medium.com

Recon Methodology For Bug Bounty Hunting

Md. Raihan

9–11 minutes

6 min read

Oct 17, 2025

Press enter or click to view image in full size

What is Reconnaissance:

The recon phase is all about gathering information. The more data you have on your target, the higher your

chances of finding vulnerabilities. This stage sets the foundation for everything that follows.

Subdomain Enumeration

Subdomains often hide forgotten features, admin panels, or unpatched vulnerabilities. Use multiple tools for better coverage.

Passive Subdomain Enumeration

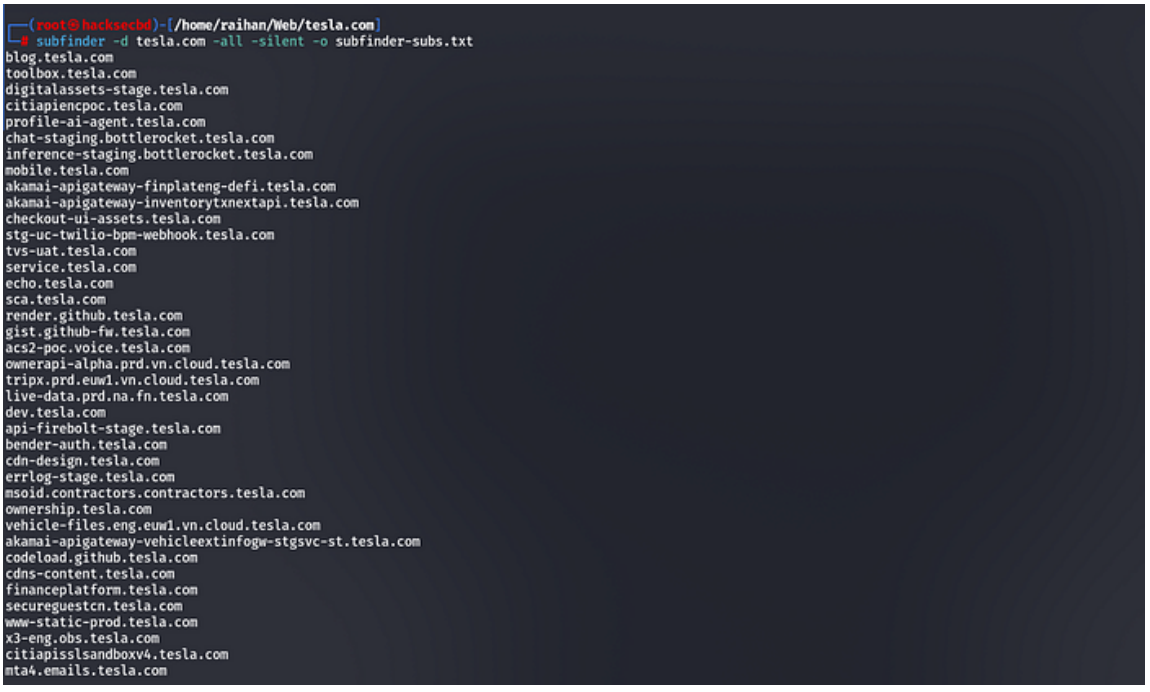
Passive enumeration relies on third-party sources and APIs, such as certificate transparency logs, search engines, and public DNS databases.

- **Tools**: `Subfinder`,`Assetfinder`, `Amass`, `Sublist3r`, `Findomain`

Subfinder Example Command :

```
subfinder -d target.com -all -silent -o subfinder-subs.txt
```

Press enter or click to view image in full size



Assetfinder Example Command :

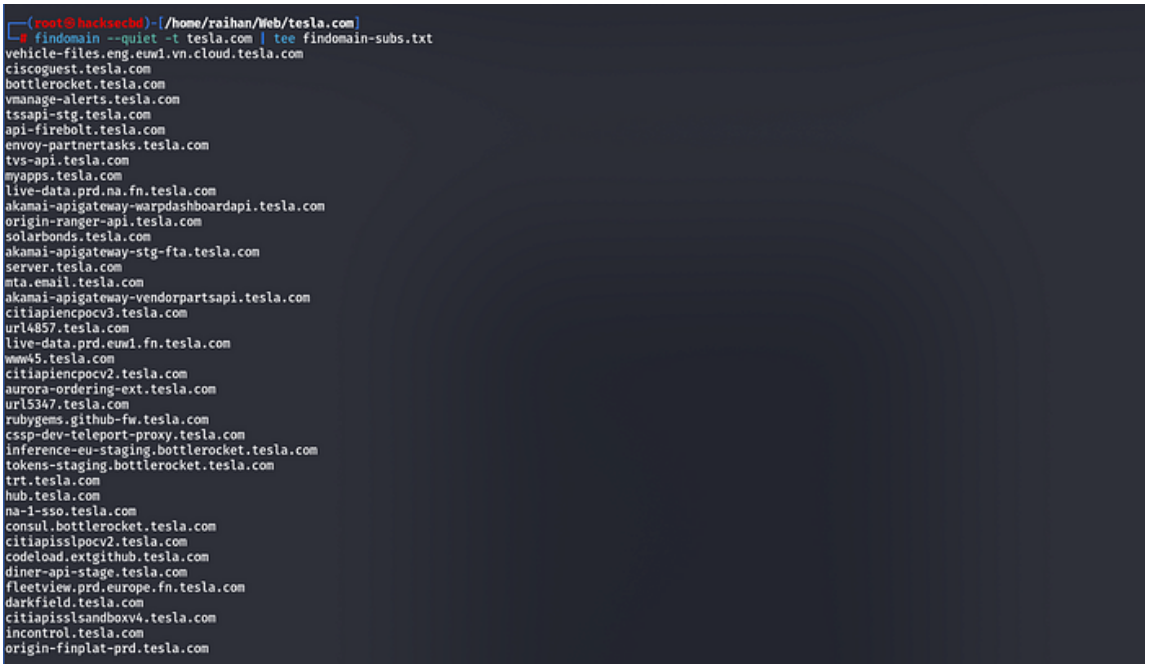
```
assetfinder -subs-only target.com | tee
assetfinder_subs.txt
```

