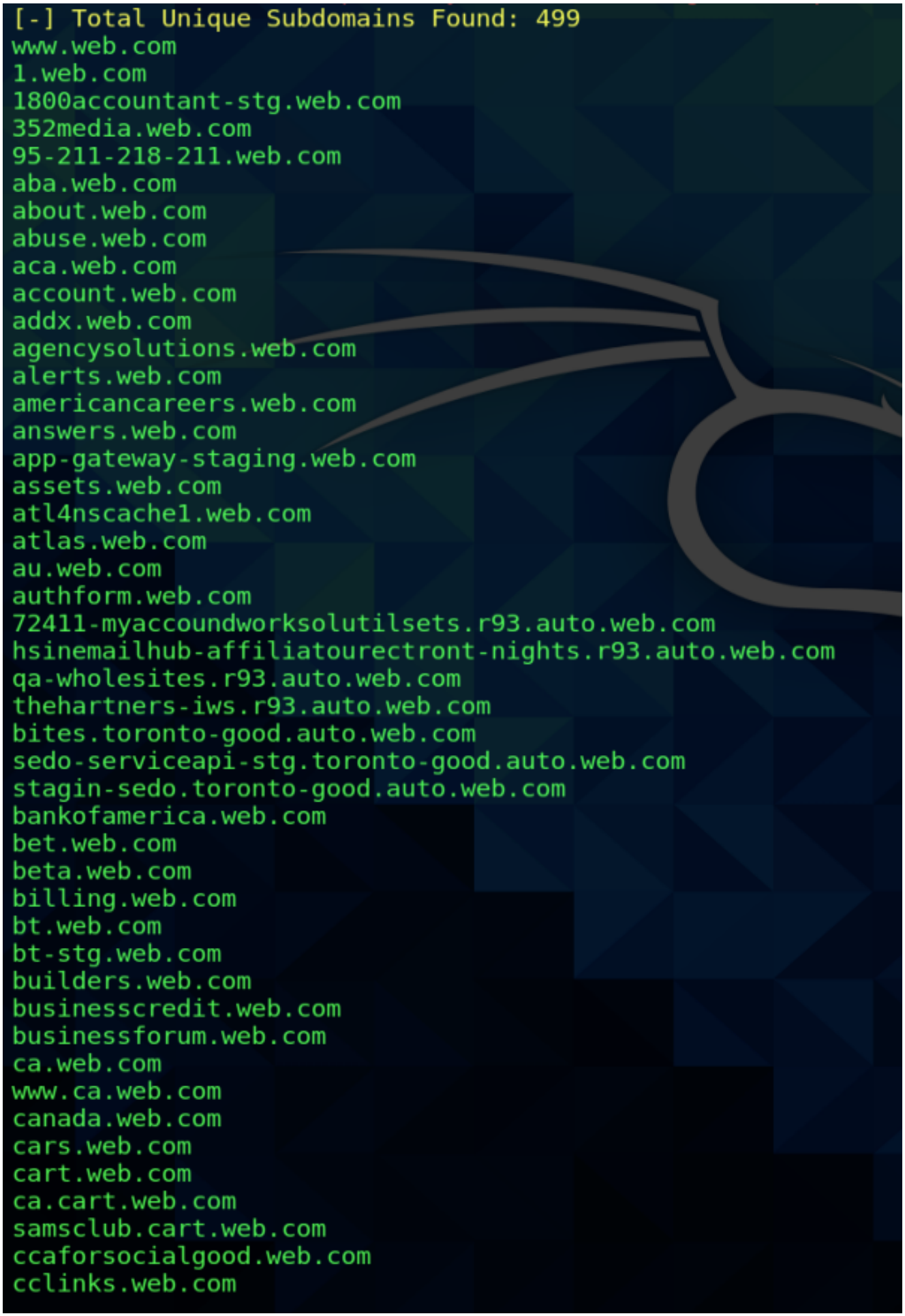
Searching for Sub-domains

A subdomain list is essential because it will help us discover other websites or login forms related to our primary target website. It might consist of vulnerabilities that can be exploited to gain our target system's foothold.

I want to know all sub-domains related to Web.com, and I am going to use one of the best subdomain enumeration tools called "**Sublist3r**." It will help the user to create a virtual sub-domain map of that particular website. Moreover, it has a subroute integration that can perform a brute force subdomain discovery attack with wordlists using Google dorks and other search engines such as Baidu, Ask, Yahoo or Bing.



I found 499 unique subdomains of https://www.web.com.


[-] Total Unique Subdomains Found: 499
www.web.com
1.web.com
1800accountant-stg.web.com
352media.web.com
95-211-218-211.web.com
aba.web.com
about.web.com
abuse.web.com
aca.web.com
account.web.com
addx.web.com
agencysolutions.web.com
alerts.web.com
americancareers.web.com
answers.web.com
app-gateway-staging.web.com
assets.web.com
atl4nscache1.web.com
atlas.web.com
au.web.com
authform.web.com
72411-myaccoundworksolutilsets.r93.auto.web.com
hsinemailhub-affiliatourectront-nights.r93.auto.web.com
qa-wholesites.r93.auto.web.com
thehartners-iws.r93.auto.web.com
bites.toronto-good.auto.web.com
sedo-serviceapi-stg.toronto-good.auto.web.com
stagin-sedo.toronto-good.auto.web.com
bankofamerica.web.com
bet.web.com
beta.web.com
billing.web.com
bt.web.com
bt-stg.web.com
builders.web.com
businesscredit.web.com
businessforum.web.com
ca.web.com
www.ca.web.com
canada.web.com
cars.web.com
cart.web.com
ca.cart.web.com
samsclub.cart.web.com
ccaforsocialgood.web.com
cclinks.web.com

Like the below image, all unique sub-domain names have been displayed on the terminal, but I can save it directly to a text file to use later. This command will store them in a text file.

Alive sub-domains

Even though we got a list of sub-domains, all of them might not be alive. I will use a tool named “httprobe” to extract only alive subdomains out of our sub-domain.txt file, which we created previously by using the Sublist3r tool.

```
┌──(root💀kali)-[/home/pandora/Desktop/Sublist3r]
└─# cat /home/pandora/Desktop/sub-domain.txt | httprobe >> /home/pandora/Desktop/alive-subdomain.txt
```

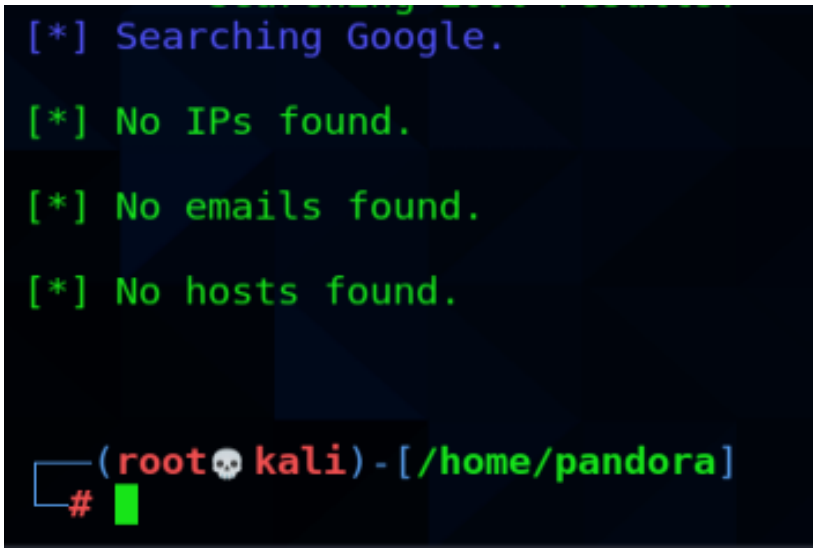
I stored the result of httprobe in a text file called “alive-subdomain.txt,” and the content should be displayed as follows.

```
┌──(root💀kali)-[/home/pandora/Desktop/Sublist3r]
└─# cat /home/pandora/Desktop/alive-subdomain.txt
http://1800accountant-stg.web.com
http://www.web.com
http://account.web.com
http://app-gateway-staging.web.com
https://account.web.com
http://about.web.com
http://aba.web.com
https://1800accountant-stg.web.com
http://abuse.web.com
http://aca.web.com
https://app-gateway-staging.web.com
https://www.web.com
http://352media.web.com
http://agencysolutions.web.com
http://answers.web.com
https://abuse.web.com
http://assets.web.com
http://au.web.com
http://atlas.web.com
https://assets.web.com
https://about.web.com
https://aca.web.com
https://aba.web.com
https://answers.web.com
https://352media.web.com
https://72411-myaccoundworksolutilsets.r93.auto.web.com
http://hsinemailhub-affiliatourectront-nights.r93.auto.web.com
https://agencysolutions.web.com
http://qa-wholesites.r93.auto.web.com
https://hsinemailhub-affiliatourectront-nights.r93.auto.web.com
https://atlas.web.com
http://sedo-serviceapi-stg.toronto-good.auto.web.com
http://thehartners-iws.r93.auto.web.com
http://bites.toronto-good.auto.web.com
https://qa-wholesites.r93.auto.web.com
https://thehartners-iws.r93.auto.web.com
https://bites.toronto-good.auto.web.com
https://sedo-serviceapi-stg.toronto-good.auto.web.com
https://au.web.com
http://stagin-sedo.toronto-good.auto.web.com
http://bet.web.com
http://bt-stg.web.com
http://bt.web.com
https://stagin-sedo.toronto-good.auto.web.com
http://bankofamerica.web.com
https://bet.web.com
```

Harvesting E-mails, usernames and passwords

Now, we are trying to discover some emails or usernames and passwords from leaked databases on the internet. For that, I will use a tool called **"theHarvester"** to gather subdomains and email addresses. We need to specify the target domain, the search engine, and the length of the search.

Great, I could not find any information which related to web.com.



Discovering archived information

Sometimes, we can discover very useful information snapshots or backup files of the websites, known as “Archived information.” Even though it is the past of our targeted website, they might be very useful for our work.

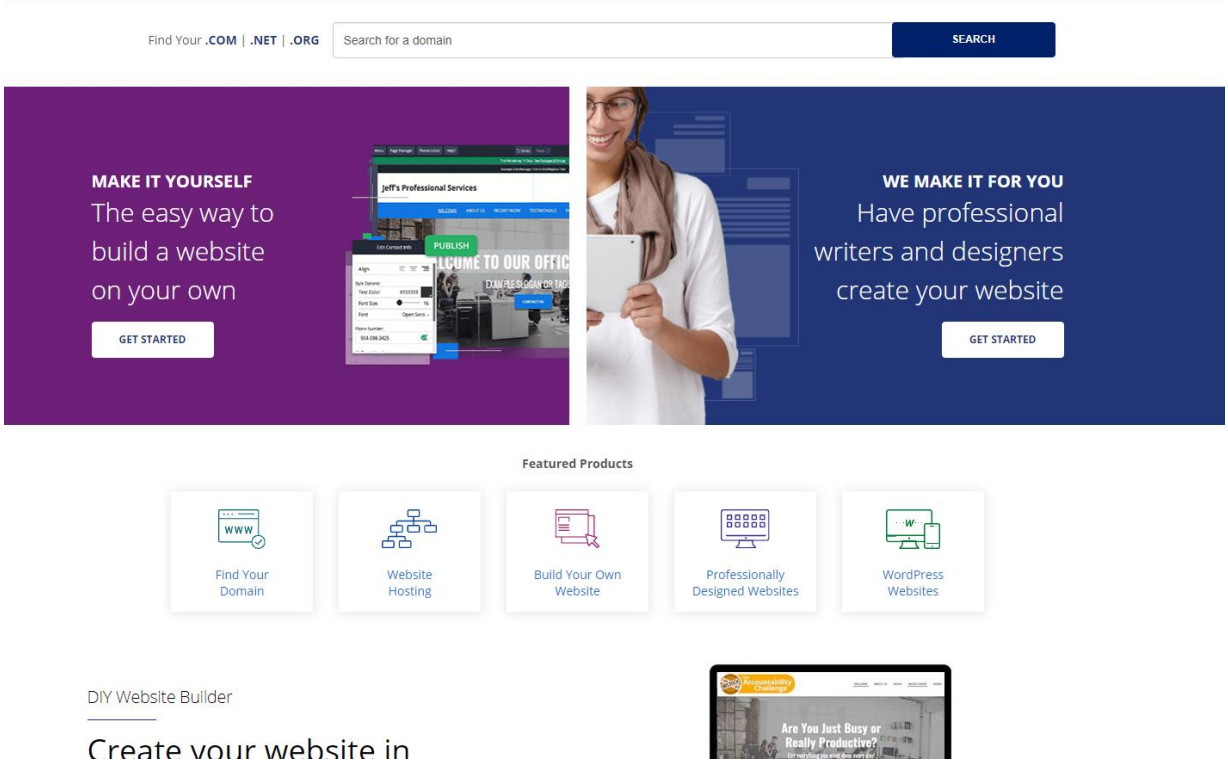
I will use an internet archive called **“Way Back Machine”** to seek some hidden past of our targeted website. It contains a massive database of archived information such as snapshots, audios, videos, and copies of web pages, and many more.

Website link: - http://archive.org/web/

These are the some of the information that I found about our targeted website by using Way Back Machine.

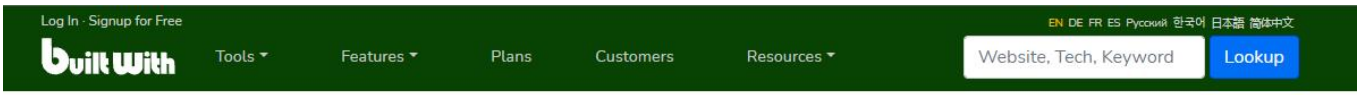
This is a snapshot from 01st January, 2011.

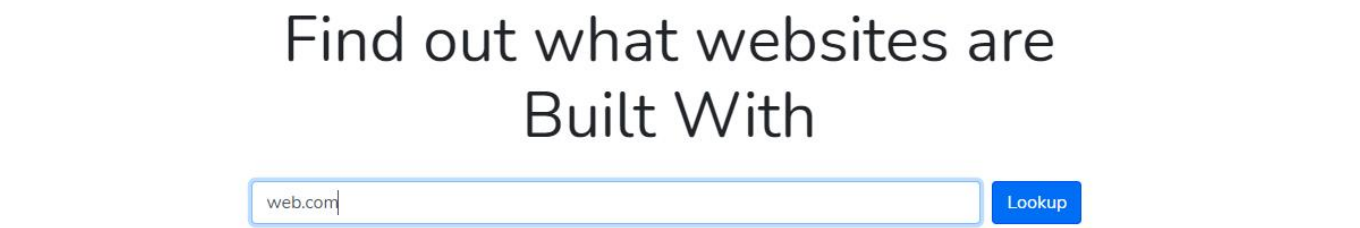
This is a snapshot from 01st January, 2020.

