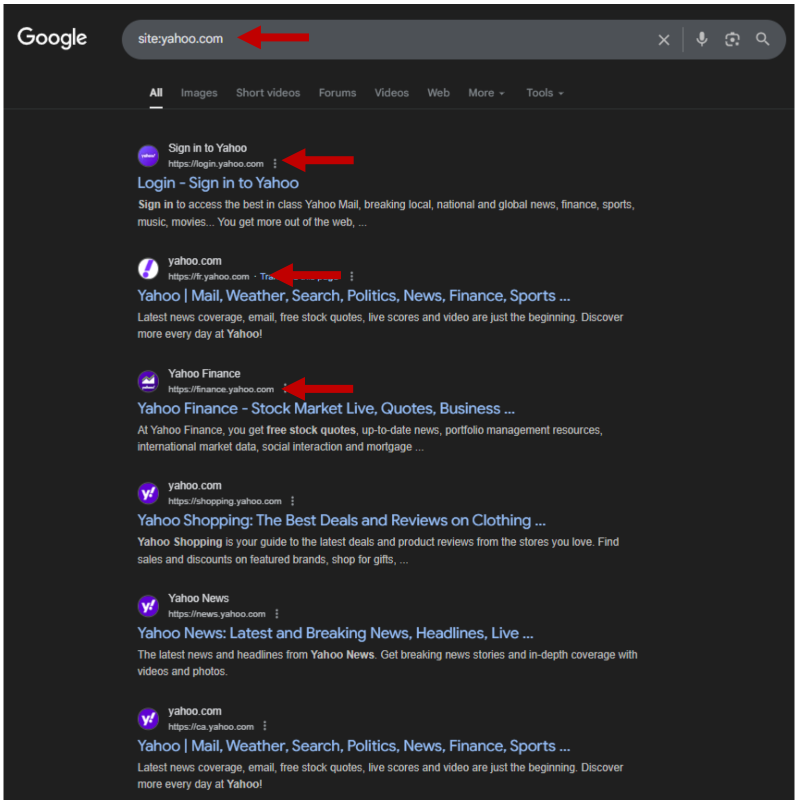
NTI – Ethical Hacking
Assignment 1 Report
Information gathering “reconnaissance”

Prepared by:
Hadeer Amr

For my first task, I was required to select a target and conduct an information gathering phase. This involved identifying a suitable domain, applying reconnaissance techniques, and collecting relevant data to better understand the target’s structure and potential attack surface.

My Target was : yahoo.com and tools I have used : sublist3r, amass, theHarvester

Also I tried searching in google and I have found subdomains



Sign in to Yahoo
https://login.yahoo.com
Login - Sign in to Yahoo
Sign in to access the best in class Yahoo Mail, breaking local, national and global news, finance, sports, music, movies... You get more out of the web, ...

yahoo.com
https://fr.yahoo.com · Tra
Yahoo | Mail, Weather, Search, Politics, News, Finance, Sports ...
Latest news coverage, email, free stock quotes, live scores and video are just the beginning. Discover more every day at **Yahoo!**

Yahoo Finance
https://finance.yahoo.com
Yahoo Finance - Stock Market Live, Quotes, Business ...
At Yahoo Finance, you get **free stock quotes**, up-to-date news, portfolio management resources, international market data, social interaction and mortgage ...

yahoo.com
https://shopping.yahoo.com
Yahoo Shopping: The Best Deals and Reviews on Clothing ...
Yahoo Shopping is your guide to the latest deals and product reviews from the stores you love. Find sales and discounts on featured brands, shop for gifts, ...

Yahoo News
https://news.yahoo.com
Yahoo News: Latest and Breaking News, Headlines, Live ...
The latest news and headlines from **Yahoo News**. Get breaking news stories and in-depth coverage with videos and photos.

yahoo.com
https://ca.yahoo.com
Yahoo | Mail, Weather, Search, Politics, News, Finance, Sports ...
Latest news coverage, email, free stock quotes, live scores and video are just the beginning. Discover more every day at **Yahoo!**

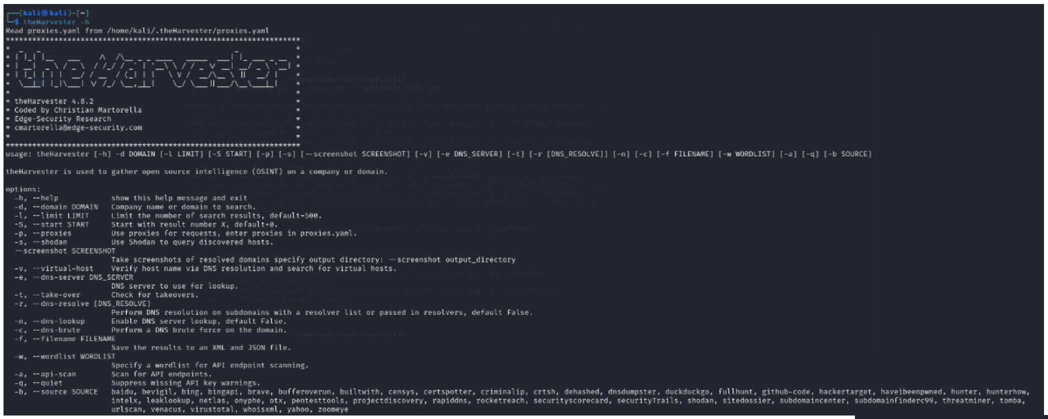
Commands:

```
theHarvester -d yahoo.com -b urlscan  -l 200 > urlscanYahoo.txt
theHarvester -d yahoo.com -b all -l 250 > theharvesterYahoo_Sub.txt
python3 sublist3r.py -d yahoo.com > Sublist3r_subs.txt
python3 sublist3r.py -d yahoo.com -e urlscan
amass enum -passive -d yahoo.com > amassyahoo.txt “most one takes time”
cat  theharvesterYahoo_Sub.txt | sort -u > cleaned.txt
```