Press enter or click to view image in full size

Findomain Example Command:

```
findomain --quiet -t target.com | tee
findomain-subs.txt
```

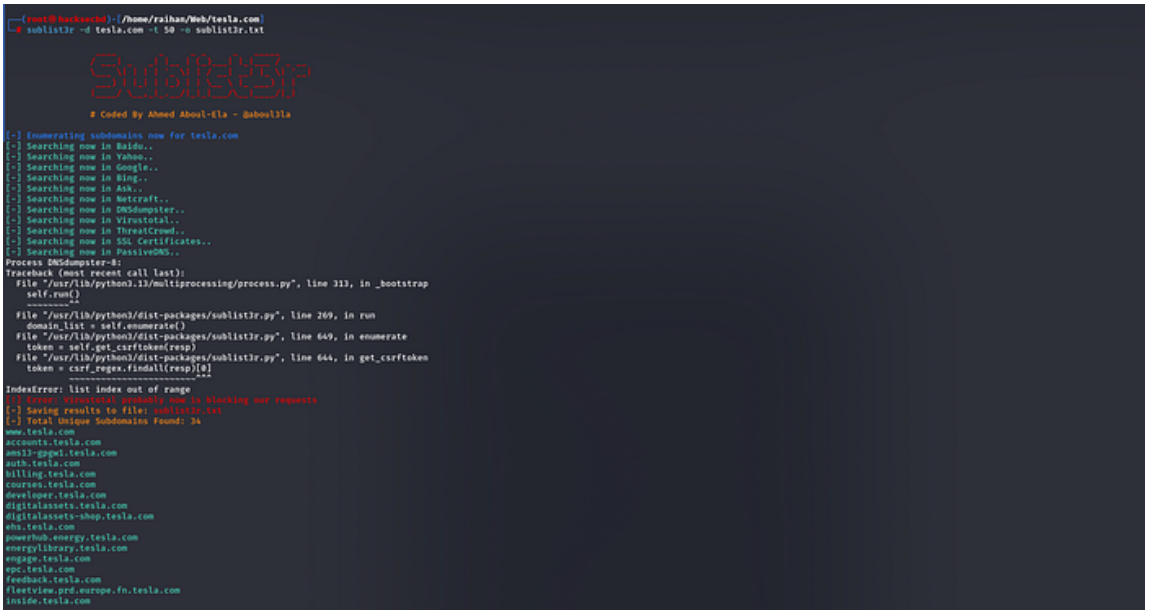
Press enter or click to view image in full size



Sublist3r Example Command:

```
sublist3r -d target.com -t 50 -o sublist3r.txt
```

Press enter or click to view image in full size



Amass Example Command:

```
amass enum -passive -d target.com -o
amass_passive.txt
amass enum -active -d example.com -o
amass_active.txt
amass enum -active -brute -d example.com -
config config.ini -o amass_brute.txt
```

Active Subdomain Enumeration

Active enumeration involves brute-forcing subdomains using wordlists and DNS queries.

Tools: Subbrute, MassDNS, Shuffledns, DNSX

Using a wordlist for brute force :

Example Command : Subdomain Brute-Forcing

```
python3 subbrute.py target.com -w wordlist.txt
-o brute_subs.txt
./Tools/massdns/scripts/subbrute.py target.com
```

```
/usr/share/wordlists/2m-subdomains.txt |
massdns -r /usr/share/wordlists/resolvers.txt -
t A -o S -w target.com.txt
echo target.com | shuffledns -w wordlist.txt -r
resolvers.txt -mode bruteforce | tee
shuffledns.txt
```

Subdomain Brute-Forcing

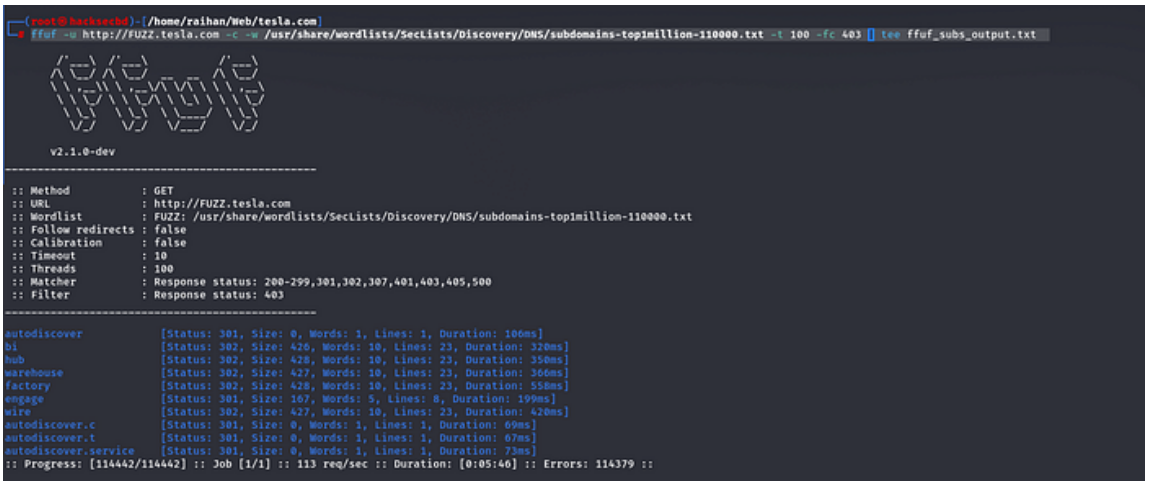
Brute-forcing is useful when passive methods miss some hidden subdomains.