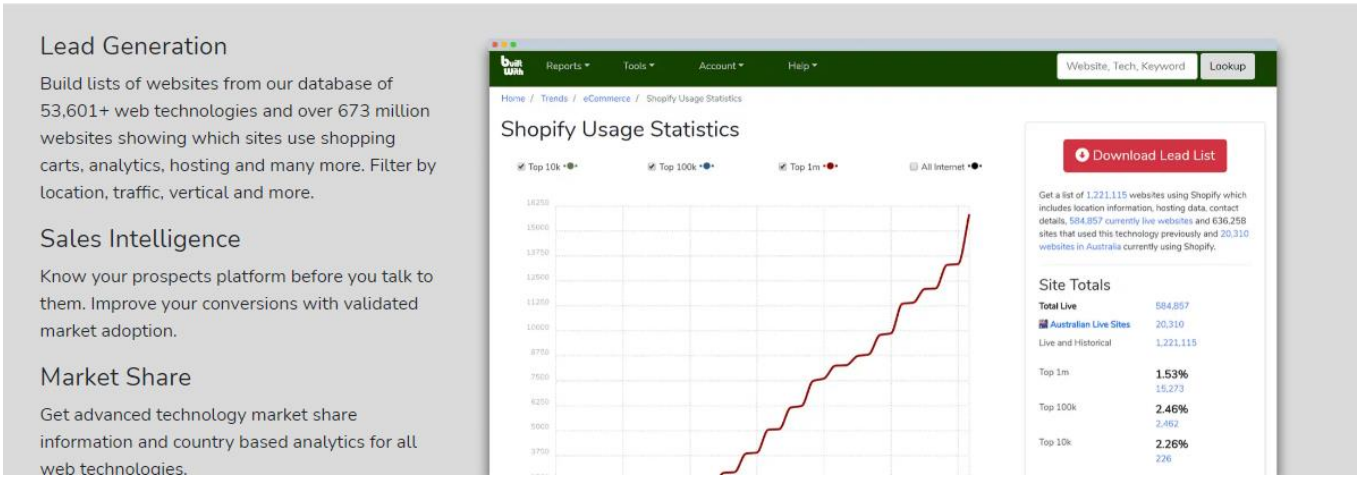


Identifying Website Technologies

There is a way to understand the technologies about our targeted website because it might be useful in our vulnerability assessment. For that, we can use a website called **"builtwith.com"** (https://builtwith.com). We can find out technologies which are used in our targeted website.







These are the some of the results that I got for my targeted website which is web.com

Search Results for *web.com*

Technology Matches

Web.com

Web.com Usage Statistics · Download List of All Websites using Web.com

Web.com is a U.S.-based company that creates affordable websites, online marketing campaigns and eCommerce stores for small and medium businesses.

Content Management System · Hosted Solution · Simple Website Builder

Web.com DNS

Web.com DNS Usage Statistics · Download List of All Websites using Web.com DNS

DNS services provided by Web.com.

Name Server

Web.com Email

Web.com Email Usage Statistics · Download List of All Websites using Web.com Email

Web.com email hosting.

Email Hosting Providers · Web Hosting Provider Email

Web.com Hosting

Web.com Hosting Usage Statistics · Download List of All Websites using Web.com Hosting

Web hosting solutions from web.com

Web Hosting Providers · US hosting

Web.com 2013 Site Builder

Web.com 2013 Site Builder Usage Statistics · Download List of All Websites using Web.com 2013 Site Builder

Web.com's more recent site builder service.

Content Management System

Clicinfo-web

Clicinfo-web Usage Statistics · Download List of All Websites using Clicinfo-web

France based digital agency.

Content Management System · Agency

First Call Web

DNS enumeration

There are so many tools for examples, DNSRecon, Nmap and DNSSenum that can use to obtain details of domain name systems.

1. nmap

Usually, we use Nmap to scan the ports of a system. However, in this scenario, we can use it for different tasks. There is a built-in "dns-brute" script in Nmap, and I will use it to obtain the DNS information of the web.com. It is used as follows.

```
┌──(root💀kali)-[/home/pandora]
└─# nmap -T4 -p 53 --script dns-brute web.com
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-10 23:06 +0530
Nmap scan report for web.com (162.159.133.36)
Host is up (0.0027s latency).
Other addresses for web.com (not scanned): 162.159.130.36

PORT   STATE    SERVICE
53/tcp filtered domain
```

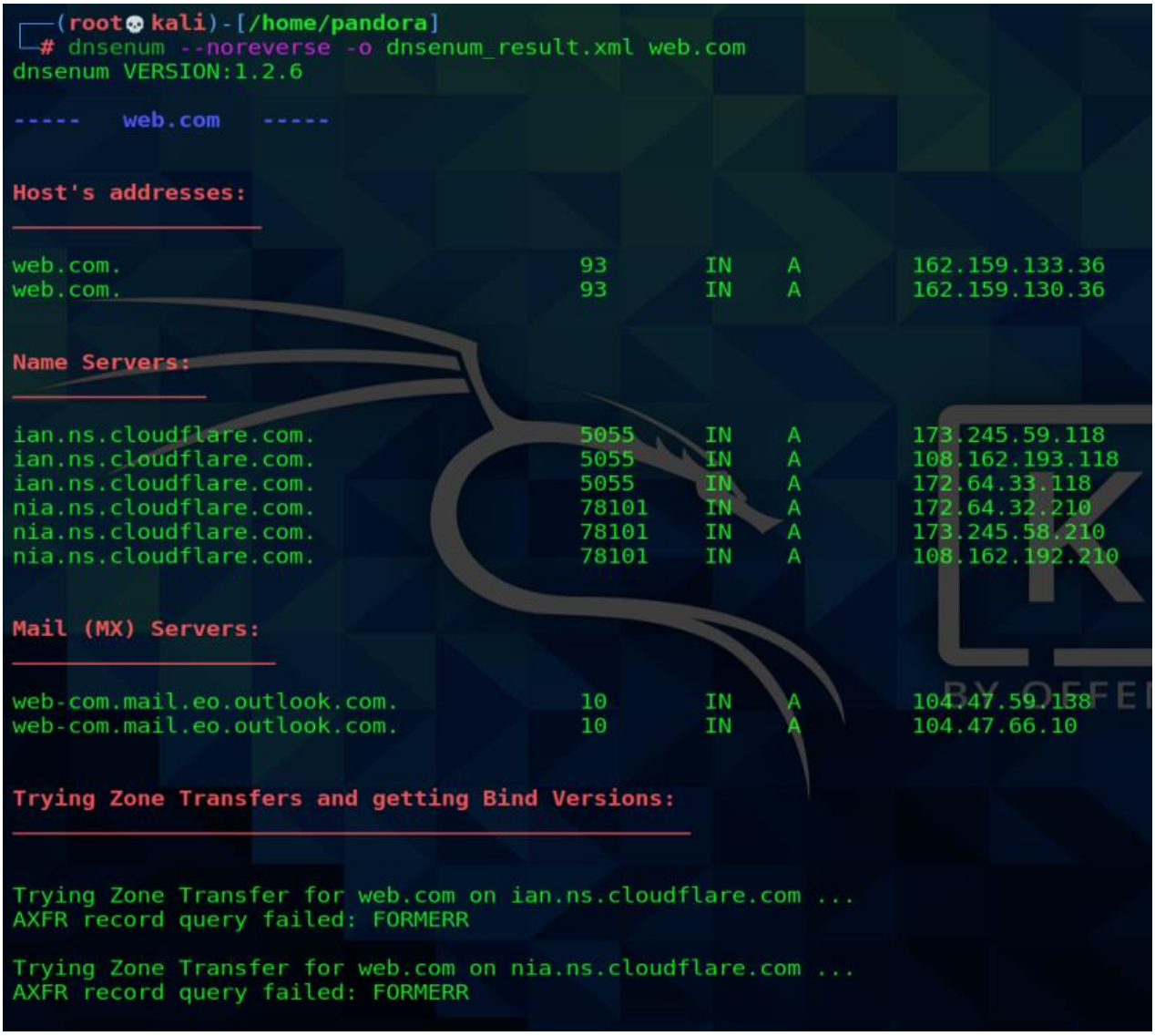
The result can be seen as follows.

```
Host script results:
| dns-brute:
|   DNS Brute-force hostnames:
|     mx.web.com - 209.237.135.69
|     id.web.com - 145.239.170.42
|     test.web.com - 38.101.198.41
|     images.web.com - 64.69.220.61
|     alerts.web.com - 207.204.50.196
|     info.web.com - 145.239.170.42
|     noc.web.com - 192.168.24.142
|     dns.web.com - 63.123.39.248
|     internet.web.com - 127.0.0.1
|     app.web.com - 104.18.24.158
|     app.web.com - 104.18.25.158
|     ns1.web.com - 207.204.40.31
|     app.web.com - 2606:4700::6812:189e
|     app.web.com - 2606:4700::6812:199e
|     dns1.web.com - 207.204.40.31
|     ns2.web.com - 207.204.21.31
|     dns2.web.com - 207.204.21.31
|     ns3.web.com - 207.204.40.31
|     vpn.web.com - 64.69.210.40
|     download.web.com - 209.237.135.69
|     web.web.com - 209.237.135.69
|     beta.web.com - 64.226.3.45
|     owa.web.com - 209.17.114.40
|     local.web.com - 145.239.170.100
|     whois.web.com - 64.69.216.61
|     whois.web.com - 2607:f408:1010:a:64:69:216:61
|     cdn.web.com - 216.21.224.222
|     www.web.com - 162.159.130.36
|     www.web.com - 162.159.133.36
|     chat.web.com - 209.237.135.69
|     secure.web.com - 216.21.224.199
|     www2.web.com - 209.237.135.69
|     forum.web.com - 209.237.135.69
|     shop.web.com - 127.0.0.1
|     corp.web.com - 209.17.115.13
|     help.web.com - 216.21.227.50
|_    demo.web.com - 64.69.219.92

Nmap done: 1 IP address (1 host up) scanned in 12.00 seconds
```

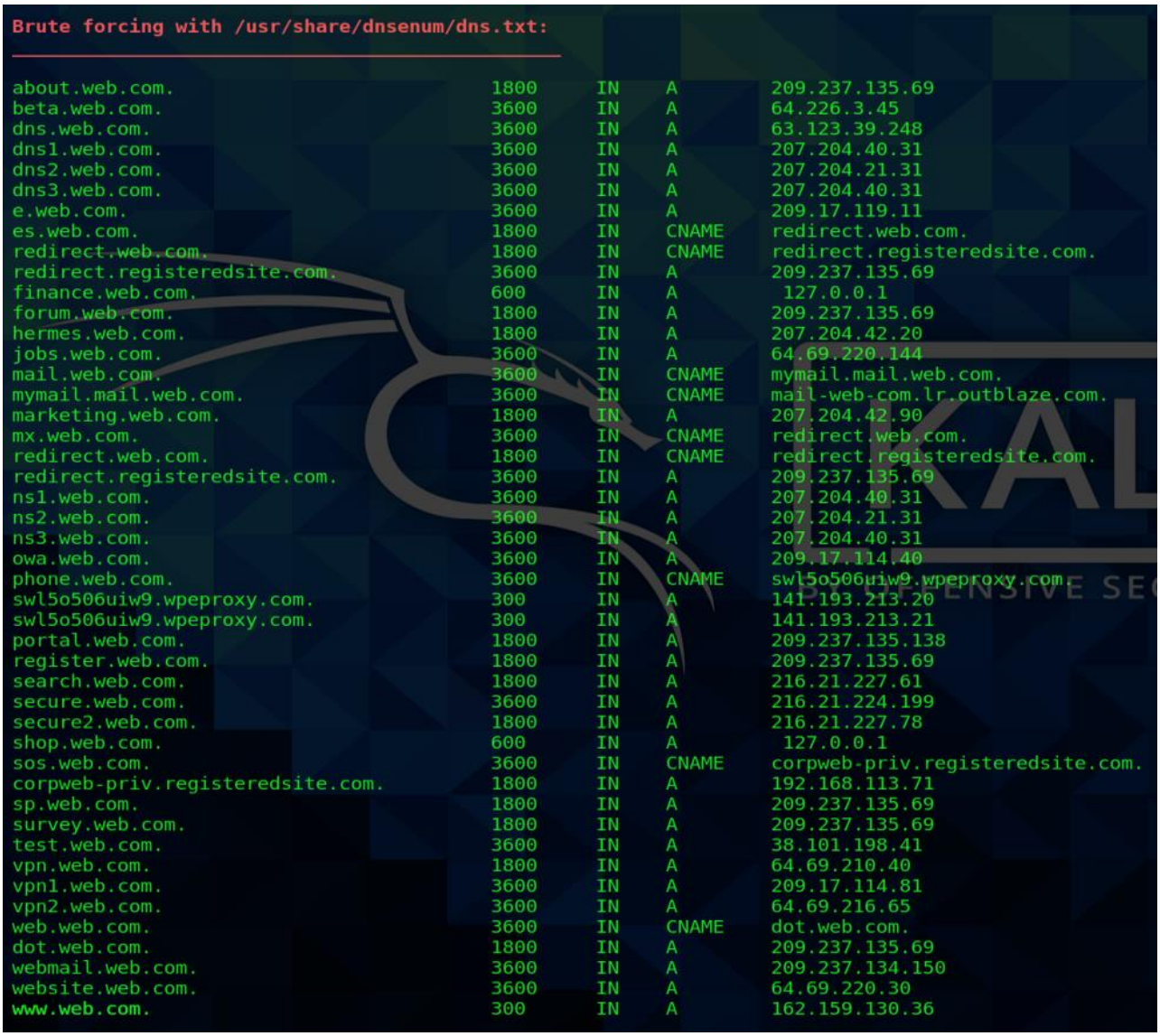
2. DNSenum

If you want to get DNS information more clearly than Nmap, the best tool is DNSenum. It is very simple to use, and I will use it without reverse lookup and save the result to an XML document.

The usage of the DNSenum as follows.