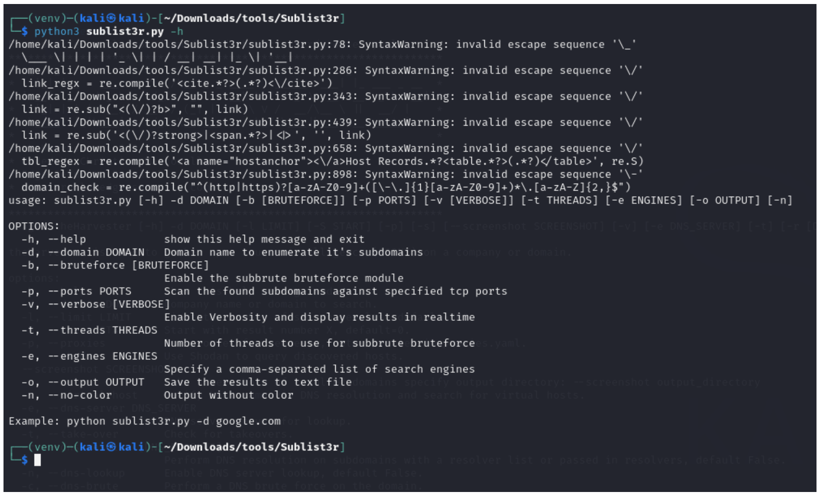
theHarvester

Assignment 1

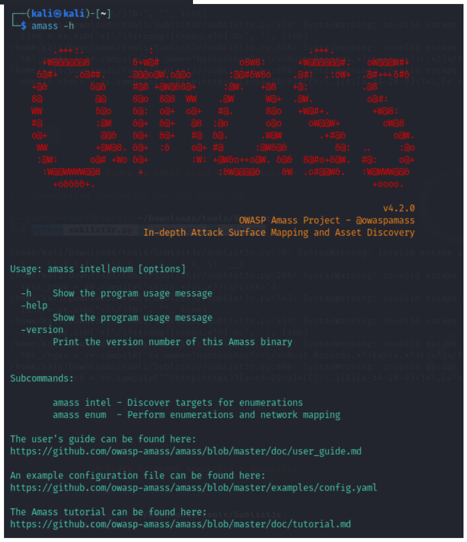
Frist, I started with discovering the three tools using **-h** option



```
┌──(kali㉿kali)-[~]
└─$ theHarvester -h
Read proxies.yaml from /home/kali/.theHarvester/proxies.yaml
*******************************************************************
*  _   _                                            _             *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* theHarvester 4.8.2                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************
usage: theHarvester [-h] -d DOMAIN [-l LIMIT] [-S START] [-p] [-s] [--screenshot SCREENSHOT] [-v] [-e DNS_SERVER] [-t] [-r [DNS_RESOLVE]] [-n] [-c] [-f FILENAME] [-w WORDLIST] [-a] [-q] [-b SOURCE]

theHarvester is used to gather open source intelligence (OSINT) on a company or domain.

options:
  -h, --help            show this help message and exit
  -d, --domain DOMAIN   Company name or domain to search.
  -l, --limit LIMIT     Limit the number of search results, default=500.
  -S, --start START     Start with result number X, default=0.
  -p, --proxies         Use proxies for requests, enter proxies in proxies.yaml.
  -s, --shodan          Use Shodan to query discovered hosts.
  --screenshot SCREENSHOT
                        Take screenshots of resolved domains specify output directory: --screenshot output_directory
  -v, --virtual-host    Verify host name via DNS resolution and search for virtual hosts.
  -e, --dns-server DNS_SERVER
                        DNS server to use for lookup.
  -t, --take-over       Check for takeovers.
  -r, --dns-resolve [DNS_RESOLVE]
                        Perform DNS resolution on subdomains with a resolver list or passed in resolvers, default False.
  -n, --dns-lookup      Enable DNS server lookup, default False.
  -c, --dns-brute       Perform a DNS brute force on the domain.
  -f, --filename FILENAME
                        Save the results to an XML and JSON file.
  -w, --wordlist WORDLIST
                        Specify a wordlist for API endpoint scanning.
  -a, --api-scan        Scan for API endpoints.
  -q, --quiet           Suppress missing API key warnings.
  -b, --source SOURCE   baidu, bevigil, bing, bingapi, brave, bufferoverun, builtwith, censys, certspotter, criminalip, crtsh, dehashed, dnsdumpster, duckduckgo, fullhunt, github-code, hackertarget, haveibeenpwned, hunter, hunterhow,
                        intelx, leaklookup, netlas, onyphe, otx, pentesttools, projectdiscovery, rapiddns, rocketreach, securityscorecard, securityTrails, shodan, sitedossier, subdomaincenter, subdomainfinderc99, threatminer, tomba,
                        urlscan, venacus, virustotal, whoisxml, yahoo, zoomeye
```