- **Tools**: SubBrute, FFUF, dnsx, MassDNS

  FFUF Example Commad:

  ```
  ffuf -u http://FUZZ.target.com -c -w wordlists
  -t 100 -fc 403 | tee ffuf_subs_output.txt
  ```

  Press enter or click to view image in full size



  use this command to filter out the ffufoutput:

  ```
  for subs in $(cat ffuf_subs_output.txt | awk
  '{print $1}'); do
          echo "${subs}.target.com" >> ffuf-subs-
  final.txt ;
  done
  ```

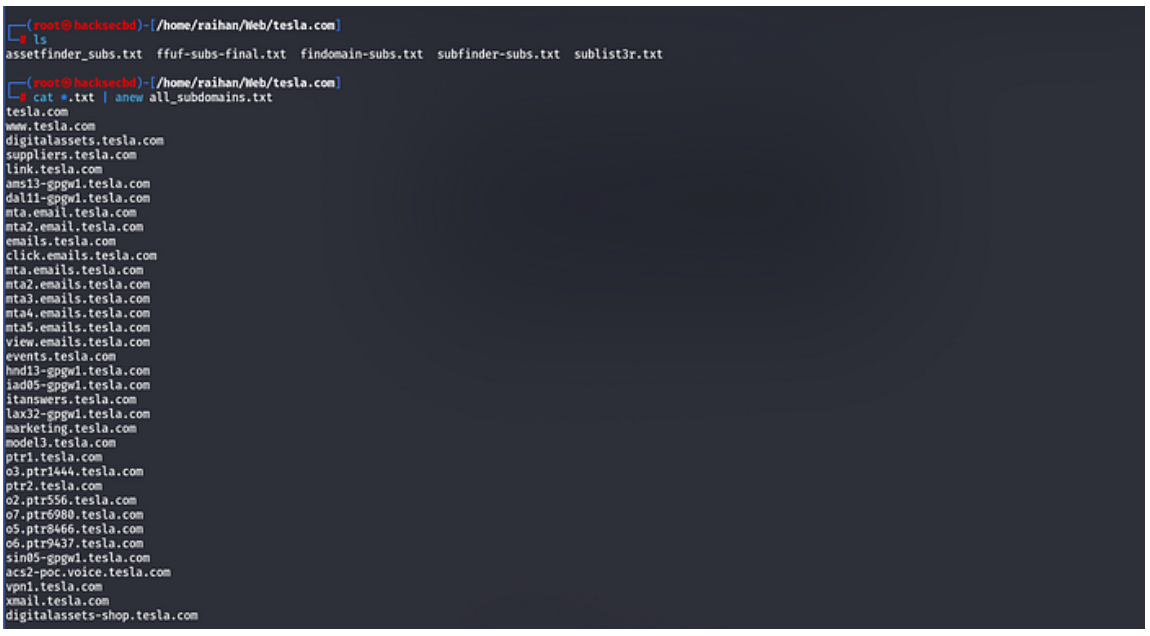
Then remove the `ffuf_subs_output.txt`,file:

Press enter or click to view image in full size

Combine all results into one file using anew:

`cat *.txt | anew all_subdomains.txt`

Press enter or click to view image in full size



Then remove the `assetfinder_subs.txt`, `ffuf_subs_final.txt`,`findomain-subs.txt`, `subfinder_subs.txt`, `sublist3r.txt`file:

`rm assetfinder_subs.txt ffuf-subs-final.txt findomain-subs.txt subfinder-subs.txt sublist3r.txt`

Finding live domains

Not every resolved subdomain will be hosting a web service. HTTP probing helps you identify which subdomains are serving websites.