Not like Nmap output, this tool's output is very easy to understand.



```
Brute forcing with /usr/share/dnsenum/dns.txt:
_____________________________________________

about.web.com.                          1800    IN    A          209.237.135.69
beta.web.com.                           3600    IN    A          64.226.3.45
dns.web.com.                            3600    IN    A          63.123.39.248
dns1.web.com.                           3600    IN    A          207.204.40.31
dns2.web.com.                           3600    IN    A          207.204.21.31
dns3.web.com.                           3600    IN    A          207.204.40.31
e.web.com.                              3600    IN    A          209.17.119.11
es.web.com.                             1800    IN    CNAME      redirect.web.com.
redirect.web.com.                       1800    IN    CNAME      redirect.registeredsite.com.
redirect.registeredsite.com.            3600    IN    A          209.237.135.69
finance.web.com.                        600     IN    A          127.0.0.1
forum.web.com.                          1800    IN    A          209.237.135.69
hermes.web.com.                         1800    IN    A          207.204.42.20
jobs.web.com.                           3600    IN    A          64.69.220.144
mail.web.com.                           3600    IN    CNAME      mymail.mail.web.com.
mymail.mail.web.com.                    3600    IN    CNAME      mail-web-com.lr.outblaze.com.
marketing.web.com.                      1800    IN    A          207.204.42.90
mx.web.com.                             3600    IN    CNAME      redirect.web.com.
redirect.web.com.                       1800    IN    CNAME      redirect.registeredsite.com.
redirect.registeredsite.com.            3600    IN    A          209.237.135.69
ns1.web.com.                            3600    IN    A          207.204.40.31
ns2.web.com.                            3600    IN    A          207.204.21.31
ns3.web.com.                            3600    IN    A          207.204.40.31
owa.web.com.                            3600    IN    A          209.17.114.40
phone.web.com.                          3600    IN    CNAME      swl5o506uiw9.wpeproxy.com.
swl5o506uiw9.wpeproxy.com.              300     IN    A          141.193.213.20
swl5o506uiw9.wpeproxy.com.              300     IN    A          141.193.213.21
portal.web.com.                         1800    IN    A          209.237.135.138
register.web.com.                       1800    IN    A          209.237.135.69
search.web.com.                         1800    IN    A          216.21.227.61
secure.web.com.                         3600    IN    A          216.21.224.199
secure2.web.com.                        1800    IN    A          216.21.227.78
shop.web.com.                           600     IN    A          127.0.0.1
sos.web.com.                            3600    IN    CNAME      corpweb-priv.registeredsite.com.
corpweb-priv.registeredsite.com.        1800    IN    A          192.168.113.71
sp.web.com.                             1800    IN    A          209.237.135.69
survey.web.com.                         1800    IN    A          209.237.135.69
test.web.com.                           3600    IN    A          38.101.198.41
vpn.web.com.                            1800    IN    A          64.69.210.40
vpn1.web.com.                           3600    IN    A          209.17.114.81
vpn2.web.com.                           3600    IN    A          64.69.216.65
web.web.com.                            3600    IN    CNAME      dot.web.com.
dot.web.com.                            1800    IN    A          209.237.135.69
webmail.web.com.                        3600    IN    A          209.237.134.150
website.web.com.                        3600    IN    A          64.69.220.30
www.web.com.                            300     IN    A          162.159.130.36
```

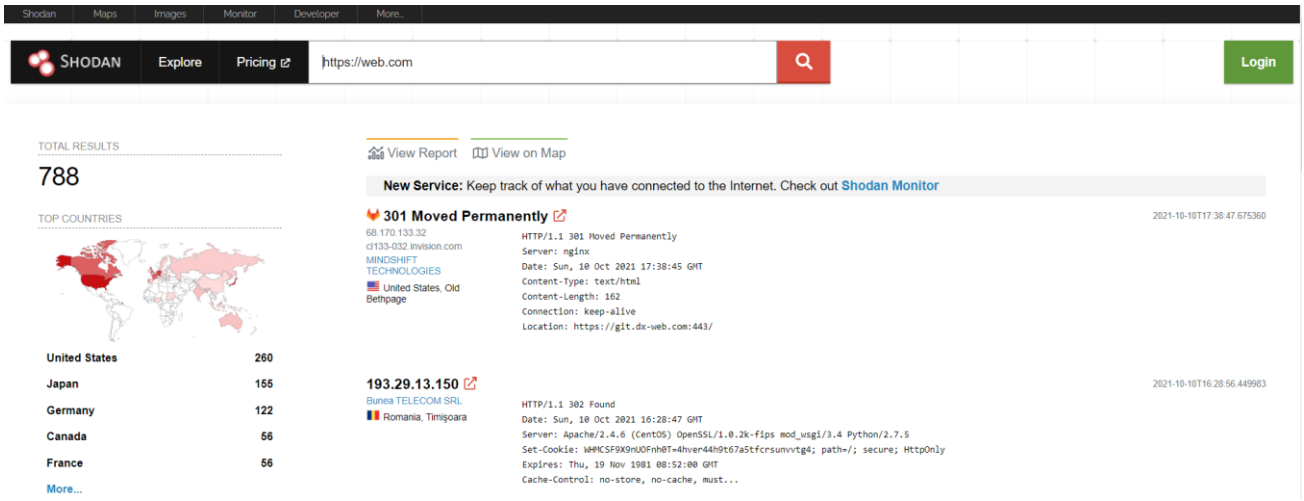

web.com class C netranges:
38.101.198.0/24
63.123.39.0/24
64.69.210.0/24
64.69.216.0/24
64.69.220.0/24
64.226.3.0/24
162.159.130.0/24
162.159.133.0/24
207.204.21.0/24
207.204.40.0/24
207.204.42.0/24
209.17.114.0/24
209.17.119.0/24
209.237.134.0/24
209.237.135.0/24
216.21.224.0/24
216.21.227.0/24
web.com ip blocks:
38.101.198.41/32
63.123.39.248/32
64.69.210.40/32
64.69.216.65/32
64.69.220.30/32
64.69.220.144/32
64.226.3.45/32
162.159.130.36/32
162.159.133.36/32
207.204.21.31/32
207.204.40.31/32
207.204.42.20/32
207.204.42.90/32
209.17.114.40/32
209.17.114.81/32
209.17.119.11/32
209.237.134.150/32
209.237.135.69/32
209.237.135.138/32
216.21.224.199/32
216.21.227.61/32
216.21.227.78/32
done.

Internet connected(public) Device Enumeration

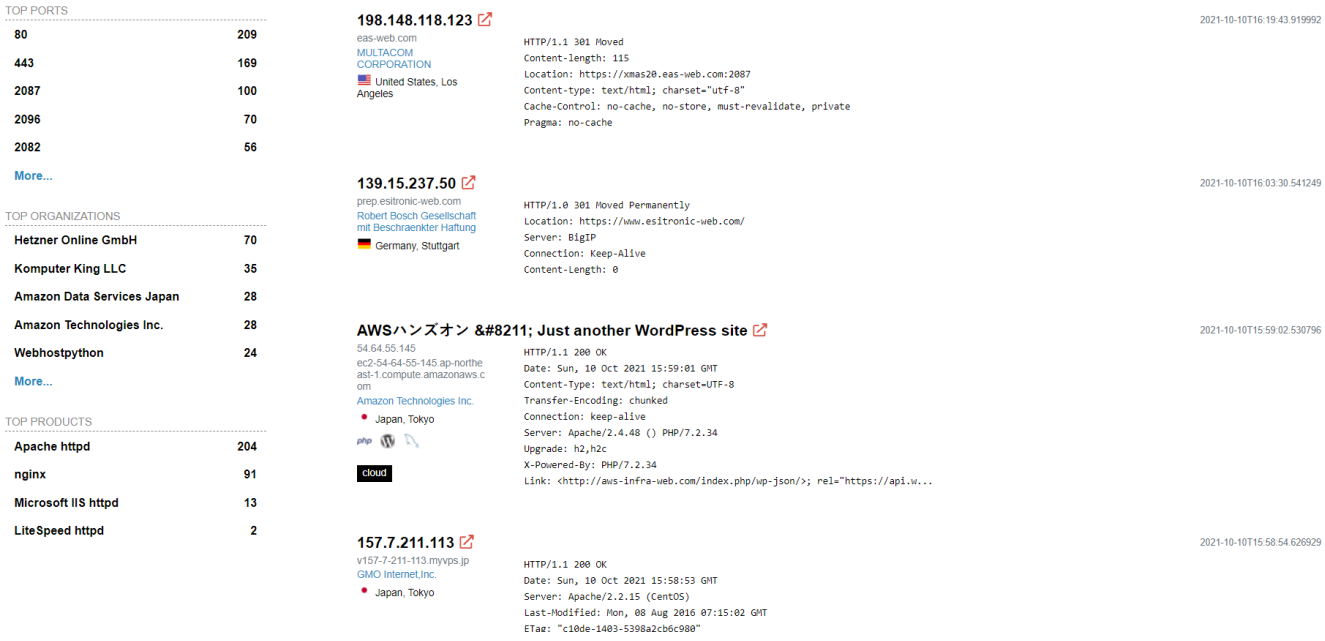
Shodan

This is online website that we can use it for seek information such as IPs, ISP, SSH, FTP, web server details and banners about the target domain's all the public devices.

Web site link: - https://www.shodan.io/



These are some of the results that I got about the web.com.



02. Footprinting and Scanning

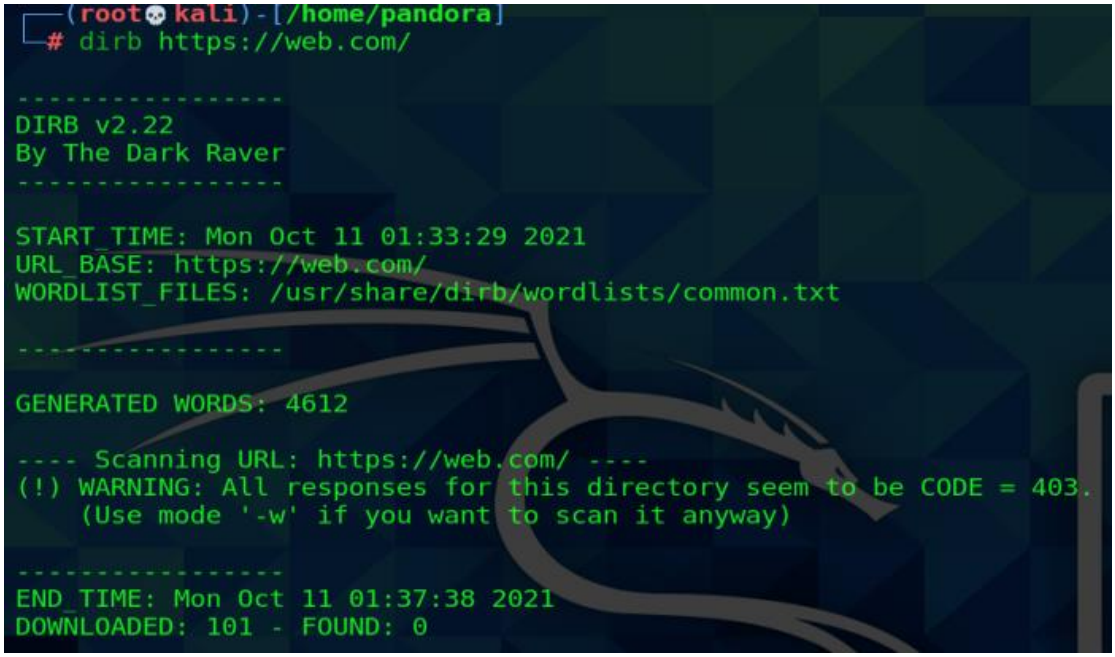
As penetration testers, we are only seeking information in our targeted system. However, in this phase (footprinting and scanning phase), we have to involve our targeted system actively. This phase is also known as the active information gathering phase since this phase is also much important as our first phase before we jump in to do any vulnerability assessment.

Brute Forcing Directories

In this scenario, we are trying to search and discover some hidden directories of our targeted website. If there is any such kind of hidden directories, the information of those directories will be very useful for us.

1. Dirb tool

Dirb is a web content scanner that we can use for brute force directories with a dictionary attack against the web server. I am going to use this scan web.com.

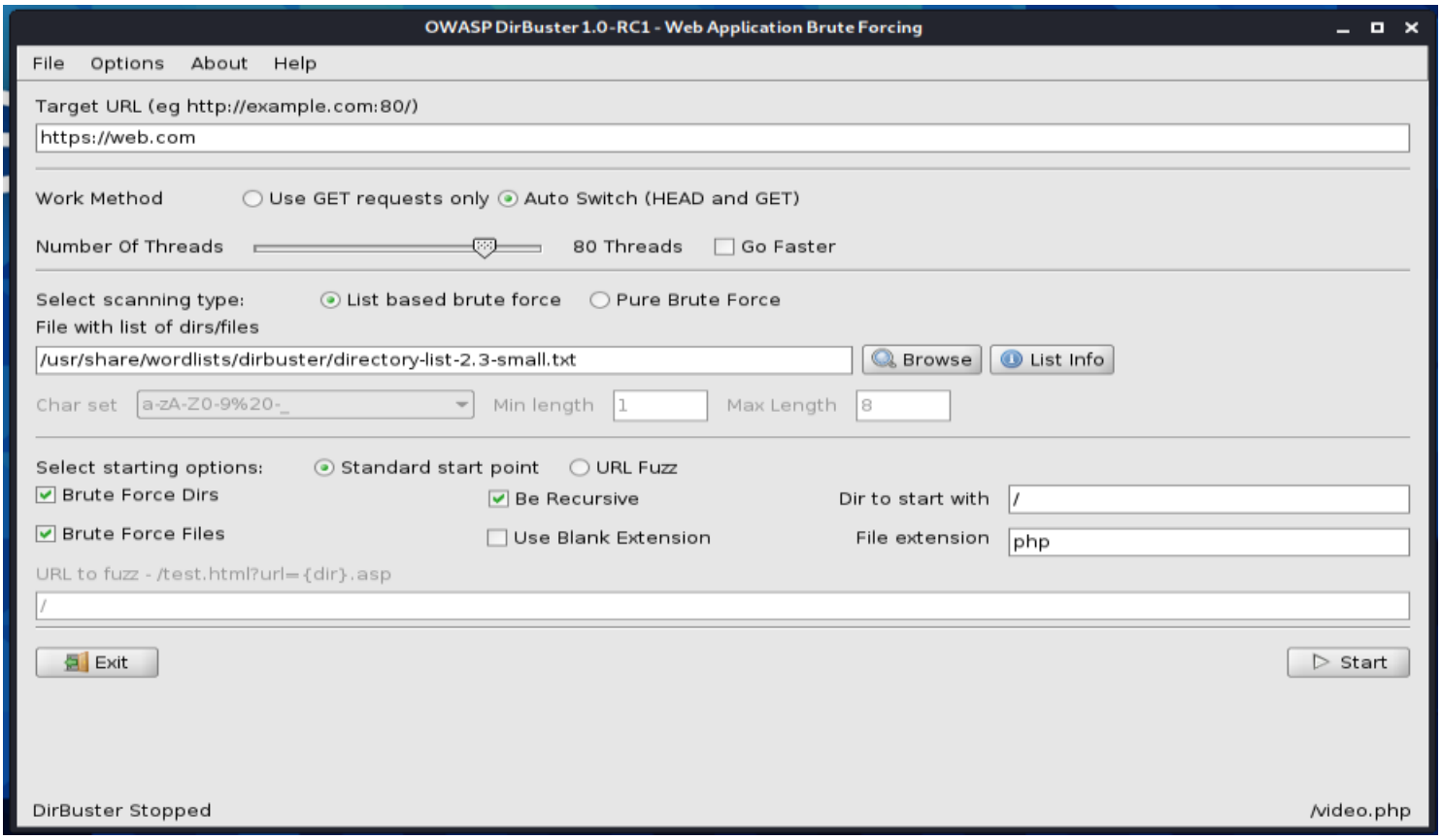


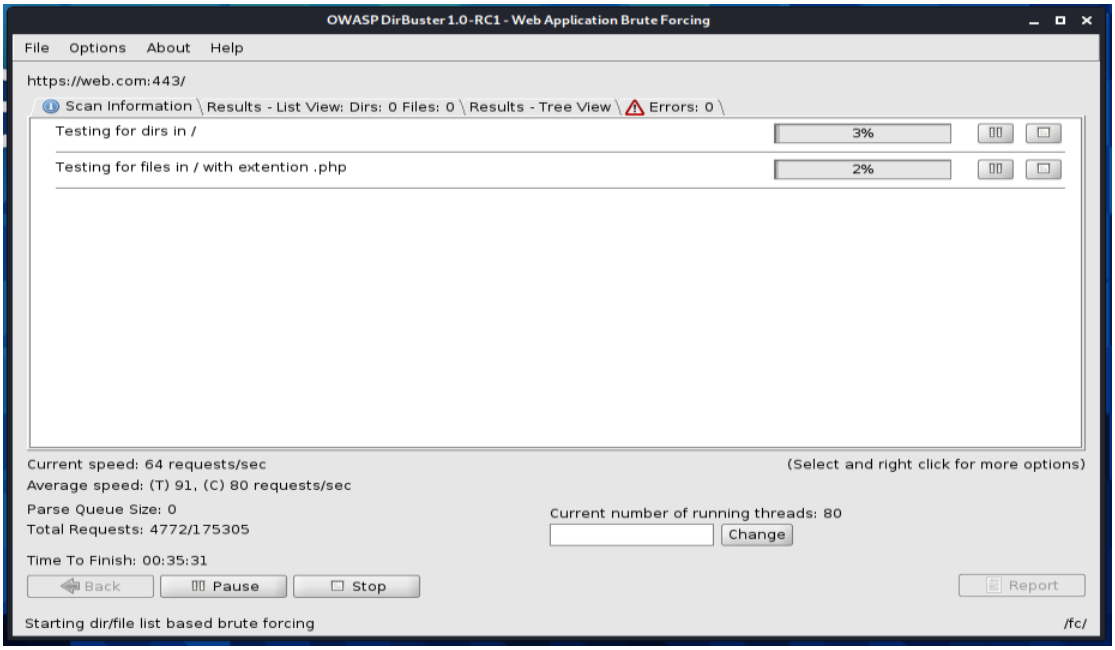
And, I did not find anything regarding to my targeted domain.