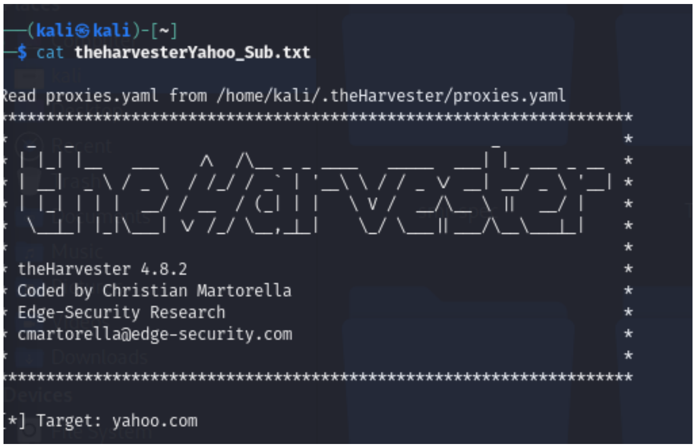


```
┌──(venv)─(kali㉿kali)-[~/Downloads/tools/Sublist3r]
└─$ python3 sublist3r.py -h
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:78: SyntaxWarning: invalid escape sequence '\_'
  \___ \| | | | | '_ \| | / __| __| |_ \| '__|
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:286: SyntaxWarning: invalid escape sequence '\/'
  link_regx = re.compile('<cite.*?>(.*?)<\/cite>')
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:343: SyntaxWarning: invalid escape sequence '\/'
  link = re.sub("<(\/)?b>", "", link)
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:439: SyntaxWarning: invalid escape sequence '\/'
  link = re.sub('<(\/)?strong>|<span.*?>|<|>', '', link)
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:658: SyntaxWarning: invalid escape sequence '\/'
  tbl_regex = re.compile('<a name="hostanchor"><\/a>Host Records.*?<table.*?>(.*?)</table>', re.S)
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:898: SyntaxWarning: invalid escape sequence '\-'
  domain_check = re.compile("^(http|https)?[a-zA-Z0-9]+([\-\.]{1}[a-zA-Z0-9]+)*\.[a-zA-Z]{2,}$")
usage: sublist3r.py [-h] -d DOMAIN [-b [BRUTEFORCE]] [-p PORTS] [-v [VERBOSE]] [-t THREADS] [-e ENGINES] [-o OUTPUT] [-n]

OPTIONS:
  -h, --help            show this help message and exit
  -d, --domain DOMAIN   Domain name to enumerate it's subdomains
  -b, --bruteforce [BRUTEFORCE]
                        Enable the subbrute bruteforce module
  -p, --ports PORTS     Scan the found subdomains against specified tcp ports
  -v, --verbose [VERBOSE]
                        Enable Verbosity and display results in realtime
  -t, --threads THREADS
                        Number of threads to use for subbrute bruteforce
  -e, --engines ENGINES
                        Specify a comma-separated list of search engines
  -o, --output OUTPUT   Save the results to text file
  -n, --no-color        Output without color

Example: python sublist3r.py -d google.com

┌──(venv)─(kali㉿kali)-[~/Downloads/tools/Sublist3r]
└─$ ▮
```

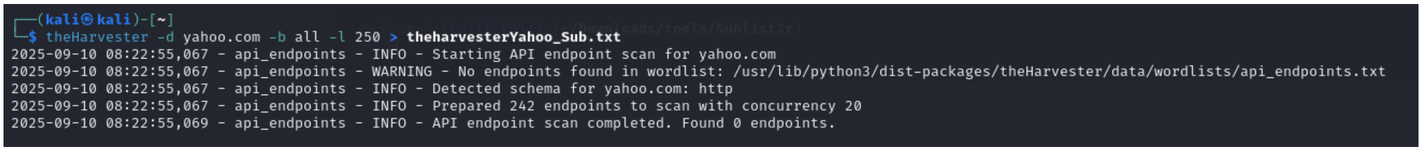


```
┌──(kali㉿kali)-[~]
└─$ amass -h

        .+++:.            :                             .+++.
      +W@@@@@@8        &+W@#               o8W8:      +W@@@@@@#.   oW@@@W#+
     &@#+   .o@##.    .@@@o@W.o@@o       :@@#&W8o    .@#:  .:oW+  .@#+++&#&
    +@&         &@&     #@8 +@W@&8@+     :@W.   +@8   +@:          .@8
    8@          @@      8@o  8@8  WW    .@W      W@+  .@W.          o@#:
    WW          &@o     &@:  o@+  o@+   #@.      8@o   +W@#+.        +W@8:
    #@          :@W     &@+  &@+   @8  :@o       o@o     oW@@W+        oW@8
    o@+          @@&    &@+  &@+   #@  &@.      .W@W       .+#@&         o@W.
     WW         +@W@8.  &@+  :&    o@+ #@      :@W&@&         &@:  ..     :@o
     :@W:      o@# +Wo  &@+        :W: +@W&o++o@W.  &@&  8@#o+&@W.  #@:     o@+
      :W@@WWWW@@8       +              :&W@@@@&    &W   .o#@@W&.    :W@WWW@@&
        +o&&&&+.                                                  +oooo.

                                                                        v4.2.0
                                               OWASP Amass Project - @owaspamass
                                   In-depth Attack Surface Mapping and Asset Discovery

Usage: amass intel|enum [options]

  -h    Show the program usage message
  -help
        Show the program usage message
  -version
        Print the version number of this Amass binary

Subcommands:

        amass intel - Discover targets for enumerations
        amass enum  - Perform enumerations and network mapping

The user's guide can be found here:
https://github.com/owasp-amass/amass/blob/master/doc/user_guide.md

An example configuration file can be found here:
https://github.com/owasp-amass/amass/blob/master/examples/config.yaml

The Amass tutorial can be found here:
https://github.com/owasp-amass/amass/blob/master/doc/tutorial.md
```