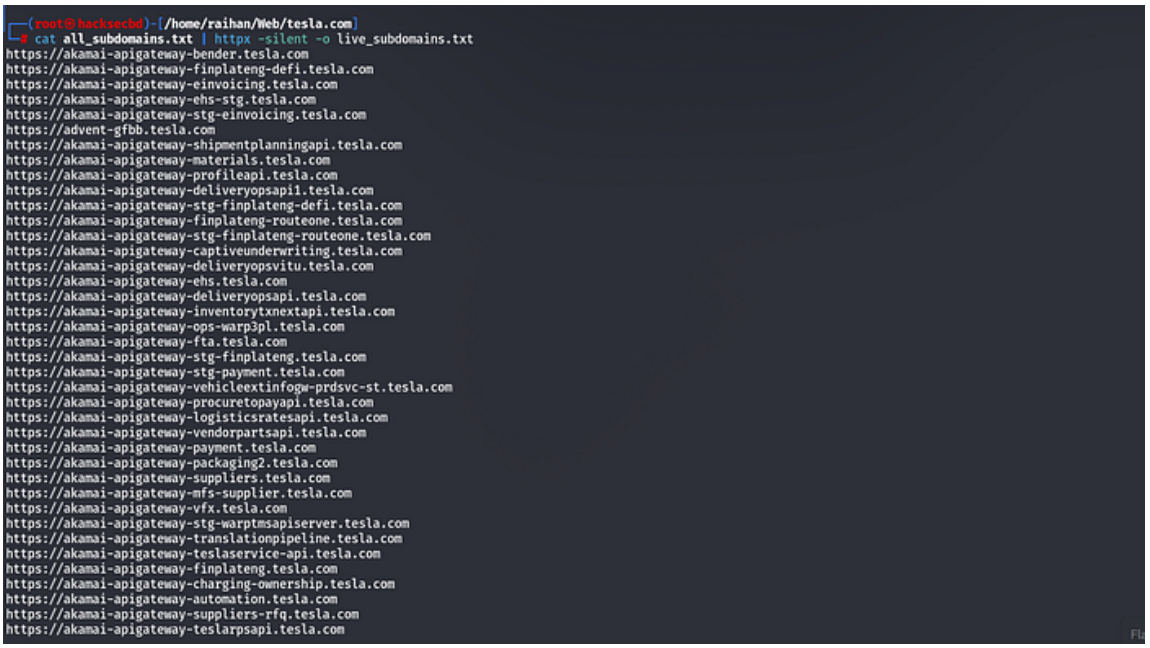
Finding live domains

- **Tools**: `httpx`, `httprobe`

  httpx Example Command:

  `cat all_subdomains.txt | httpx -silent -o live_subdomains.txt`

  Press enter or click to view image in full size



httprobe Example Command:

`cat all_subdomains.txt | httprobe | tee -a alive_subdomains.txt`

Collecting IP Addresses

Once the subdomains are resolved, it’s often useful to collect their associated IP addresses for future port

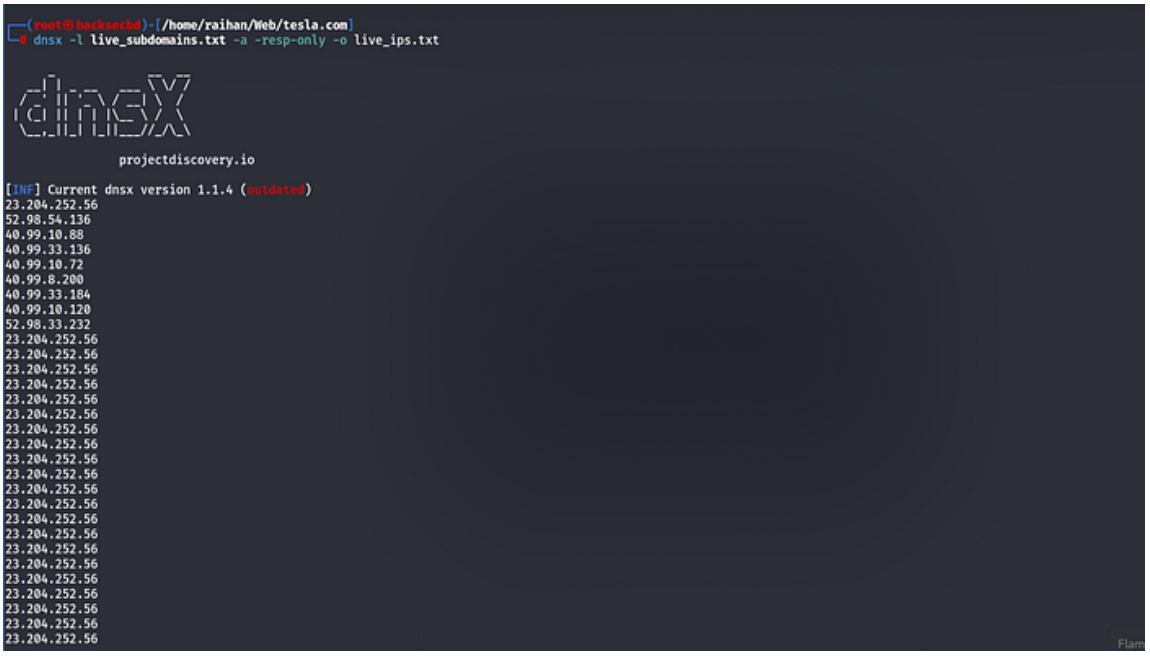
scanning and fingerprinting.

- **Tools**: dnsx, shodanx

  dnsx with IP Output:

  ```
  dnsx -l live_subdomains.txt -a -resp-only -o live_ips.txt
  ```

  Press enter or click to view image in full size



  shodanx Example:

  ```
  shodanx domain --domain "tesla.com"
  ```

  Press enter or click to view image in full size

```
148.135.51.2
154.219.96.164
159.253.120.113
160.30.4.51
163.61.102.106
```