2. OWASP DirBuster

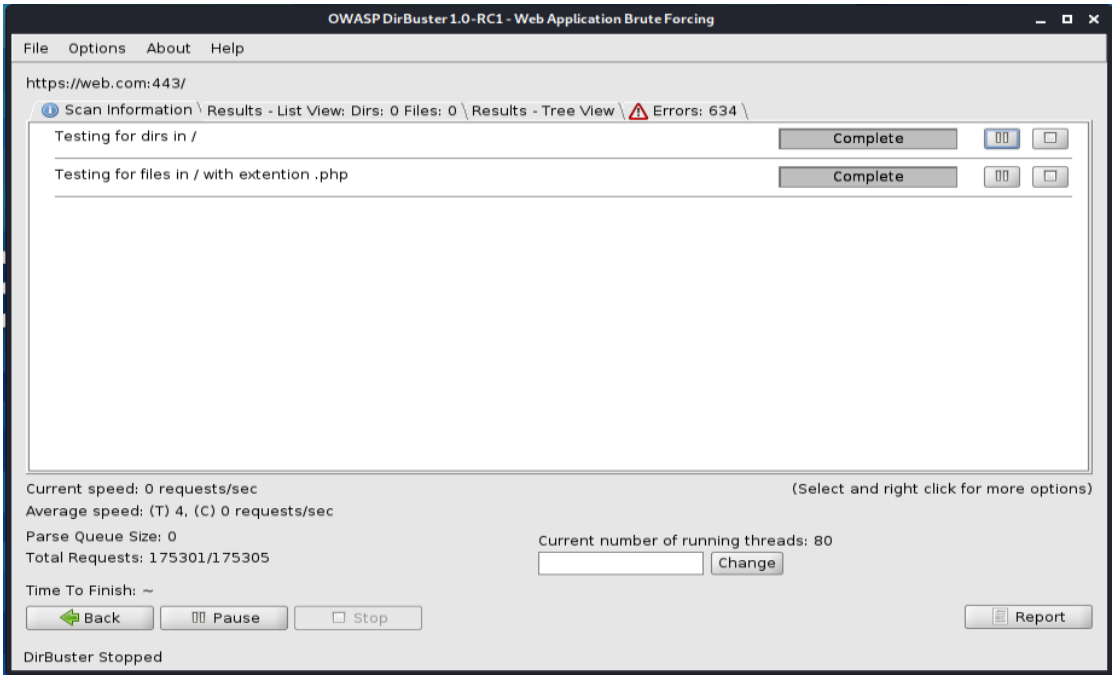
OWASP Dirbuster is a tool designed by using Java programming language, and it is used for brute force directories and files by using its wordlists. This tool can find hidden web pages of our targeted website. Even though Dirbuster has nine-word lists sometimes, all the word lists can be failed. However, this tool's pure brute force can use to discover hidden files.

Nevertheless, the problem is, this is a very time-consuming process.





But unfortunately, I could not find any thing important.



Availability of a firewall protection in the target domain

Wafw00f

Wafw00f is an in-built tool in Kali Linux that can use to discover firewall detection of the targeted website if there are any. This tool can analyze the server's responses that it got receive by using special web requests to the server.

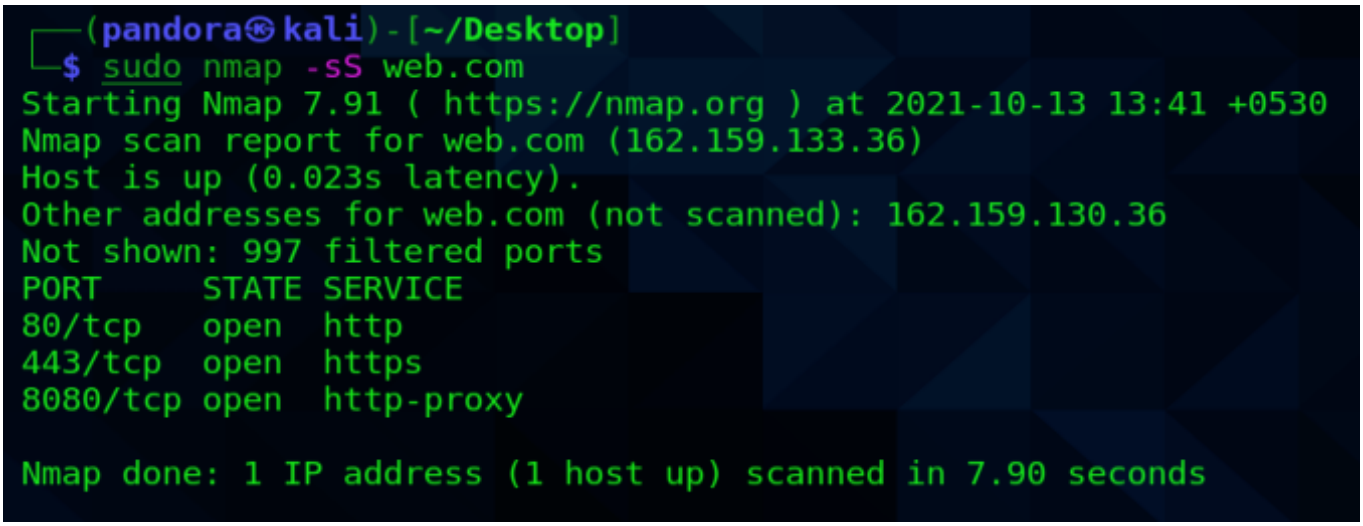
We can simply use this tool as follows to detect whether our targeted domain is protected behind a firewall.

Discover open ports

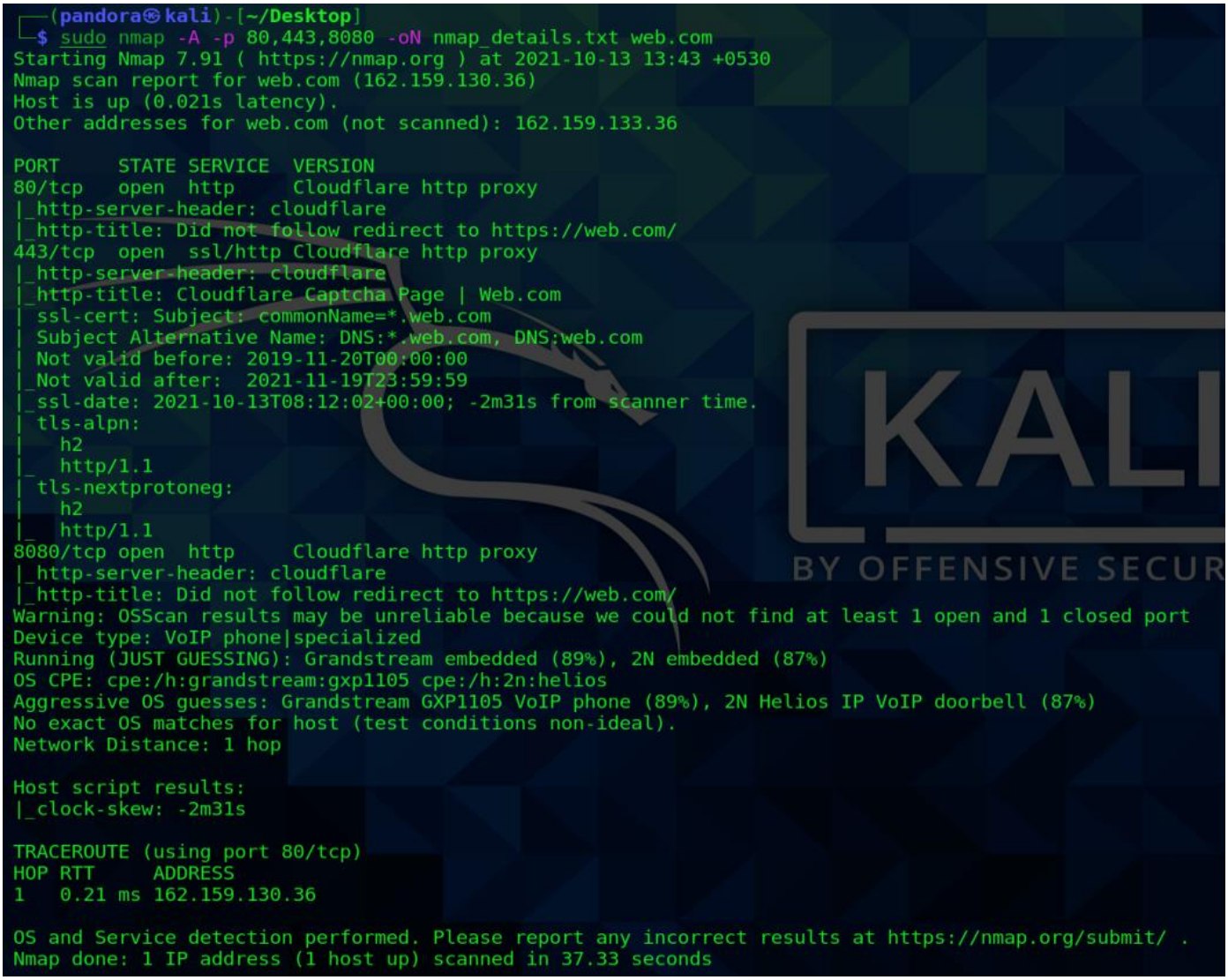
Nmap

Nmap is a great port scanning tool that can be used to discover open ports of our targeted website.

We can use this tool as follows to discover open ports.



We can see three open ports in the result, and now we can scan only those three open ports to seek some information about them. Moreover, I saved the result in a .txt file named “nmap_details.txt” for later usage. We can use the command as follows.


┌──(pandora㉿kali)-[~/Desktop]
└─$ sudo nmap -A -p 80,443,8080 -oN nmap_details.txt web.com
Starting Nmap 7.91 (https://nmap.org) at 2021-10-13 13:43 +0530
Nmap scan report for web.com (162.159.130.36)
Host is up (0.021s latency).
Other addresses for web.com (not scanned): 162.159.133.36
PORT STATE SERVICE VERSION
80/tcp open http Cloudflare http proxy
|_http-server-header: cloudflare
|_http-title: Did not follow redirect to https://web.com/
443/tcp open ssl/http Cloudflare http proxy
|_http-server-header: cloudflare
|_http-title: Cloudflare Captcha Page | Web.com
| ssl-cert: Subject: commonName=*.web.com
| Subject Alternative Name: DNS:*.web.com, DNS:web.com
| Not valid before: 2019-11-20T00:00:00
|_Not valid after: 2021-11-19T23:59:59
|_ssl-date: 2021-10-13T08:12:02+00:00; -2m31s from scanner time.
| tls-alpn:
| h2
|_ http/1.1
| tls-nextprotoneg:
| h2
|_ http/1.1
8080/tcp open http Cloudflare http proxy
|_http-server-header: cloudflare
|_http-title: Did not follow redirect to https://web.com/
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: VoIP phone|specialized
Running (JUST GUESSING): Grandstream embedded (89%), 2N embedded (87%)
OS CPE: cpe:/h:grandstream:gxp1105 cpe:/h:2n:helios
Aggressive OS guesses: Grandstream GXP1105 VoIP phone (89%), 2N Helios IP VoIP doorbell (87%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Host script results:
|_clock-skew: -2m31s
TRACEROUTE (using port 80/tcp)
HOP RTT ADDRESS
1 0.21 ms 162.159.130.36
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 37.33 seconds

03.Vulnerability Assessment

In this phase, we try to discover what are the possible vulnerabilities that we can find on our targeted website and categorize them by analyzing the risk of those vulnerabilities.

There are two type of vulnerability assessments.

1. Vulnerability assessment by using automated tools.
2. Manual vulnerability assessment.

Vulnerability assessment by using automated tools

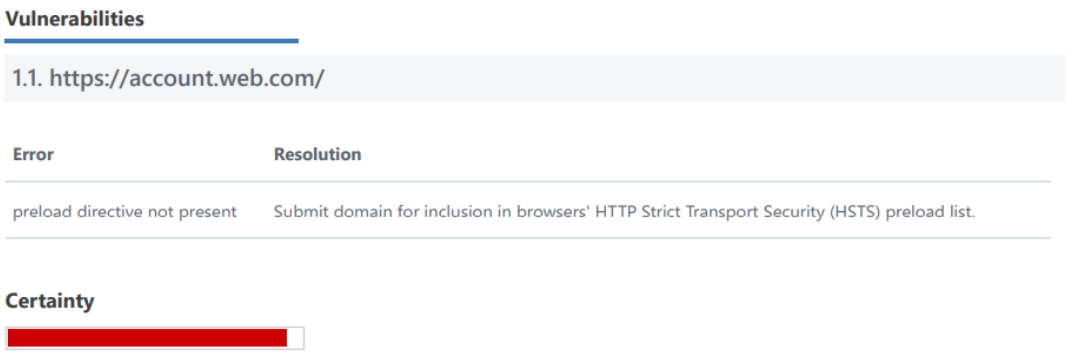
I used Netsparker Professional which is a wen application vulnerability scanner that can automatically detect almost every web application vulnerability.