And those are my steps running the commands

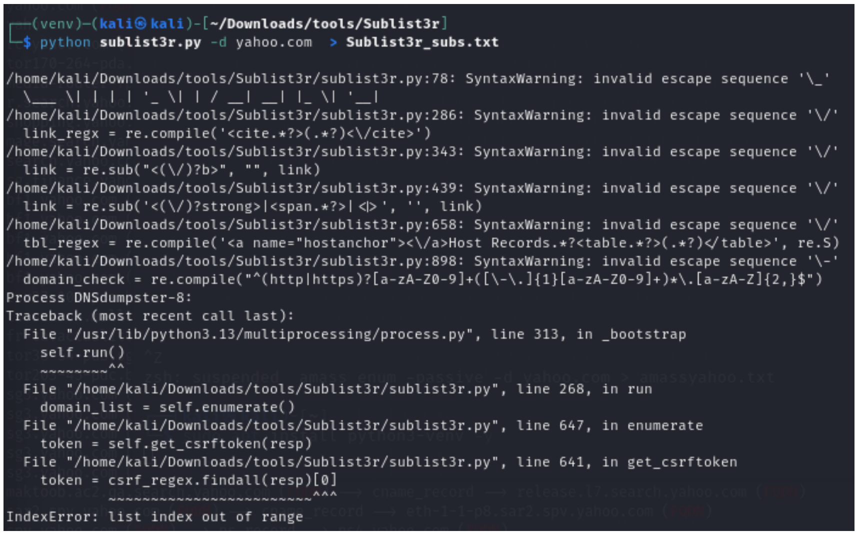


```
┌──(kali㉿kali)-[~]
└─$ cat theharvesterYahoo_Sub.txt

Read proxies.yaml from /home/kali/.theHarvester/proxies.yaml
*******************************************************************
*  _   _                                            _             *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* theHarvester 4.8.2                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************

*] Target: yahoo.com
```

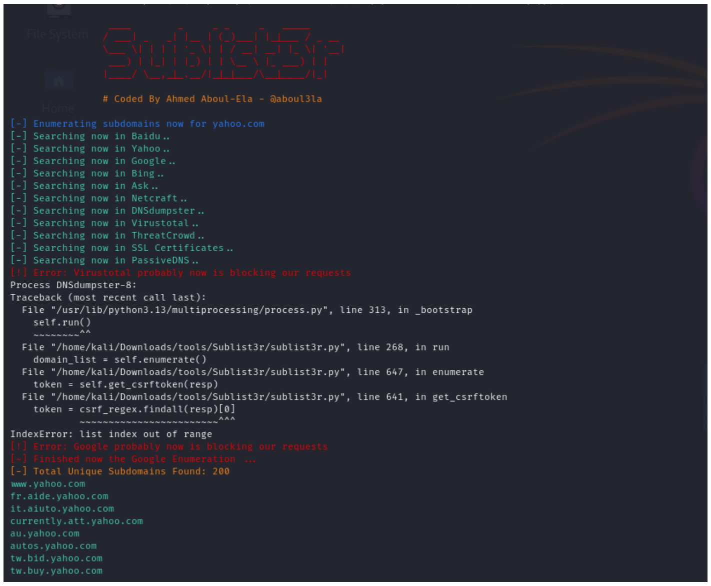


```
┌──(kali㉿kali)-[~]
└─$ theHarvester -d yahoo.com -b all -l 250 > theharvesterYahoo_Sub.txt
2025-09-10 08:22:55,067 - api_endpoints - INFO - Starting API endpoint scan for yahoo.com
2025-09-10 08:22:55,067 - api_endpoints - WARNING - No endpoints found in wordlist: /usr/lib/python3/dist-packages/theHarvester/data/wordlists/api_endpoints.txt
2025-09-10 08:22:55,067 - api_endpoints - INFO - Detected schema for yahoo.com: http
2025-09-10 08:22:55,067 - api_endpoints - INFO - Prepared 242 endpoints to scan with concurrency 20
2025-09-10 08:22:55,069 - api_endpoints - INFO - API endpoint scan completed. Found 0 endpoints.
```