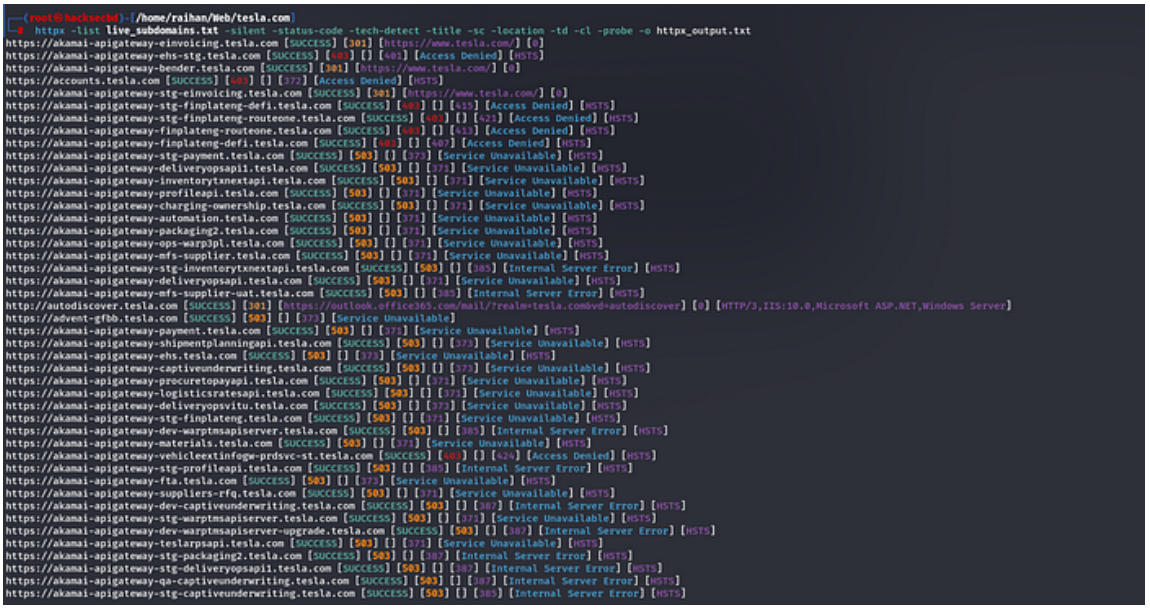
Identifying, Web Services, technology

- **Tools**: `httpx`, `wappalyzer`

Httpx Example Command:

```
httpx -list live_subdomains.txt -silent -status-code -tech-detect -title -sc -location -td -cl -probe -o httpx_output.txt
```

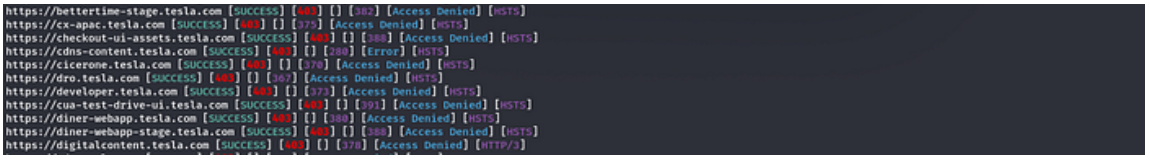
Press enter or click to view image in full size



Finding WAF (Web Application Firewall)

```
cat httpx_output.txt | grep 403
```

Press enter or click to view image in full size



Find Subdomain Without WAF

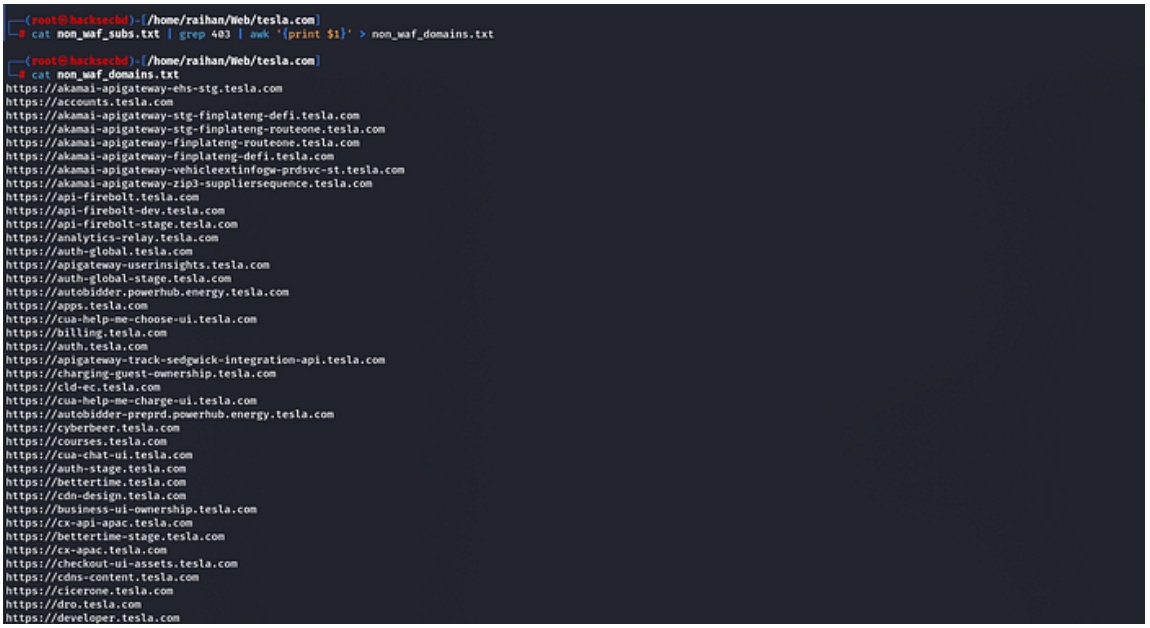
```
cat httpx_output.txt | grep -v -i -E 'Cloudfront|imperva|cloudflare' > non_waf_subs.txt
```

Press enter or click to view image in full size

Visit All Non_Waf Subdomain Manualy

```
cat non_waf_subs.txt | grep 403 | awk '{print $1}'
```

Press enter or click to view image in full size



Screenshotting Web Services

When dealing with a large number of subdomains, taking screenshots of each live web server is a great way to visualize them quickly, identify login portals, or other points of interest.