1. Target Domain: http://account.web.com

i. HTTP Strict Transport Security (HSTS) Errors and Warnings

- Risk: MEDIUM
- Method: GET

Proof of Concept:



Impact:

The HSTS Warning and Error may allow attackers to bypass HSTS, effectively allowing them to read and modify your communication with the website.

Recommendation:

Remedy

Ideally, after fixing the errors and warnings, you should consider adding your domain to the the HSTS preload list. This will ensure that browsers automatically connect your website by using HTTPS, actively preventing users from visiting your site using HTTP. Since this list is hardcoded in users' browsers, it will enable HSTS even before they visit your page for the first time, eliminating the need for Trust On First Use (TOFU) with its associated risks and disadvantages. Unless you fix the errors and warnings your website won't meet the conditions required to enter the browser's preload list.

Browser vendors declared:

- Serve a valid certificate
- If you are listening on port 80, redirect all domains from HTTP to HTTPS on the same host. Serve all subdomains over HTTPS:
  - In particular, you must support HTTPS for the www subdomain if a DNS record for that subdomain exists
- Serve an HSTS header on the base domain for HTTPS requests:
  - The `max-age`must be at least 31536000 seconds (1 year)
  - The `includeSubDomains`directive must be specified
  - The `preload`directive must be specified
  - If you are serving an additional redirect from your HTTPS site, that redirect must have the HSTS header (rather than the page it redirects to)

ii. Weak Ciphers Enabled

- Risk: MEDIUM
- Method: GET

Proof of Concept:

Vulnerabilities

2.1. https://account.web.com/

CONFIRMED

List of Supported Weak Ciphers

- TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
- TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
- TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003C)
- TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003D)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC014)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC027)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xC028)

Impact:

Attackers might decrypt SSL traffic between your server and your visitors.

Recommendation:

Actions to Take

1. For Apache, you should modify the SSLCipherSuite directive in the `httpd.conf`.

```
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4
```

2. Lighttpd:

```
ssl.honor-cipher-order = "enable"
ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM"
```

3. For Microsoft IIS, you should make some changes to the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**

   a.Click Start, click Run, type `regedt32`or type `regedit`, and then click OK.
   b.In Registry Editor, locate the following registry key: `HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`
   c.Set "Enabled" DWORD to "0x0" for the following registry keys:

```
SCHANNEL\Ciphers\DES 56/56
SCHANNEL\Ciphers\RC4 64/128
SCHANNEL\Ciphers\RC4 40/128
SCHANNEL\Ciphers\RC2 56/128
SCHANNEL\Ciphers\RC2 40/128
SCHANNEL\Ciphers\NULL
SCHANNEL\Hashes\MD5
```

Remedy

Configure your web server to disallow using weak ciphers.

Summary:

http://account.web.com/

Scan Time	: 10/14/2021 1:54:06 AM (UTC+05:30)
Scan Duration	: 00:00:01:22
Total Requests	: 1,042
Average Speed	: 12.6r/s

Risk Level: MEDIUM

9 IDENTIFIED	2 CONFIRMED	0 CRITICAL
0 HIGH	2 MEDIUM	0 LOW
	4 BEST PRACTICE	3 INFORMATION

Identified Vulnerabilities

Critical	0
High	0
Medium	2
Low	0
Best Practice	4
Information	3
TOTAL	**9**

Confirmed Vulnerabilities

Critical	0
High	0
Medium	1
Low	0
Best Practice	0
Information	1
TOTAL	**2**

Vulnerability Summary

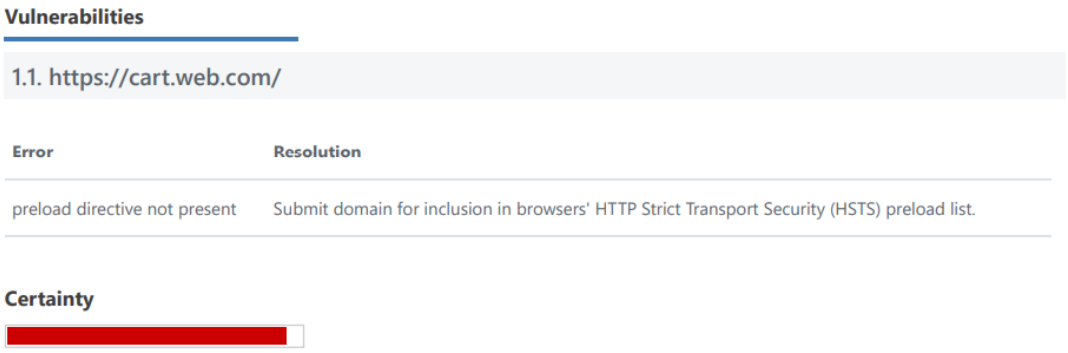
CONFIRM	VULNERABILITY	METHOD	URL	PARAMETER
	HTTP Strict Transport Security (HSTS) Errors and Warnings	GET	https://account.web.com/	
	Weak Ciphers Enabled	GET	https://account.web.com/	
	Content Security Policy (CSP) Not Implemented	GET	https://account.web.com/sitemap.xml	
	Missing X-XSS-Protection Header	GET	https://account.web.com/sitemap.xml	
	Referrer-Policy Not Implemented	GET	https://account.web.com/sitemap.xml	
	SameSite None Cookie Not Marked as Secure	GET	http://account.web.com/	
	Expect-CT in Report Only Mode	GET	https://account.web.com/sitemap.xml	
	HTTP Strict Transport Security (HSTS) Max-Age Value Too Low	GET	https://account.web.com/	
	Forbidden Resource	GET	https://account.web.com/.well-known/	

2. Target Domain: https://cart.web.com

1. HTTP Strict Transport Security (HSTS) Errors and Warnings

- Risk: MEDIUM
- Method: GET

Proof of Concept:



Vulnerabilities

1.1. https://cart.web.com/

Error	Resolution
preload directive not present	Submit domain for inclusion in browsers' HTTP Strict Transport Security (HSTS) preload list.

Certainty

Impact:

The HSTS Warning and Error may allow attackers to bypass HSTS, effectively allowing them to read and modify your communication with the website.

Recommendation:

Remedy

Ideally, after fixing the errors and warnings, you should consider adding your domain to the the HSTS preload list. This will ensure that browsers automatically connect your website by using HTTPS, actively preventing users from visiting your site using HTTP. Since this list is hardcoded in users' browsers, it will enable HSTS even before they visit your page for the first time, eliminating the need for Trust On First Use (TOFU) with its associated risks and disadvantages. Unless you fix the errors and warnings your website won't meet the conditions required to enter the browser's preload list.

Browser vendors declared:

- Serve a valid certificate
- If you are listening on port 80, redirect all domains from HTTP to HTTPS on the same host. Serve all subdomains over HTTPS:
  - In particular, you must support HTTPS for the www subdomain if a DNS record for that subdomain exists
- Serve an HSTS header on the base domain for HTTPS requests:
  - The `max-age`must be at least 31536000 seconds (1 year)
  - The `includeSubDomains`directive must be specified
  - The `preload`directive must be specified
  - If you are serving an additional redirect from your HTTPS site, that redirect must have the HSTS header (rather than the page it redirects to)

2. Weak Ciphers Enabled

- Risk:
- Method:

Proof of Concept:

Vulnerabilities

2.1. https://cart.web.com/

CONFIRMED

List of Supported Weak Ciphers

- TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
- TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
- TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003C)
- TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003D)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC014)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC027)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xC028)

Impact:

Attackers might decrypt SSL traffic between your server and your visitors.

Recommendation:

Actions to Take

1. For Apache, you should modify the SSLCipherSuite directive in the `httpd.conf`.

```
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4
```

2. Lighttpd:

```
ssl.honor-cipher-order = "enable"
ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM"
```

3. For Microsoft IIS, you should make some changes to the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**

a.Click Start, click Run, type `regedt32`or type `regedit`, and then click OK.
b.In Registry Editor, locate the following registry key: `HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`
c.Set "Enabled" DWORD to "0x0" for the following registry keys:

```
SCHANNEL\Ciphers\DES 56/56
SCHANNEL\Ciphers\RC4 64/128
SCHANNEL\Ciphers\RC4 40/128
SCHANNEL\Ciphers\RC2 56/128
SCHANNEL\Ciphers\RC2 40/128
SCHANNEL\Ciphers\NULL
SCHANNEL\Hashes\MD5
```

Remedy

Configure your web server to disallow using weak ciphers.

Summary:

https://cart.web.com/

Scan Time	: 10/14/2021 2:37:44 AM (UTC+05:30)
Scan Duration	: 00:00:02:00
Total Requests	: 829
Average Speed	: 6.9r/s

Risk Level: MEDIUM

9 IDENTIFIED	2 CONFIRMED	0 CRITICAL
0 HIGH	2 MEDIUM	0 LOW
	3 BEST PRACTICE	4 INFORMATION

Identified Vulnerabilities

Critical	0
High	0
Medium	2
Low	0
Best Practice	3
Information	4
TOTAL	**9**

Confirmed Vulnerabilities

Critical	0
High	0
Medium	1
Low	0
Best Practice	0
Information	1
TOTAL	**2**

Vulnerability Summary

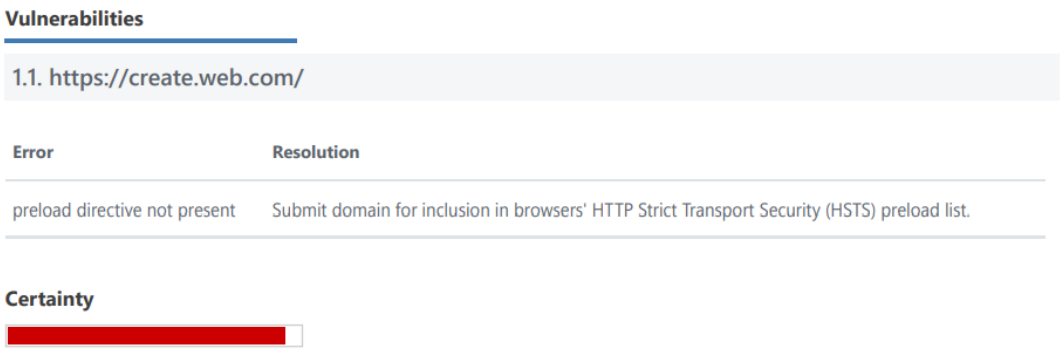
CONFIRM	VULNERABILITY	METHOD	URL	PARAMETER
	HTTP Strict Transport Security (HSTS) Errors and Warnings	GET	https://cart.web.com/	
	Weak Ciphers Enabled	GET	https://cart.web.com/	
	Content Security Policy (CSP) Not Implemented	GET	https://cart.web.com/	
	Missing X-XSS-Protection Header	GET	https://cart.web.com/	
	Referrer-Policy Not Implemented	GET	https://cart.web.com/	
	Expect-CT in Report Only Mode	GET	https://cart.web.com/	
	HTTP Strict Transport Security (HSTS) Max-Age Value Too Low	GET	https://cart.web.com/	
	Web Application Firewall Detected	GET	https://cart.web.com/%3Cscript%3Ealert(0)%3C/script%3E	URI-BASED
	Forbidden Resource	GET	https://cart.web.com/	

3. Target Domain: https://create.web.com

i. HTTP Strict Transport Security (HSTS) Errors and Warnings

- Risk: MEDIUM
- Method: GET

Proof of Concept:



Impact:

The HSTS Warning and Error may allow attackers to bypass HSTS, effectively allowing them to read and modify your communication with the website.

Recommendation:

Remedy

Ideally, after fixing the errors and warnings, you should consider adding your domain to the the HSTS preload list. This will ensure that browsers automatically connect your website by using HTTPS, actively preventing users from visiting your site using HTTP. Since this list is hardcoded in users' browsers, it will enable HSTS even before they visit your page for the first time, eliminating the need for Trust On First Use (TOFU) with its associated risks and disadvantages. Unless you fix the errors and warnings your website won't meet the conditions required to enter the browser's preload list.

Browser vendors declared:

- Serve a valid certificate
- If you are listening on port 80, redirect all domains from HTTP to HTTPS on the same host. Serve all subdomains over HTTPS:
  - In particular, you must support HTTPS for the www subdomain if a DNS record for that subdomain exists
- Serve an HSTS header on the base domain for HTTPS requests:
  - The `max-age`must be at least 31536000 seconds (1 year)
  - The `includeSubDomains`directive must be specified
  - The `preload`directive must be specified
  - If you are serving an additional redirect from your HTTPS site, that redirect must have the HSTS header (rather than the page it redirects to)

ii. Weak Ciphers Enabled

- Risk: MEDIUM
- Method: GET

Proof of Concept:

Vulnerabilities

2.1. https://create.web.com/

CONFIRMED

List of Supported Weak Ciphers

- TLS_RSA_WITH_RC4_128_SHA (0x0005)
- TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000A)
- TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
- TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC014)
- TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003C)
- TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003D)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC027)

Impact:

Attackers might decrypt SSL traffic between your server and your visitors.

Recommendation:

Actions to Take

1. For Apache, you should modify the SSLCipherSuite directive in the `httpd.conf`.

```
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4
```

2. Lighttpd:

```
ssl.honor-cipher-order = "enable"
ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM"
```

3. For Microsoft IIS, you should make some changes to the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**

a.Click Start, click Run, type `regedt32`or type `regedit`, and then click OK.
b.In Registry Editor, locate the following registry key: `HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`
c.Set "Enabled" DWORD to "0x0" for the following registry keys:

```
SCHANNEL\Ciphers\DES 56/56
SCHANNEL\Ciphers\RC4 64/128
SCHANNEL\Ciphers\RC4 40/128
SCHANNEL\Ciphers\RC2 56/128
SCHANNEL\Ciphers\RC2 40/128
SCHANNEL\Ciphers\NULL
SCHANNEL\Hashes\MD5
```

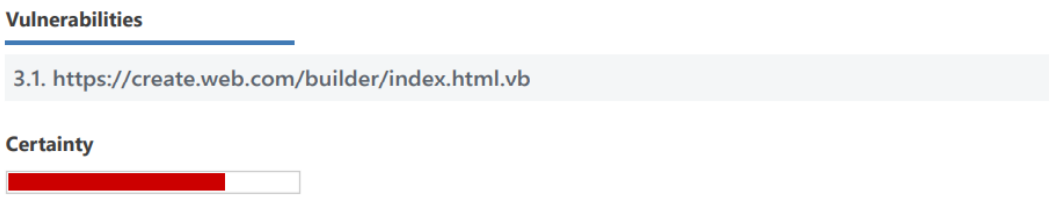
Remedy

Configure your web server to disallow using weak ciphers.

iii. [Possible] Backup File Disclosure

- Risk: LOW
- Method: GET

Proof of Concept:



Impact:

Backup files can contain old or current versions of a file on the web server. This could include sensitive data such as password files or even the application's source code. This form of issue normally leads to further vulnerabilities or, at worst, sensitive information disclosure.