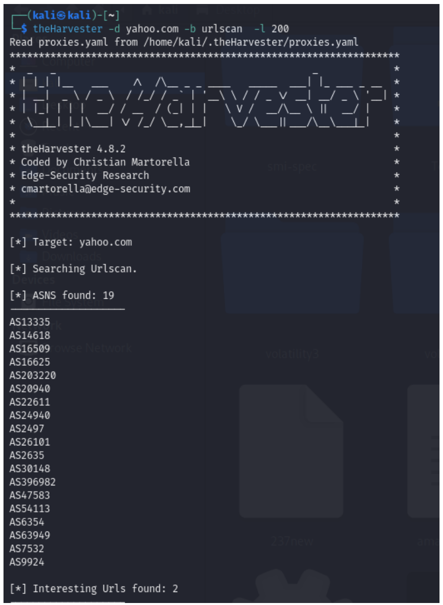
theHarvester



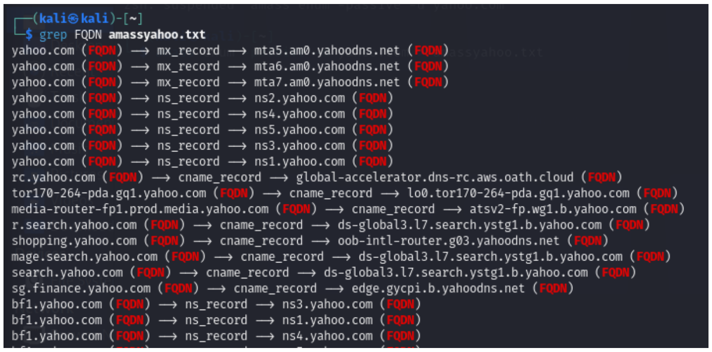
```
┌──(venv)─(kali㉿kali)-[~/Downloads/tools/Sublist3r]
└─$ python sublist3r.py -d yahoo.com  > Sublist3r_subs.txt
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:78: SyntaxWarning: invalid escape sequence '\_'
  \___ \| | | | '_ \| | / __| __| |_ \| '__|
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:286: SyntaxWarning: invalid escape sequence '\/'
  link_regx = re.compile('<cite.*?>(.*?)<\/cite>')
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:343: SyntaxWarning: invalid escape sequence '\/'
  link = re.sub("<(\/)?b>", "", link)
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:439: SyntaxWarning: invalid escape sequence '\/'
  link = re.sub('<(\/)?strong>|<span.*?>|<|>', '', link)
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:658: SyntaxWarning: invalid escape sequence '\/'
  tbl_regex = re.compile('<a name="hostanchor"><\/a>Host Records.*?<table.*?>(.*?)</table>', re.S)
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:898: SyntaxWarning: invalid escape sequence '\-'
  domain_check = re.compile("^(http|https)?[a-zA-Z0-9]+([\-\.]{1}[a-zA-Z0-9]+)*\.[a-zA-Z]{2,}$")
Process DNSdumpster-8:
Traceback (most recent call last):
  File "/usr/lib/python3.13/multiprocessing/process.py", line 313, in _bootstrap
    self.run()
    ~~~~~~~~^^
  File "/home/kali/Downloads/tools/Sublist3r/sublist3r.py", line 268, in run
    domain_list = self.enumerate()
  File "/home/kali/Downloads/tools/Sublist3r/sublist3r.py", line 647, in enumerate
    token = self.get_csrftoken(resp)
  File "/home/kali/Downloads/tools/Sublist3r/sublist3r.py", line 641, in get_csrftoken
    token = csrf_regex.findall(resp)[0]
            ~~~~~~~~~~~~~~~~~~~~~~~~^^^
IndexError: list index out of range
```



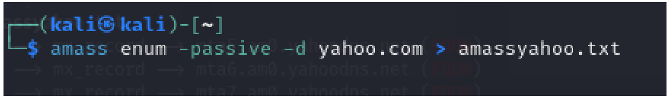
```
                ____        _     _ _     _   _____
               / ___| _   _| |__ | (_)___| |_|___ / _ __
               \___ \| | | | '_ \| | / __| __| |_ \| '__|
                ___) | |_| | |_) | | \__ \ |_ ___) | |
               |____/ \__,_|_.__/|_|_|___/\__|____/|_|

                # Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for yahoo.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[!] Error: Virustotal probably now is blocking our requests
Process DNSdumpster-8:
Traceback (most recent call last):
  File "/usr/lib/python3.13/multiprocessing/process.py", line 313, in _bootstrap
    self.run()
    ~~~~~~~~^^
  File "/home/kali/Downloads/tools/Sublist3r/sublist3r.py", line 268, in run
    domain_list = self.enumerate()
  File "/home/kali/Downloads/tools/Sublist3r/sublist3r.py", line 647, in enumerate
    token = self.get_csrftoken(resp)
  File "/home/kali/Downloads/tools/Sublist3r/sublist3r.py", line 641, in get_csrftoken
    token = csrf_regex.findall(resp)[0]
            ~~~~~~~~~~~~~~~~~~~~~~~~^^^
IndexError: list index out of range
[!] Error: Google probably now is blocking our requests
[~] Finished now the Google Enumeration ...
[-] Total Unique Subdomains Found: 200
www.yahoo.com
fr.aide.yahoo.com
it.aiuto.yahoo.com
currently.att.yahoo.com
au.yahoo.com
autos.yahoo.com
tw.bid.yahoo.com
tw.buy.yahoo.com
```



```
┌──(kali㉿kali)-[~]
└─$ theHarvester -d yahoo.com -b urlscan  -l 200
Read proxies.yaml from /home/kali/.theHarvester/proxies.yaml
*******************************************************************
*  _   _                                            _             *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* theHarvester 4.8.2                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************

[*] Target: yahoo.com

[*] Searching Urlscan.

[*] ASNS found: 19
--------------------
AS13335
AS14618
AS16509
AS16625
AS203220
AS20940
AS22611
AS24940
AS2497
AS26101
AS2635
AS30148
AS396982
AS47583
AS54113
AS6354
AS63949
AS7532
AS9924

[*] Interesting Urls found: 2
```

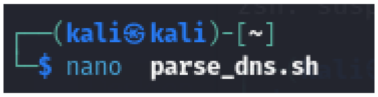


```
┌──(kali㉿kali)-[~]
└─$ grep FQDN amassyahoo.txt
yahoo.com (FQDN) ⟶ mx_record ⟶ mta5.am0.yahoodns.net (FQDN)
yahoo.com (FQDN) ⟶ mx_record ⟶ mta6.am0.yahoodns.net (FQDN)
yahoo.com (FQDN) ⟶ mx_record ⟶ mta7.am0.yahoodns.net (FQDN)
yahoo.com (FQDN) ⟶ ns_record ⟶ ns2.yahoo.com (FQDN)
yahoo.com (FQDN) ⟶ ns_record ⟶ ns4.yahoo.com (FQDN)
yahoo.com (FQDN) ⟶ ns_record ⟶ ns5.yahoo.com (FQDN)
yahoo.com (FQDN) ⟶ ns_record ⟶ ns3.yahoo.com (FQDN)
yahoo.com (FQDN) ⟶ ns_record ⟶ ns1.yahoo.com (FQDN)
rc.yahoo.com (FQDN) ⟶ cname_record ⟶ global-accelerator.dns-rc.aws.oath.cloud (FQDN)
tor170-264-pda.gq1.yahoo.com (FQDN) ⟶ cname_record ⟶ lo0.tor170-264-pda.gq1.yahoo.com (FQDN)
media-router-fp1.prod.media.yahoo.com (FQDN) ⟶ cname_record ⟶ atsv2-fp.wg1.b.yahoo.com (FQDN)
r.search.yahoo.com (FQDN) ⟶ cname_record ⟶ ds-global3.l7.search.ystg1.b.yahoo.com (FQDN)
shopping.yahoo.com (FQDN) ⟶ cname_record ⟶ oob-intl-router.g03.yahoodns.net (FQDN)
mage.search.yahoo.com (FQDN) ⟶ cname_record ⟶ ds-global3.l7.search.ystg1.b.yahoo.com (FQDN)
search.yahoo.com (FQDN) ⟶ cname_record ⟶ ds-global3.l7.search.ystg1.b.yahoo.com (FQDN)
sg.finance.yahoo.com (FQDN) ⟶ cname_record ⟶ edge.gycpi.b.yahoodns.net (FQDN)
bf1.yahoo.com (FQDN) ⟶ ns_record ⟶ ns3.yahoo.com (FQDN)
bf1.yahoo.com (FQDN) ⟶ ns_record ⟶ ns1.yahoo.com (FQDN)
bf1.yahoo.com (FQDN) ⟶ ns_record ⟶ ns4.yahoo.com (FQDN)
```