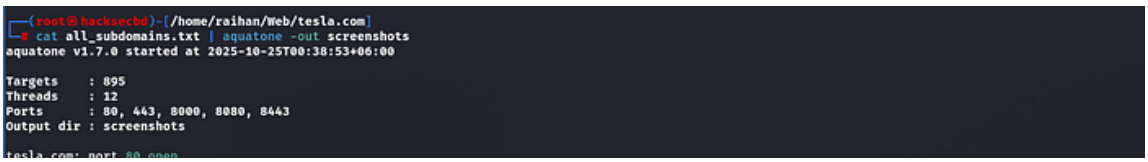
Tools: `Aquatone`, `EyeWitness`, `Gowitness`

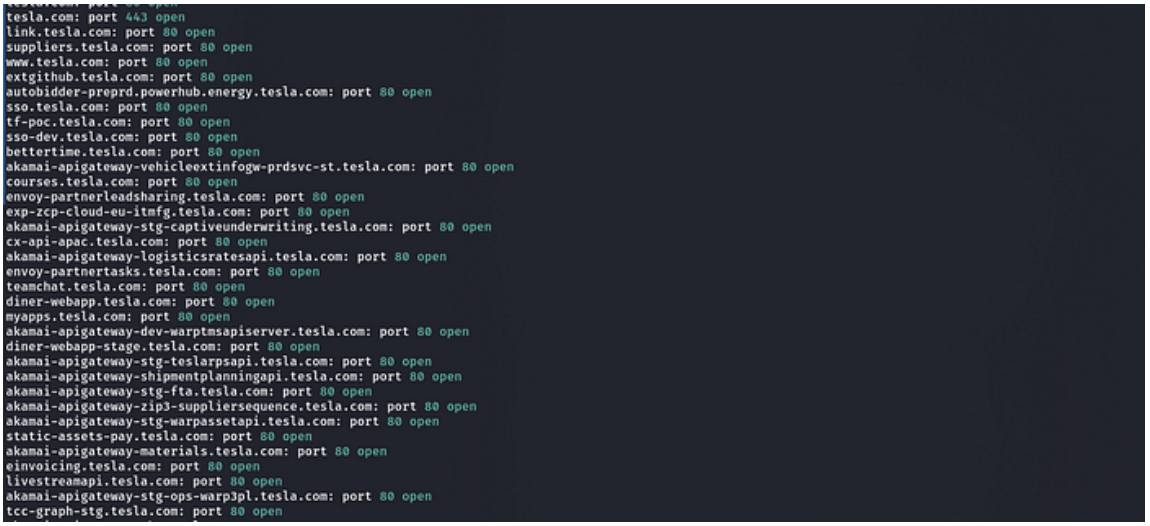
One subscription.
Endless stories.
Become a Medium member
for unlimited reading.
Upgrade now

Aquatone Example Command:

```
cat live_subdomains.txt | aquatone -out screenshots
```

Press enter or click to view image in full size

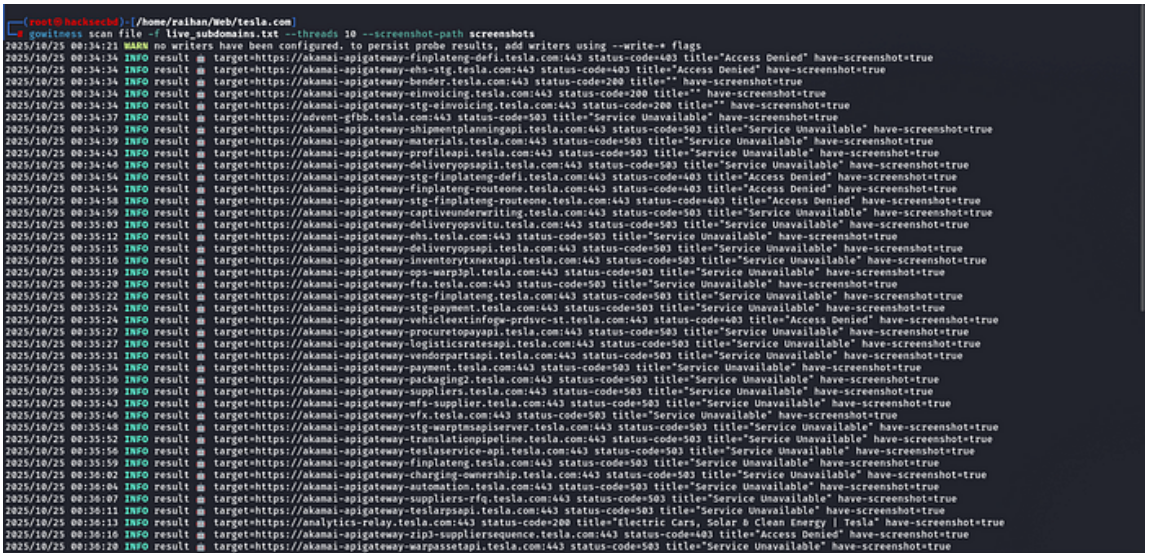




Gowitness Example Command:

```
gowitness scan file -f live_subdomains.txt --threads 10 --screenshot-path screenshots
```

Press enter or click to view image in full size



Eyewitness Example Command:

```
eyewitness -f ips.txt --web --directory /path/to/output
```

Port scranning

Port scanning is the process of probing a network host to identify open, closed, or filtered network ports, which reveals which services are running and potential points of entry.