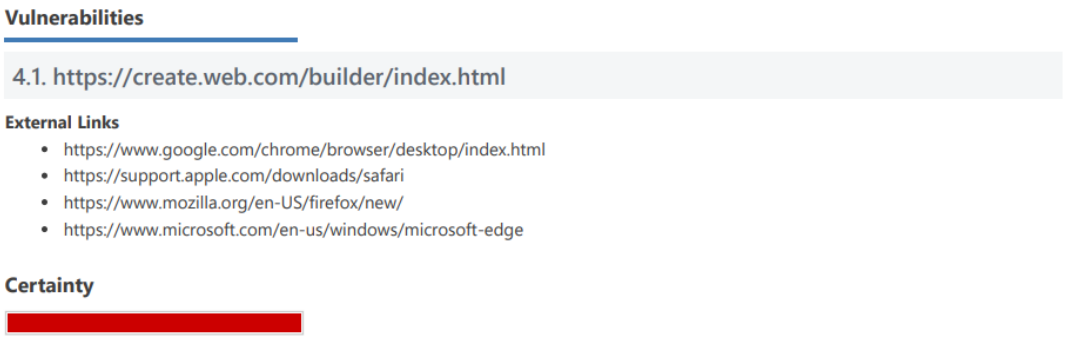
Recommendation:

Do not store backup files on production servers.

iv. [Possible] Phishing by Navigating Browser Tabs

- Risk: LOW
- Method: GET

Proof of Concept:



Impact:

While this vulnerability doesn't allow script execution, it does allow phishing attacks that silently replace the parent tab. If the links lack rel="noopener noreferrer"attribute, a third party site can change the URL of the source tab using window.opener.location.assignand trick the users into thinking that they're still in a trusted page and lead them to enter their sensitive data on the malicious website.

Recommendation:

Remedy

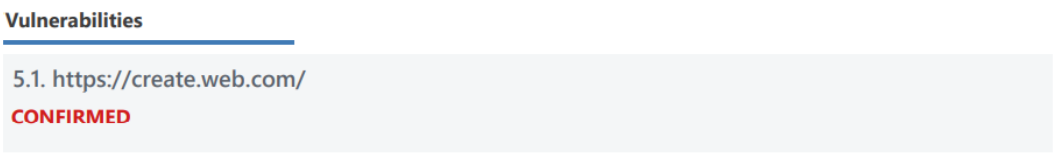
- Add `rel=noopener`to the linksto prevent pages from abusing *window.opener*. This ensures that the page cannot access the *window.opener*property in Chrome and Opera browsers.
- For older browsers and in Firefox, you can add `rel=noreferrer`which additionally disables the Referer header.

```
<a href="..." target="_blank" rel="noopener noreferrer">...</a>
```

v. Insecure Transportation Security Protocol Supported (TLS 1.0)

- Risk: LOW
- Method: GET

Proof of Concept:



Impact:

Attackers can perform man-in-the-middle attacks and observe the encryption traffic between your website and its visitors.

Recommendation:

Actions to Take

We recommended to disable TLS 1.0 and replace it with TLS 1.2 or higher. See Remedy section for more details.

Remedy

Configure your web server to disallow using weak ciphers. You need to restart the web server to enable changes.

- For Apache, adjust the SSLProtocol directive provided by the mod_ssl module. This directive can be set either at the server level or in a virtual host configuration.

```
SSLProtocol +TLSv1.2
```

- For Nginx, locate any use of the directive ssl_protocols in the `nginx.conf`file and remove `TLSv1`.

```
ssl_protocols TLSv1.2;
```

- For Microsoft IIS, you should make some changes on the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**
  1. Click on Start and then Run, type `regedt32`or `regedit`, and then click OK.
  2. In Registry Editor, locate the following registry key or create if it does not exist:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.0\
```

3. Locate a key named `Server`or create if it doesn't exist.
4. Under the `Server`key, locate a DWORD value named `Enabled`or create if it doesn't exist and set its value to "0".

- For lighttpd, put the following lines in your configuration file:

```
ssl.use-sslv2 = "disable"
ssl.use-sslv3 = "disable"
ssl.openssl.ssl-conf-cmd = ("Protocol" => "-TLSv1.1, -TLSv1, -SSLv3") # v1.4.48 or up
ssl.ec-curve = "secp384r1"
```

vi. Internal Server Error

- Risk: LOW
- Method: GET

Proof of Concept:

Vulnerabilities

6.1. https://create.web.com/builder/index.htmlc:/
CONFIRMED

Impact:

The impact may vary depending on the condition. Generally this indicates poor coding practices, not enough error checking, sanitization and whitelisting. However, there might be a bigger issue, such as SQL injection. If that's the case, Netsparker will check for other possible issues and report them separately.

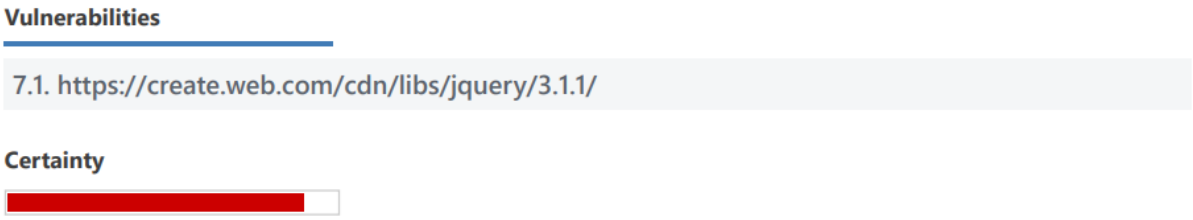
Recommendation:

Analyze this issue and review the application code in order to handle unexpected errors; this should be a generic practice, which does not disclose further information upon an error. All errors should be handled server-side only.

vii. Missing X-Frame-Options Header

- Risk: LOW
- Method: GET

Proof of Concept:



Impact:

Clickjacking is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on a framed page when they were intending to click on the top level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to other another page, most likely owned by another application, domain, or both. Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

Recommendation:

Remedy

- Sending the proper X-Frame-Options in HTTP response headers that instruct the browser to not allow framing from other domains.
  - `X-Frame-Options: DENY` It completely denies to be loaded in frame/iframe.
  - `X-Frame-Options: SAMEORIGIN` It allows only if the site which wants to load has a same origin.
  - `X-Frame-Options: ALLOW-FROM` *URL* It grants a specific URL to load itself in a iframe. However please pay attention to that, not all browsers support this.
- Employing defensive code in the UI to ensure that the current frame is the most top level window.

Summary:

https://create.web.com/

Scan Time	: 10/14/2021 2:44:20 AM (UTC+05:30)
Scan Duration	: 00:00:20:13
Total Requests	: 9,677
Average Speed	: 8.0r/s

Risk Level: MEDIUM

17 IDENTIFIED	6 CONFIRMED	0 CRITICAL
0 HIGH	2 MEDIUM	5 LOW
	6 BEST PRACTICE	4 INFORMATION

Identified Vulnerabilities

Critical	0
High	0
Medium	2
Low	5
Best Practice	6
Information	4
TOTAL	**17**

Confirmed Vulnerabilities

Critical	0
High	0
Medium	1
Low	2
Best Practice	1
Information	2
TOTAL	**6**

Vulnerability Summary

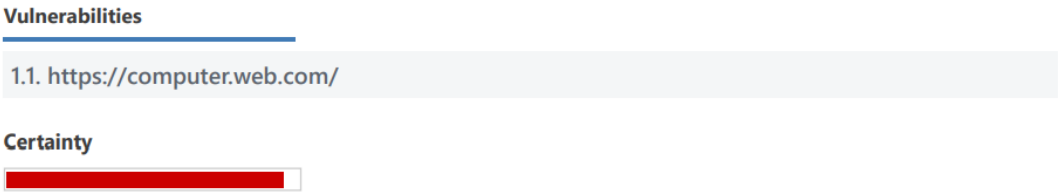
CONFIRM	VULNERABILITY	METHOD	URL	PARAMETER
	HTTP Strict Transport Security (HSTS) Errors and Warnings	GET	https://create.web.com/	
	Weak Ciphers Enabled	GET	https://create.web.com/	
	[Possible] Backup File Disclosure	GET	https://create.web.com/builder/index.html.vb	
	[Possible] Phishing by Navigating Browser Tabs	GET	https://create.web.com/builder/index.html	
	Missing X-Frame-Options Header	GET	https://create.web.com/cdn/libs/jquery/3.1.1/	
	Insecure Transportation Security Protocol Supported (TLS 1.0)	GET	https://create.web.com/	
	Internal Server Error	GET	https://create.web.com/builder/index.htmlc:/	
	Content Security Policy (CSP) Not Implemented	GET	https://create.web.com/builder/	
	Expect-CT Not Enabled	GET	https://create.web.com/	
	Missing X-XSS-Protection Header	GET	https://create.web.com/cdn/libs/jquery/3.1.1/	
	Referrer-Policy Not Implemented	GET	https://create.web.com/cdn/libs/jquery/3.1.1/	
	Subresource Integrity (SRI) Not Implemented	GET	https://create.web.com/builder/index.html	
	Insecure Transportation Security Protocol Supported (TLS 1.1)	GET	https://create.web.com/	
	Email Address Disclosure	GET	https://create.web.com/builder/app.2_0_0.7610.min.js	
	Nginx Web Server Identified	GET	https://create.web.com/	

4. Target Domain: https://computer.web.com

i. HTTP Strict Transport Security (HSTS) Policy Not Enabled

- Risk: MEDIUM
- Method: GET

Proof of concept:



Recommendation:

Remedy

Configure your webserver to redirect HTTP requests to HTTPS.

i.e. for Apache, you should have modification in the httpd.conf. For more configurations, please refer to External References section.

```
# load module
LoadModule headers_module modules/mod_headers.so

# redirect all HTTP to HTTPS (optional)
<VirtualHost *:80>
       ServerAlias *
       RewriteEngine On
       RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [redirect=301]
</VirtualHost>

# HTTPS-Host-Configuration
<VirtualHost *:443>
      # Use HTTP Strict Transport Security to force client to use secure connections only
      Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"
```

```
      # Further Configuration goes here
      [...]
</VirtualHost>
```

ii. Weak Ciphers Enabled

- Risk: MEDIUM
- Method: GET

Proof of concept:

Vulnerabilities

2.1. https://computer.web.com/

CONFIRMED

List of Supported Weak Ciphers

- TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000A)
- TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
- TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
- TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (0xC012)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC014)

Impact:

Attackers might decrypt SSL traffic between your server and your visitors.

Recommendation:

Actions to Take

1. For Apache, you should modify the SSLCipherSuite directive in the `httpd.conf`.

```
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4
```

2. Lighttpd:

```
ssl.honor-cipher-order = "enable"
ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM"
```

3. For Microsoft IIS, you should make some changes to the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**

a.Click Start, click Run, type `regedt32`or type `regedit`, and then click OK.
b.In Registry Editor, locate the following registry key: `HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`
c.Set "Enabled" DWORD to "0x0" for the following registry keys:

```
SCHANNEL\Ciphers\DES 56/56
```

```
SCHANNEL\Ciphers\RC4 64/128
SCHANNEL\Ciphers\RC4 40/128
SCHANNEL\Ciphers\RC2 56/128
SCHANNEL\Ciphers\RC2 40/128
SCHANNEL\Ciphers\NULL
SCHANNEL\Hashes\MD5
```

Remedy

Configure your web server to disallow using weak ciphers.

iii. Insecure Transportation Security Protocol Supported (TLS 1.0)

- Risk: LOW
- Method: GET

Proof of concept:

Vulnerabilities

3.1. https://computer.web.com/

CONFIRMED

Impact:

Attackers can perform man-in-the-middle attacks and observe the encryption traffic between your website and its visitors.

Recommendation:

Actions to Take

We recommended to disable TLS 1.0 and replace it with TLS 1.2 or higher. See Remedy section for more details.

Remedy

Configure your web server to disallow using weak ciphers. You need to restart the web server to enable changes.

- For Apache, adjust the SSLProtocol directive provided by the mod_ssl module. This directive can be set either at the server level or in a virtual host configuration.

```
SSLProtocol +TLSv1.2
```

- For Nginx, locate any use of the directive ssl_protocols in the `nginx.conf`file and remove `TLSv1`.

```
ssl_protocols TLSv1.2;
```

- For Microsoft IIS, you should make some changes on the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**
    1. Click on Start and then Run, type `regedt32`or `regedit`, and then click OK.
    2. In Registry Editor, locate the following registry key or create if it does not exist:

    3. Locate a key named `Server`or create if it doesn't exist.
    4. Under the `Server`key, locate a DWORD value named `Enabled`or create if it doesn't exist and set its value to "0".
- For lighttpd, put the following lines in your configuration file:

```
ssl.use-sslv2 = "disable"
ssl.use-sslv3 = "disable"
ssl.openssl.ssl-conf-cmd = ("Protocol" => "-TLSv1.1, -TLSv1, -SSLv3") # v1.4.48 or up
ssl.ec-curve = "secp384r1"
```

iv. Missing X-Frame-Options Header

- Risk: LOW
- Method: GET

Proof of concept:

Vulnerabilities

4.1. https://computer.web.com/

Certainty

Impact:

Clickjacking is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on a framed page when they were intending to click on the top level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to other another page, most likely owned by another application, domain, or both. Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

Recommendation:

Remedy

- Sending the proper X-Frame-Options in HTTP response headers that instruct the browser to not allow framing from other domains.
  - `X-Frame-Options: DENY`It completely denies to be loaded in frame/iframe.
  - `X-Frame-Options: SAMEORIGIN`It allows only if the site which wants to load has a same origin.
  - `X-Frame-Options: ALLOW-FROM` *URL*It grants a specific URL to load itself in a iframe. However please pay attention to that, not all browsers support this.
- Employing defensive code in the UI to ensure that the current frame is the most top level window.

Summary:

https://computer.web.com/

Scan Time	: 10/14/2021 3:14:31 AM (UTC+05:30)
Scan Duration	: 00:00:03:26
Total Requests	: 639
Average Speed	: 3.1r/s

Risk Level: **MEDIUM**

11 IDENTIFIED	4 CONFIRMED	0 CRITICAL
0 HIGH	2 MEDIUM	2 LOW
	5 BEST PRACTICE	2 INFORMATION

Identified Vulnerabilities

Critical	0
High	0
Medium	2
Low	2
Best Practice	5
Information	2
TOTAL	**11**

Confirmed Vulnerabilities

Critical	0
High	0
Medium	1
Low	1
Best Practice	1
Information	1
TOTAL	**4**

Vulnerability Summary

CONFIRM	VULNERABILITY	METHOD	URL	PARAMETER
	HTTP Strict Transport Security (HSTS) Policy Not Enabled	GET	https://computer.web.com/	
	Weak Ciphers Enabled	GET	https://computer.web.com/	
	Missing X-Frame-Options Header	GET	https://computer.web.com/	
	Insecure Transportation Security Protocol Supported (TLS 1.0)	GET	https://computer.web.com/	
	Content Security Policy (CSP) Not Implemented	GET	https://computer.web.com/	
	Expect-CT Not Enabled	GET	https://computer.web.com/	
	Missing X-XSS-Protection Header	GET	https://computer.web.com/	
	Referrer-Policy Not Implemented	GET	https://computer.web.com/	
	Insecure Transportation Security Protocol Supported (TLS 1.1)	GET	https://computer.web.com/	
	Nginx Web Server Identified	GET	https://computer.web.com/	
	Forbidden Resource	GET	https://computer.web.com/	

Manual vulnerability assessment

Target Domain: https://www.web.com

Cross Site Scripting vulnerability (XSS)

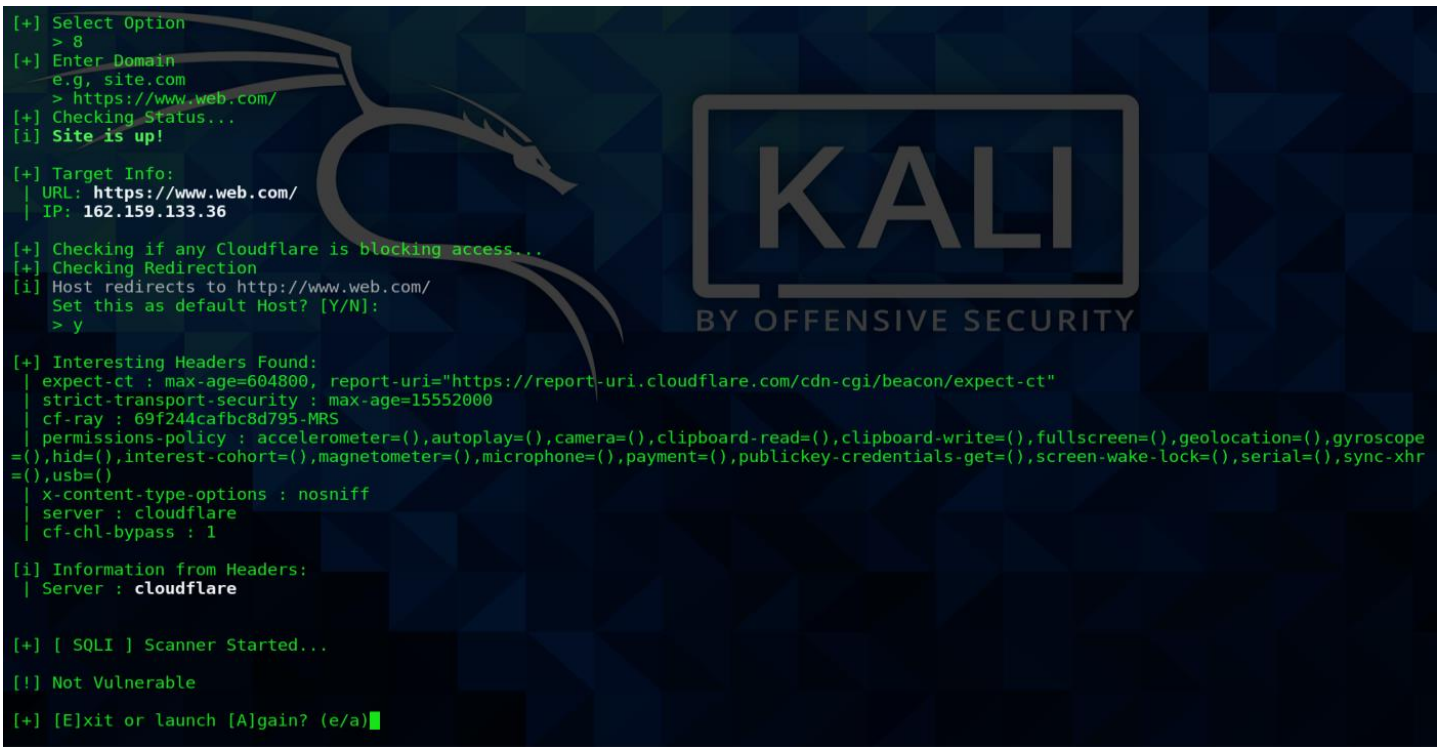
D-Tech

D-Tech is a vulnerability tool that is designed by using Python language that can discover the following vulnerabilities. We can input the number of the vulnerability according to the list we want to scan on our targeted website, and we need to give the targeted domain that we wish to scan.

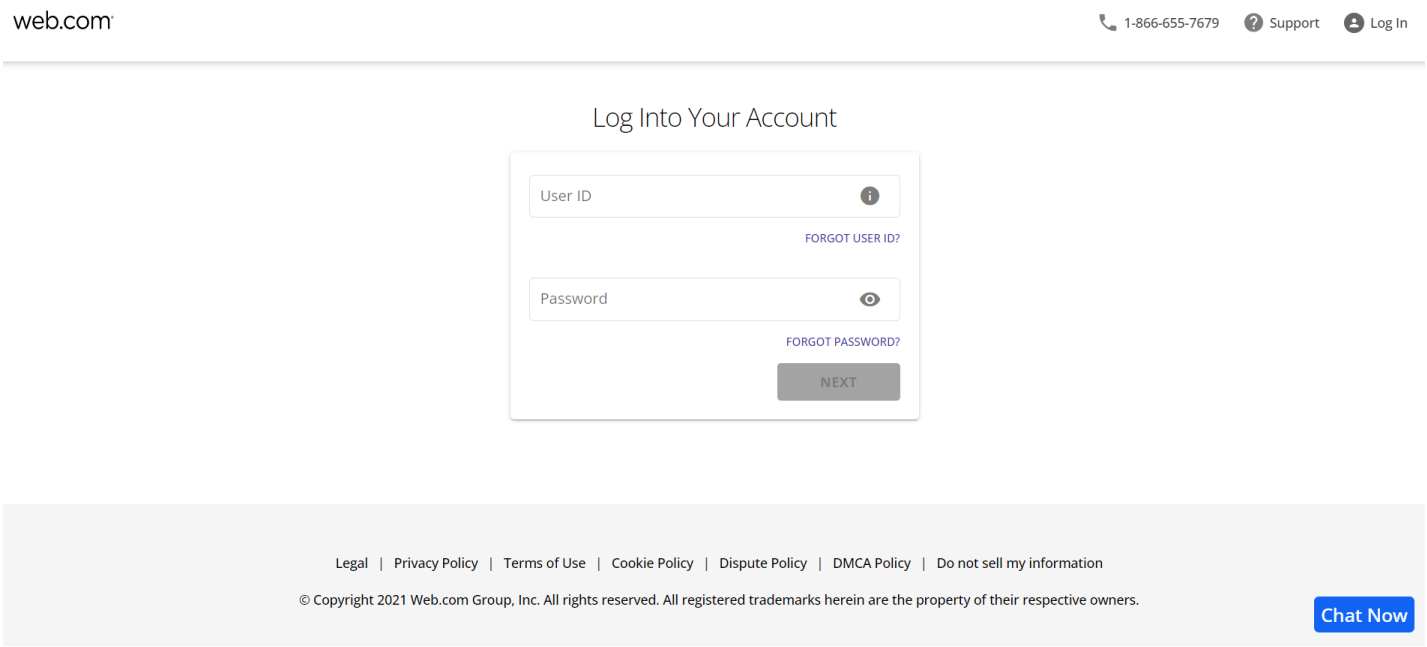
Result: Target is not vulnerable to XSS attacks.

SQL Injection vulnerability

I can again use D-Tech to try to discover SQL Injection vulnerability on my targeted website.



It shows this target is not vulnerable for SQL Injection attacks but we can also check it manually as follows.



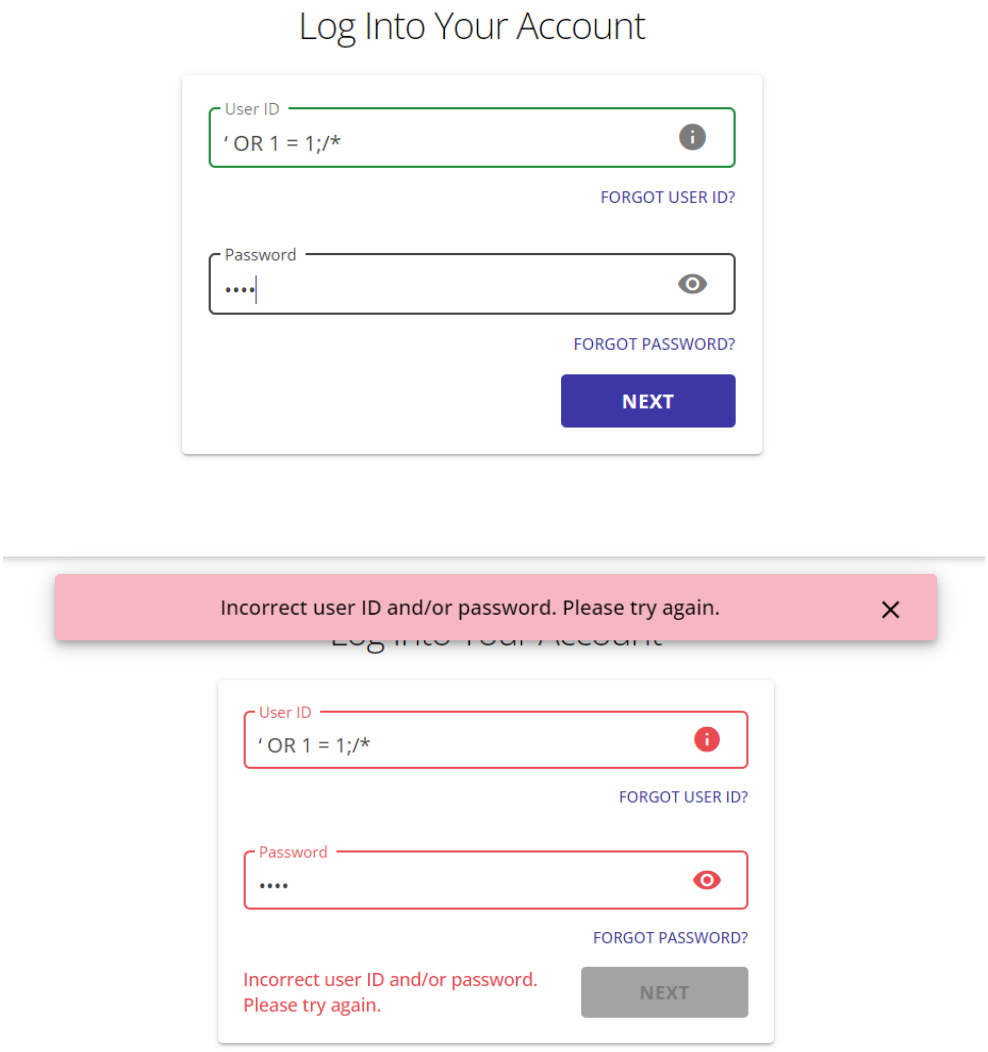
This is the login page of the web.com website. Now I am going to try to send an input as follows.

The User ID: **‘ OR 1 = 1;/***
The Password: ***/--**

If there is not any security measures to stop these kind of attacks, the SELECT query will be like as follows.

SELECT * FROM users
WHERE user_id = ‘ OR 1 = 1;/*’ AND password = ‘*/--’

The result as follows.



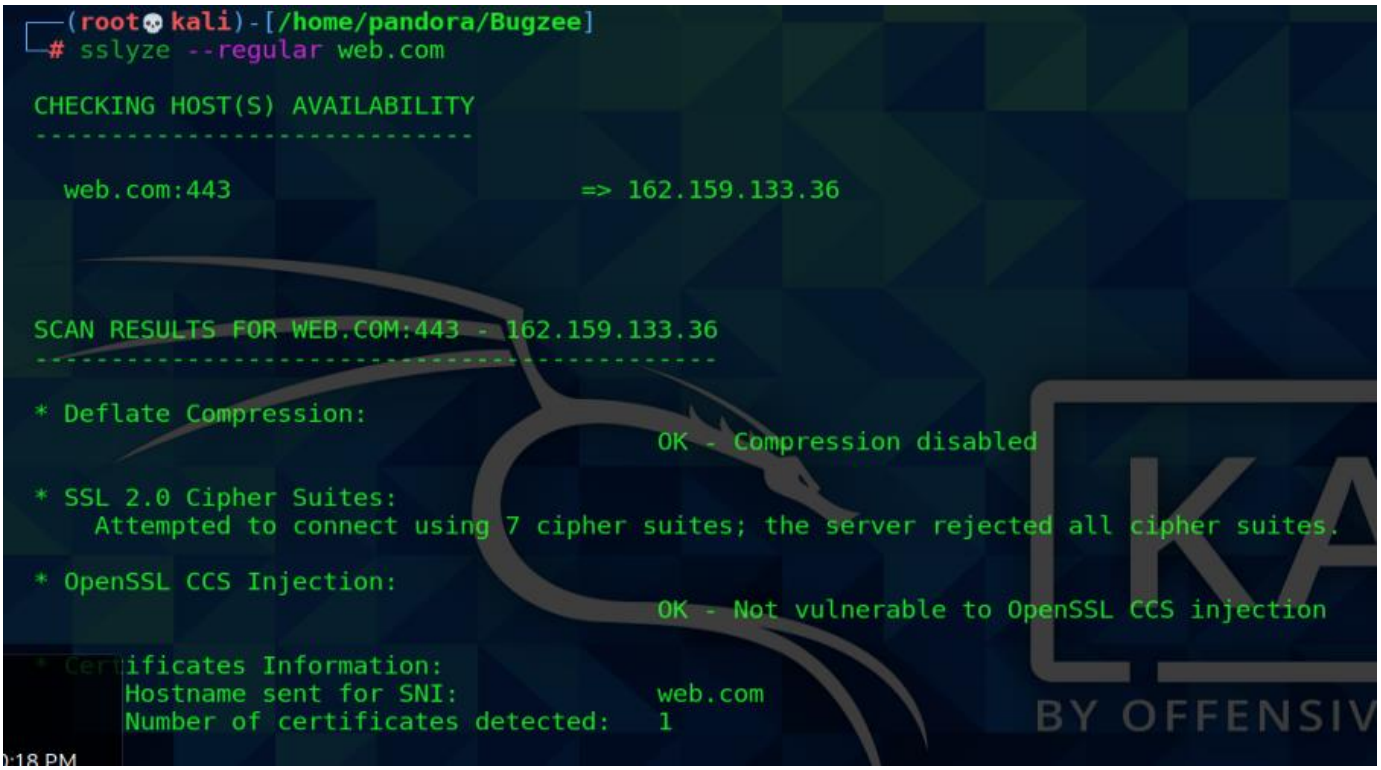
Result: Target is not vulnerable to SQL Injection attacks.

Cipher strength of the target domain

SSLyze

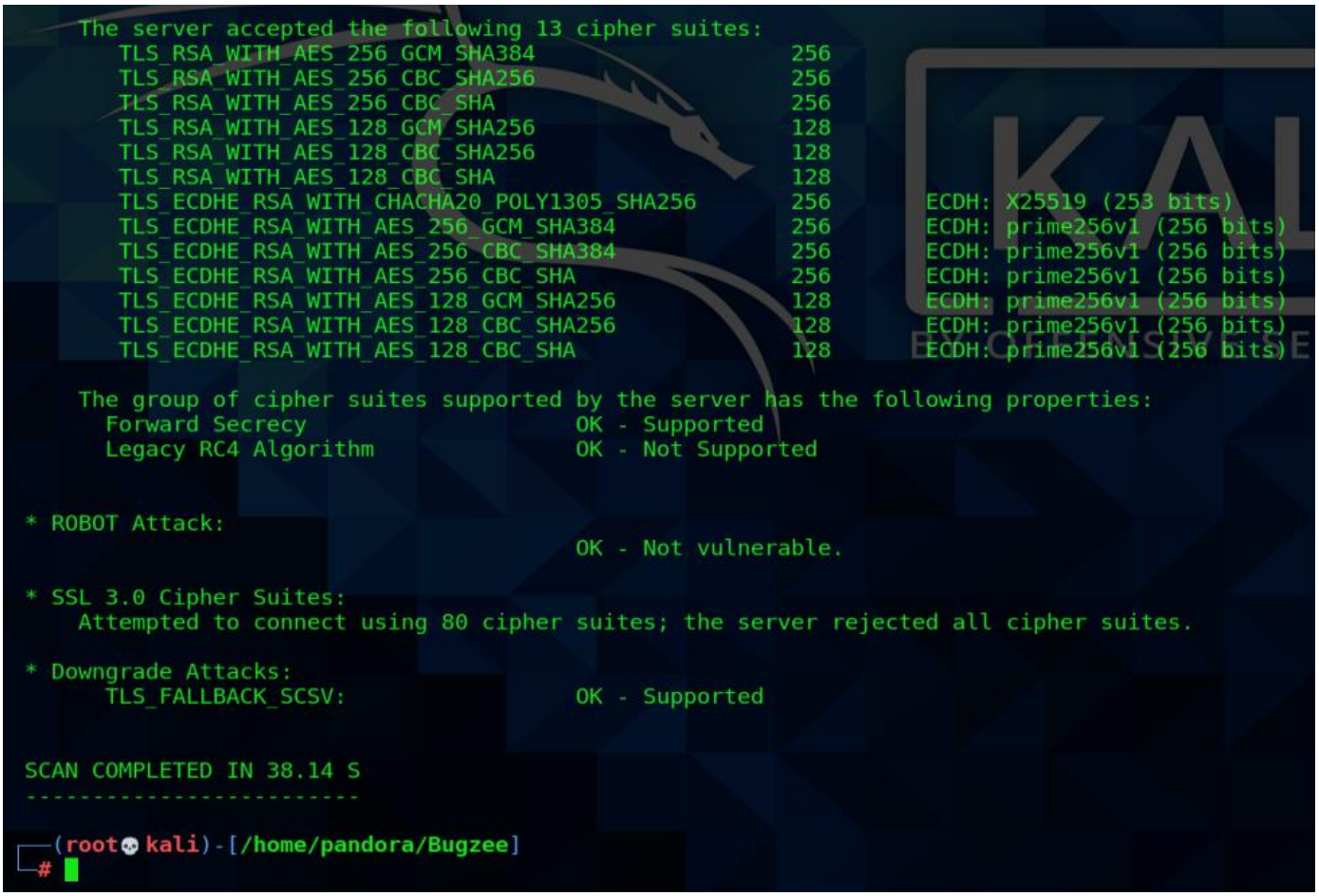
We can use SSLyze to identify SSL/TLS configuration of a server and we can identify problems like, weak cipher suits, bad certificates very easily.

The usage of the SSLyze as follows.