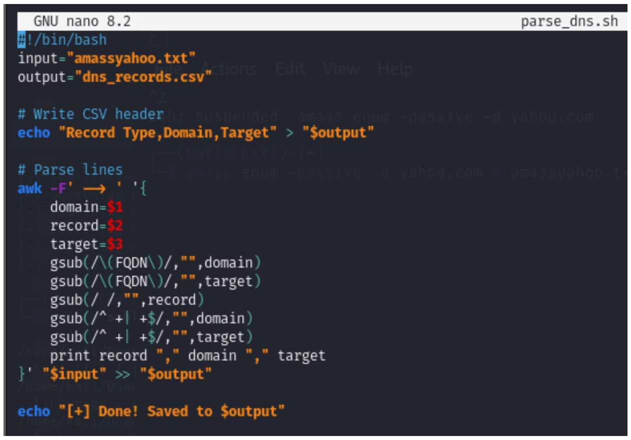
```
┌──(kali㉿kali)-[~]
└─$ amass enum -passive -d yahoo.com > amassyahoo.txt
```

Amass tool gave me a lot of information but was not organized and I found some duplicates to make it more easier I searched for script to automate the sorting and organizing and I used this one to gave me excel sheet and I attached it in the link that contain my findings in this stage

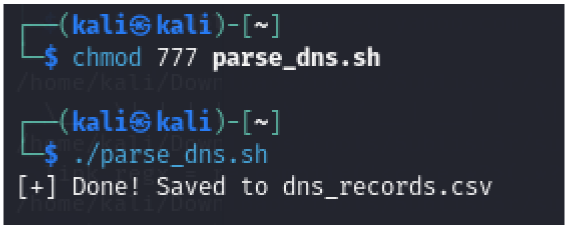


```
┌──(kali㉿kali)-[~]
└─$ nano  parse_dns.sh
```

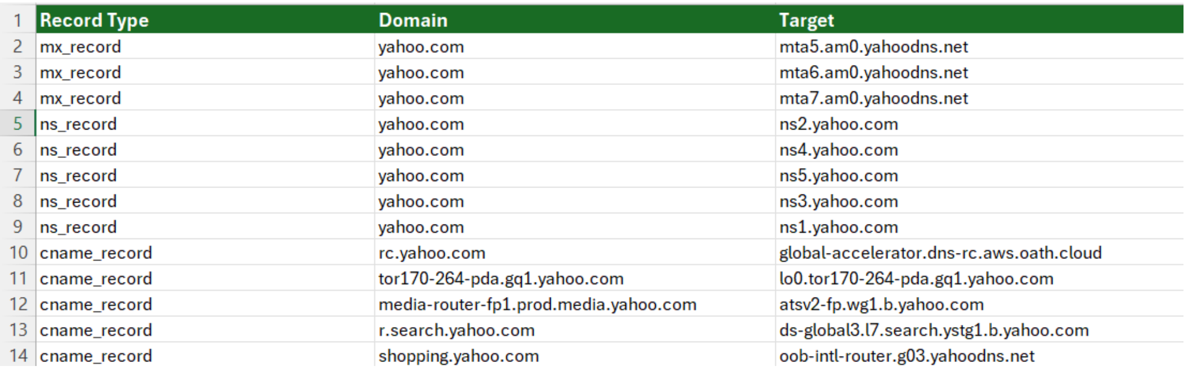
theHarvester



```
GNU nano 8.2                                                    parse_dns.sh
#!/bin/bash
input="amassyahoo.txt"
output="dns_records.csv"

# Write CSV header
echo "Record Type,Domain,Target" > "$output"

# Parse lines
awk -F' → ' '{
    domain=$1
    record=$2
    target=$3
    gsub(/\(FQDN\)/,"",domain)
    gsub(/\(FQDN\)/,"",target)
    gsub(/ /,"",record)
    gsub(/^ +| +$/,"",domain)
    gsub(/^ +| +$/,"",target)
    print record "," domain "," target
}' "$input" >> "$output"

echo "[+] Done! Saved to $output"
```



```
┌──(kali㉿kali)-[~]
└─$ chmod 777 parse_dns.sh

┌──(kali㉿kali)-[~]
└─$ ./parse_dns.sh
[+] Done! Saved to dns_records.csv
```