Tools: nmap, naabu

Naabu Example Command:

```
naabu -list live_subs.txt -o naabu_scans.txt
```

Content Discovery (Directory and File Bruteforcing)

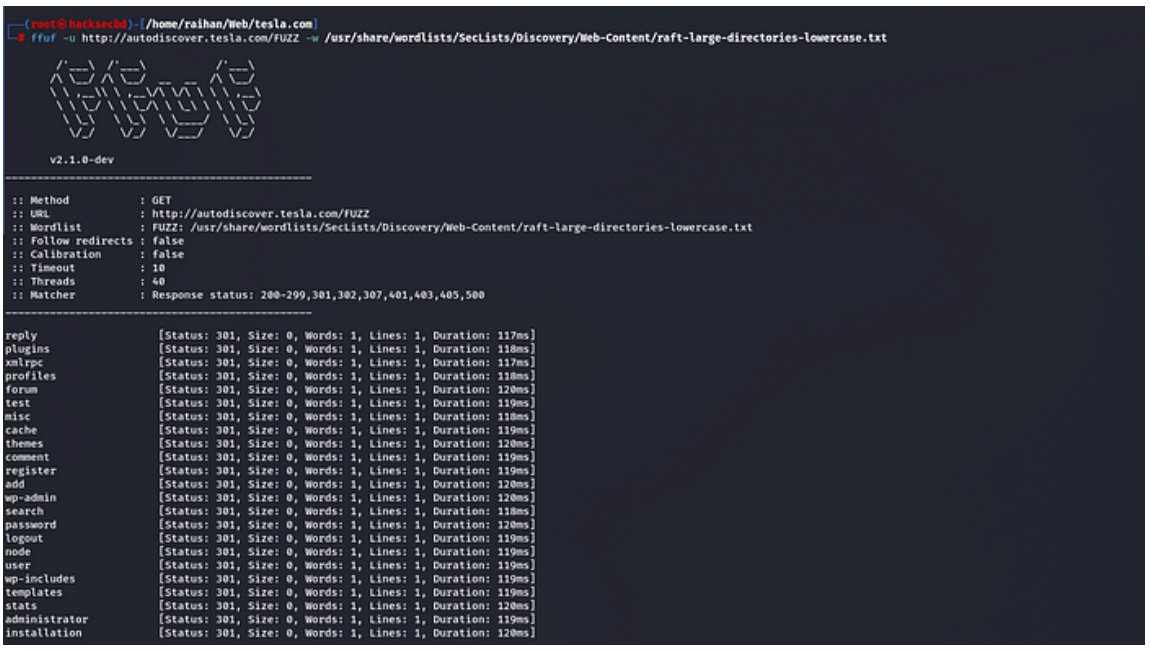
Directory bursting (also known as directory brute forcing) is a web application technology used to find and identify possible hidden directories in websites. This is done with the aim of finding forgotten or unsecured web directories to see if they are vulnerable to exploitation.

Tools: FFuF, Gobuster,Dirsearch

FFUF Example Command:

```
ffuf -u https://example.com/FUZZ -w //path/to/wordlist.txt
```

Press enter or click to view image in full size



Gobuster Example Command:

```
gobuster dir --url https://example.com --
```

```
wordlist /path/to/wordlist.txt
```

Dirsearch Examples Command:

```
dirsearch -u https://target.com/ -w /usr/share/wordlists/custom.txt --full-url --random-agent -x 404,400 -e php,html,js,json,ini
```

Extract URLs

Sometimes, older versions of the website might expose endpoints or parameters that are not available on the live site. You can use archived data from the Wayback Machine to discover such URLs.

Tools: `waybackurls`,`GAU`,`waymore`

Acctive Tool: `katana`

Waybackurls Examples Command:

```
cat live_subs.txt | waybackurls | anew wayback_urls.txt
```

Gau Examples Command:

```
cat live_subs.txt | gau | anew gau_urls.txt
```

Waymore Examples Command:

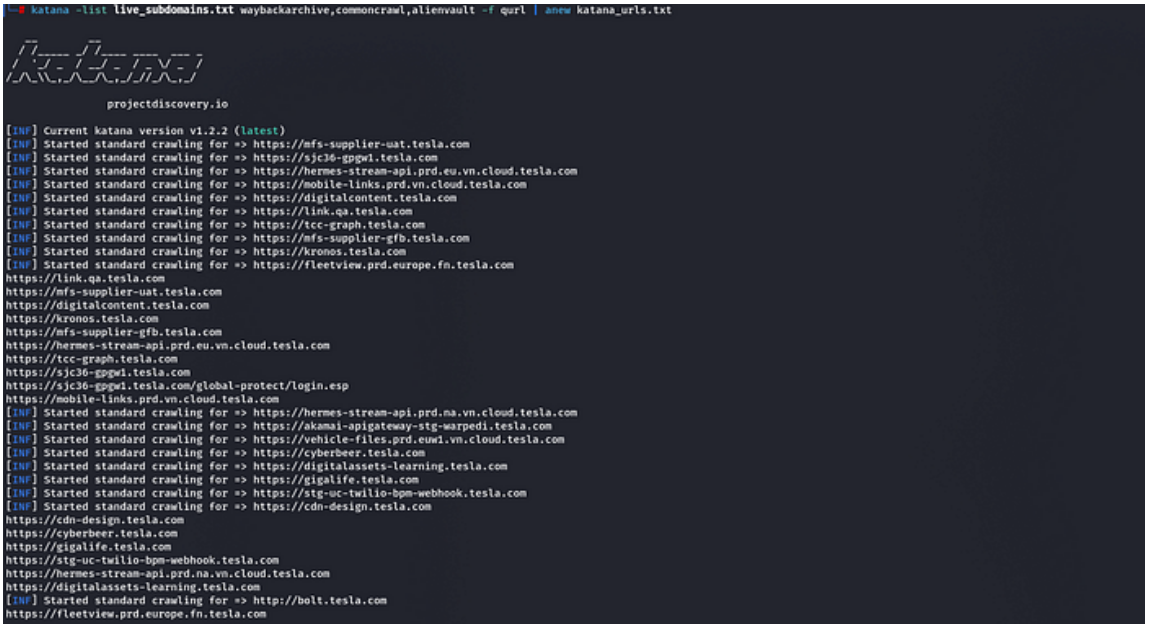
```
cat live_subs.txt | waymore | anew wayback_urls.txt
```