```
    Certificate #0 ( _RSAPublicKey )
      SHA1 Fingerprint:                  52b4df1e4dad2ef10751f137cfbf6180cbb77269
      Common Name:                       *.web.com
      Issuer:                            Sectigo RSA Domain Validation Secure Server CA
      Serial Number:                     167269307541966867734731797005145316303
      Not Before:                        2019-11-20
      Not After:                         2021-11-19
      Public Key Algorithm:              _RSAPublicKey
      Signature Algorithm:               sha256
      Key Size:                          2048
      Exponent:                          65537
      DNS Subject Alternative Names:     ['*.web.com', 'web.com']

    Certificate #0 - Trust
      Hostname Validation:               OK - Certificate matches server hostname
      Android CA Store (9.0.0_r9):       OK - Certificate is trusted
      Apple CA Store (iOS 14, iPadOS 14, macOS 11, watchOS 7, and tvOS 14):OK - Certificate is trusted
      Java CA Store (jdk-13.0.2):        OK - Certificate is trusted
      Mozilla CA Store (2021-01-24):     OK - Certificate is trusted
      Windows CA Store (2021-02-08):     OK - Certificate is trusted
      Symantec 2018 Deprecation:         OK - Not a Symantec-issued certificate
      Received Chain:                    *.web.com --> Sectigo RSA Domain Validation Secure Server CA --> USERTrust RSA Certification Aut
hority
      Verified Chain:                    *.web.com --> Sectigo RSA Domain Validation Secure Server CA --> USERTrust RSA Certification Aut
hority
      Received Chain Contains Anchor:    OK - Anchor certificate not sent
      Received Chain Order:              OK - Order is valid
      Verified Chain contains SHA1:      OK - No SHA1-signed certificate in the verified certificate chain

    Certificate #0 - Extensions
      OCSP Must-Staple:                  NOT SUPPORTED - Extension not found
      Certificate Transparency:          OK - 3 SCTs included

    Certificate #0 - OCSP Stapling
      OCSP Response Status:              SUCCESSFUL
      Validation w/ Mozilla Store:       OK - Response is trusted
      Responder Key Hash:                b'\x8d\x8c^\xc4T\xad\x8a\xe1w\xe9\x9b\xf9\x9b\x05\xe1\xb8\x01\x8da\xe1'
      Cert Status:                       GOOD
      Cert Serial Number:                167269307541966867734731797005145316303
      This Update:                       2021-10-13
      Next Update:                       2021-10-20
```

```
      Next Update:                       2021-10-20

 * TLS 1.3 Cipher Suites:
    Attempted to connect using 5 cipher suites.

    The server accepted the following 3 cipher suites:
       TLS_CHACHA20_POLY1305_SHA256                    256       ECDH: X25519 (253 bits)
       TLS_AES_256_GCM_SHA384                          256       ECDH: X25519 (253 bits)
       TLS_AES_128_GCM_SHA256                          128       ECDH: X25519 (253 bits)


 * OpenSSL Heartbleed:
                                      OK - Not vulnerable to Heartbleed

 * Session Renegotiation:
      Client Renegotiation DoS Attack:   OK - Not vulnerable
      Secure Renegotiation:              OK - Supported

 * TLS 1.0 Cipher Suites:
    Attempted to connect using 80 cipher suites; the server rejected all cipher suites.

 * TLS 1.2 Session Resumption Support:
     With Session IDs: OK - Supported (5 successful resumptions out of 5 attempts).
     With TLS Tickets: OK - Supported.

 * Elliptic Curve Key Exchange:
      Supported curves:                  X25519, prime256v1, secp384r1, secp521r1
      Rejected curves:                   X448, prime192v1, secp160k1, secp160r1, secp160r2, secp192k1, secp224k1, secp224r1, secp256k1, s
ect163k1, sect163r1, sect163r2, sect193r1, sect193r2, sect233k1, sect233r1, sect239k1, sect283k1, sect283r1, sect409k1, sect409r1, sect571
k1, sect571r1

 * TLS 1.1 Cipher Suites:
    Attempted to connect using 80 cipher suites; the server rejected all cipher suites.

 * TLS 1.2 Cipher Suites:
    Attempted to connect using 156 cipher suites.
```

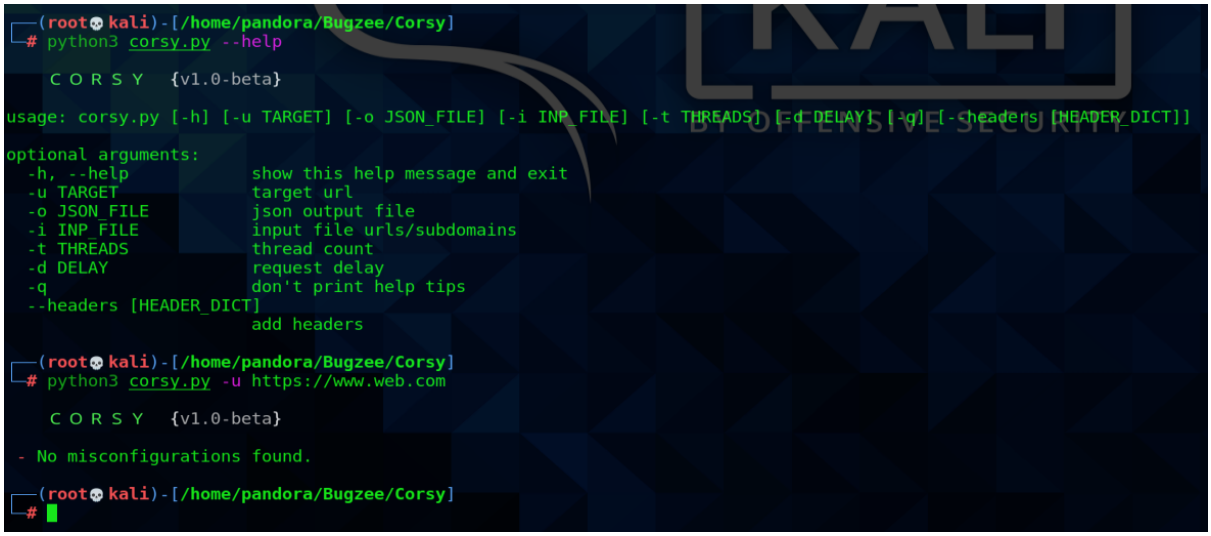


Result: Target has no critical cipher related issues.

CORS Misconfiguration

Cross-Origin Resource Sharing (CORS) is a method we can use to enable web browsers to do cross-domain requests using the XMLHttpRequest API with proper control. Mainly, this will allow access from other website resources such as subdomains and trusted third parties. Sensitive information may be disclosed to attackers by exploiting vulnerabilities related to CORS.

We can identify these kinds of attacks by using this Python language-based tool called "Corsy." The usage of this tool is as follows.

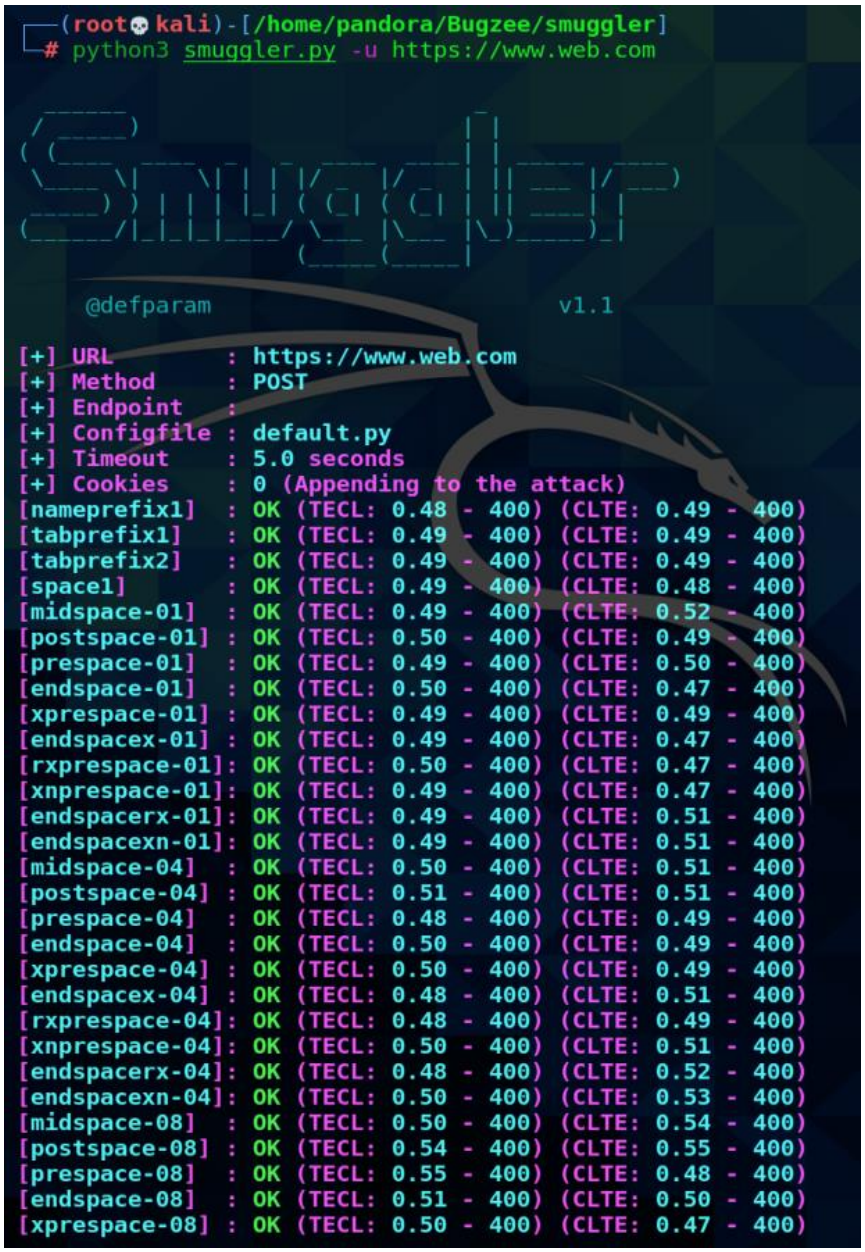


Result: Target is not vulnerable for CORS misconfiguration attacks.

HTTPS Request Smuggling

A smuggler is a tool designed by using Python language to identify the possibility of HTTPS request smuggling.

The usage and the result are as follows. If all requests are shown ok, there is no HTTP request smuggling vulnerability in our targeted website.



Result: Domain is not vulnerable to HTTPS request smuggling.

Conclusion

his report is about the vulnerability assessment of the https://www.web.com, which has been done according to the proper methodology. Throughout this vulnerability assessment, I used both automated and manual testing tools and methods. And also. I well explained every tool that was used to complete this web audit.

References

1. https://bugcrowd.com/webdotcom-vdp
2. https://www.trustnetinc.com/vulnerability-assessment/
3. https://securitytrails.com/blog/information-gathering
4. https://github.com/aboul3la/Sublist3r
5. https://github.com/tomnomnom/httprobe
6. https://github.com/laramies/theHarvester
7. http://archive.org/web/
8. https://builtwith.com
9. https://github.com/AlexisAhmed/BugBountyToolkit
10. https://www.shodan.io/
11. https://owasp.org/www-community/attacks/xss/
12. https://www.netsparker.com
13. https://thehackerish.com/bug-bounty-tools-from-enumeration-to-reporting/