1	Record Type	Domain	Target
2	mx_record	yahoo.com	mta5.am0.yahoodns.net
3	mx_record	yahoo.com	mta6.am0.yahoodns.net
4	mx_record	yahoo.com	mta7.am0.yahoodns.net
5	ns_record	yahoo.com	ns2.yahoo.com
6	ns_record	yahoo.com	ns4.yahoo.com
7	ns_record	yahoo.com	ns5.yahoo.com
8	ns_record	yahoo.com	ns3.yahoo.com
9	ns_record	yahoo.com	ns1.yahoo.com
10	cname_record	rc.yahoo.com	global-accelerator.dns-rc.aws.oath.cloud
11	cname_record	tor170-264-pda.gq1.yahoo.com	lo0.tor170-264-pda.gq1.yahoo.com
12	cname_record	media-router-fp1.prod.media.yahoo.com	atsv2-fp.wg1.b.yahoo.com
13	cname_record	r.search.yahoo.com	ds-global3.l7.search.ystg1.b.yahoo.com
14	cname_record	shopping.yahoo.com	oob-intl-router.g03.yahoodns.net

---

The reconnaissance uncovered a wide range of Yahoo's digital footprint information I have found Subdomains & FQDNs (over 200 entries via Sublist3r and Amass), PTR Records (reverse DNS mappings), Email Infrastructure (mail server hosts), IPv4 and IPv6 Addresses, Netblocks & ASNs (infrastructure ownership and ranges), Emails (4 public addresses via theHarvester)

I utilized three tools Sublist3r, Amass, and theHarvester to enumerate Yahoo's public-facing infrastructure and gather OSINT data. The results are summarized as follows:

Using Sublist3r it Successfully enumerated 200+ subdomains associated with yahoo.com and these subdomains represent different parts of Yahoo's infrastructure (mail servers, CDN nodes, internal naming conventions, etc.).

And using Amass provided deeper enumeration and infrastructure mapping, including: FQDNs (Fully Qualified Domain Names)examples include oxy-oxygen-2001-4998-44-803f-1184.ne1.yahoo.com and 218.43.30.72.in-addr.arpa (reverse DNS format) and PTR Records (Pointer Records): Mapping IPs back to hostnames example 218.43.30.72.in-addr.arpa → unknown.yahoo.com and Subdomains: Examples include loo.tor394-300-pda.bf1.yahoo.com and sonic312-7.consmr.mail.ir2.yahoo.com and Email-related Hosts: Such as sonic312-7.consmr.mail.ir2.yahoo.com (part of Yahoo Mail servers) and IPv4 and IPv6 addresses as IPv4: 72.30.30.237, 216.155.192.212 and IPv6: 2001:4998:ef99:209::2019 and Netblocks (IP Ranges): Examples include 72.30.0.0/19 and 2001:4998::/32 and ASNs Identified Autonomous System Numbers related to Yahoo infrastructure.

And finally using theHarvester I Collected 4 public email addresses linked to yahoo.com and unfortunately No employee names or LinkedIn profiles were discovered limited internal data exposure from this tool.

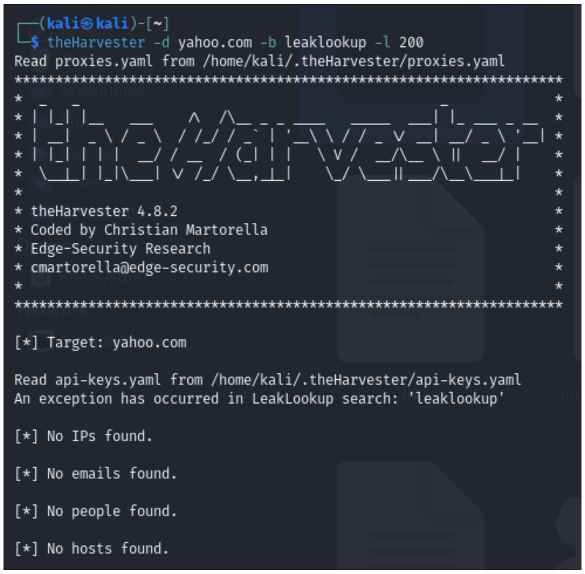
This information provides a comprehensive view of Yahoo's external attack surface and forms the baseline for further vulnerability assessment.

theHarvester

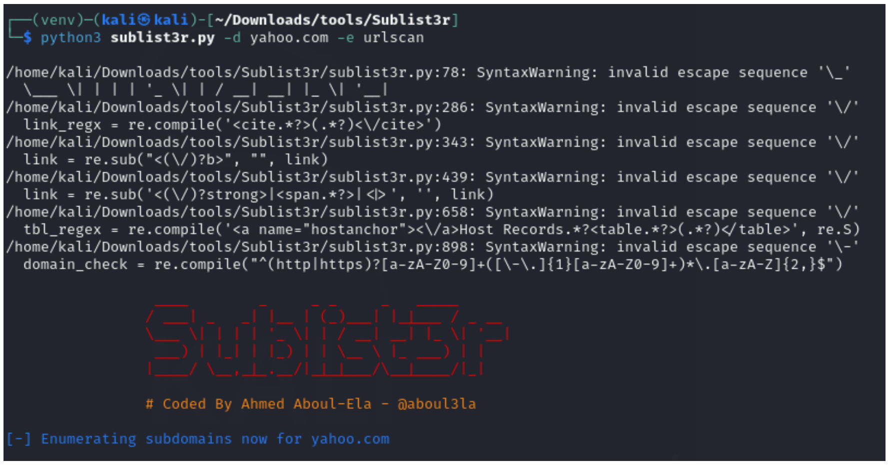
This is a list of all the information I retrieved is uploaded in this link

https://drive.google.com/drive/folders/1TKoyzk-3jfsYPuZfYQEvvPrXrRStWs8W?usp=sharing

Using multiple sources truly affect the results for example using theHarvester -b all gave me 4 emails, 1 URL, 6171 hosts, 172 Ips and 19 ASNS while -b urlscan gave me 19 ASNS, 2 URLs, 60 Ips and 2 hosts and some resources does not gave me any results