Acctive scanns

Katana Examples Command:

```
katana -list live_subdomain.txt -f qurl | anew katana_urls.txt
```

Press enter or click to view image in full size



Combine all Urls into one file

```
cat katana_urls.txt wayback_urls.txt
gau_urls.txt wayback_urls.txt | anew
all_urls.txt
```

Crawling and Spidering

You can automate crawling of the target website to discover deeper endpoints, hidden forms, or parameters for attacks like XSS or SSRF.

Tools: `katana`, ,`Gospider`

Katana Example Command:

```
katana -list All_urls.txt -silent -d 6 -rl 20 -
f qurl -ef js,css, -o katana_Crawl.txt
```

Press enter or click to view image in full size

Gospider Example Command:

```
gospider -s https://target.com -d 1 -o
gospider_crawl.txt
```

Parameter Discovery

Bruteforcing for parameters can help uncover hidden endpoints vulnerable to injections like SQLi or XSS.

Tools: Arjun,katana, FFuF

Katana Example Commad:

```
katana -list live_subdomains.txt -silent -d 6 -
rl 20 -f qurl -ef js,css, | anew
katana_params.txt
```

Arjun Example Command:

```
arjun -u "https://target.com"
```

Press enter or click to view image in full size

Filtering Interesting URLs

To focus on potentially vulnerable URLs, you can use GF (Grep patterns for fuzzing).

Tools: GF

GF Example Command:

```
cat all_urls.txt | gf xss | anew
xss_candidates.txt
cat all_urls.txt | gf sqli | anew
sqli_candidates.txt
cat all_urls.txt | gf ssti| anew
ssti_candidates.txt
cat all_urls.txt | gf xxe| anew
xxe_candidates.txt
```

Fingerprinting Web Technologies

Understanding the technologies in use can help you tailor your attacks. For instance, knowing the CMS or the web server can lead you to specific vulnerabilities or exploits.

Tools: `Wappalyzer`, `WhatWeb`

Wappalyzer (Browser Extension): Simply browse the target site to see technologies in use.

WhatWeb Example Command:

```
whatweb target.com
```

Press enter or click to view image in full size

Mapping the Attack Surface: Identifying Entry Points

Now that you've collected subdomains, URLs, and ports, it's time to map out the target's attack surface. This includes testing for vulnerable technologies and hidden paths.

JS File Analysis

JavaScript files can leak sensitive information like API keys, hardcoded secrets, or even useful endpoints.

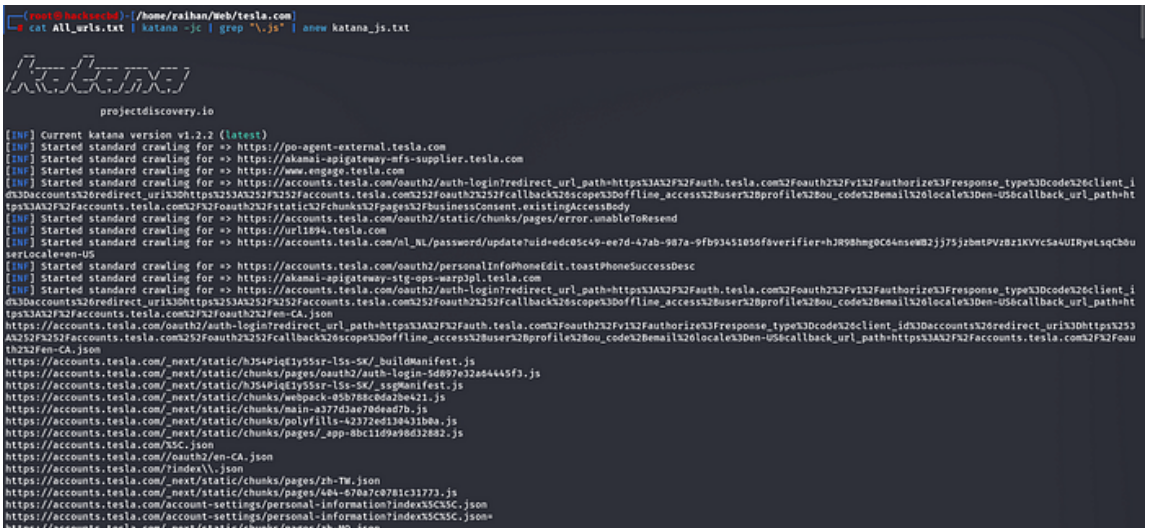
Tools: `katana`, `LinkFinder`, `JSFinder`, `subjs`

```
cat All_urls.txt | grep "\.js" | anew katana_js.txt
```

Katana Example Command:

```
cat all_urls.txt | grep "\.js" | anew katana_js.txt
```

Press enter or click to view image in full size



LinkFinder Example Command:

```
python3 linkfinder.py -i https://target.com/app.js -o cli
```

Subjs Example Command:

```
subjs -i https://target.com | anew js_endpoints.txt
```

Others File Analysis

```
cat All_urls.txt | grep -E "\.txt|\.log|
```

```
\.cache|\.secret|\.db|\.backup|\.yml|\.json|
\.gz|\.rar|\.zip|\.config"
```

Now Start Vulnerability Testing…