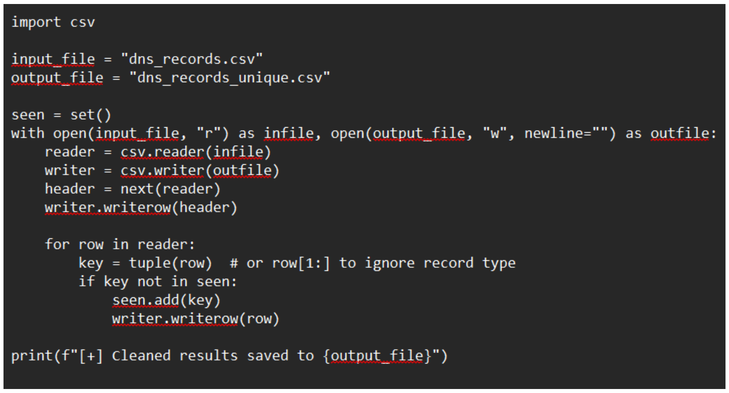
```
┌──(kali㉿kali)-[~]
└─$ theHarvester -d yahoo.com -b leaklookup -l 200
Read proxies.yaml from /home/kali/.theHarvester/proxies.yaml
*******************************************************************
*  _   _                                            _             *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* theHarvester 4.8.2                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************

[*] Target: yahoo.com

Read api-keys.yaml from /home/kali/.theHarvester/api-keys.yaml
An exception has occurred in LeakLookup search: 'leaklookup'

[*] No IPs found.

[*] No emails found.

[*] No people found.

[*] No hosts found.
```



```
┌──(venv)─(kali㉿kali)-[~/Downloads/tools/Sublist3r]
└─$ python3 sublist3r.py -d yahoo.com -e urlscan

/home/kali/Downloads/tools/Sublist3r/sublist3r.py:78: SyntaxWarning: invalid escape sequence '\_'
  \___ \| | | | '_ \| | / __| __| |_ \| '__|
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:286: SyntaxWarning: invalid escape sequence '\/'
  link_regx = re.compile('<cite.*?>(.*?)<\/cite>')
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:343: SyntaxWarning: invalid escape sequence '\/'
  link = re.sub("<(\/)?b>", "", link)
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:439: SyntaxWarning: invalid escape sequence '\/'
  link = re.sub('<(\/)?strong>|<span.*?>|<|>', '', link)
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:658: SyntaxWarning: invalid escape sequence '\/'
  tbl_regex = re.compile('<a name="hostanchor"><\/a>Host Records.*?<table.*?>(.*?)</table>', re.S)
/home/kali/Downloads/tools/Sublist3r/sublist3r.py:898: SyntaxWarning: invalid escape sequence '\-'
  domain_check = re.compile("^(http|https)?[a-zA-Z0-9]+([\-\.]{1}[a-zA-Z0-9]+)*\.[a-zA-Z]{2,}$")

                 ____        _     _ _     _   _____
                / ___| _   _| |__ | (_)___| |_|___ / _ __
                \___ \| | | | '_ \| | / __| __| |_ \| '__|
                 ___) | |_| | |_) | | \__ \ |_ ___) | |
                |____/ \__,_|_.__/|_|_|___/\__|____/|_|

                # Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for yahoo.com
```

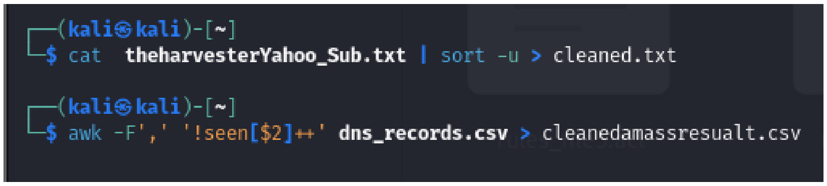
The source engine gave me the most useful results was amass the global default option and theHarvester -b all option gave me wider information and results that could help me more in my upcoming phases, but I thought it will differ based on the target.

I have found some duplicate entries across engines especially using amass tool after some thinking I thought about using sort using this ***cat theharvesterYahoo_Sub.txt | sort -u > cleaned.txt*** also another idea to enhance my script to make amass results better

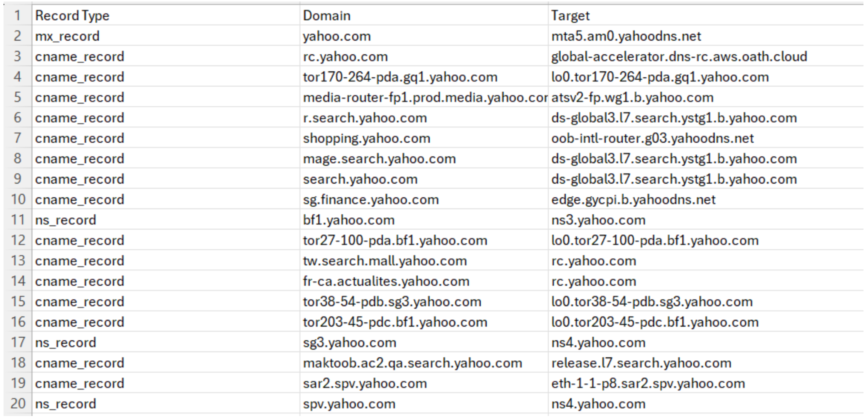


```
import csv

input_file = "dns_records.csv"
output_file = "dns_records_unique.csv"

seen = set()
with open(input_file, "r") as infile, open(output_file, "w", newline="") as outfile:
    reader = csv.reader(infile)
    writer = csv.writer(outfile)
    header = next(reader)
    writer.writerow(header)

    for row in reader:
        key = tuple(row)  # or row[1:] to ignore record type
        if key not in seen:
            seen.add(key)
            writer.writerow(row)

print(f"[+] Cleaned results saved to {output_file}")
```

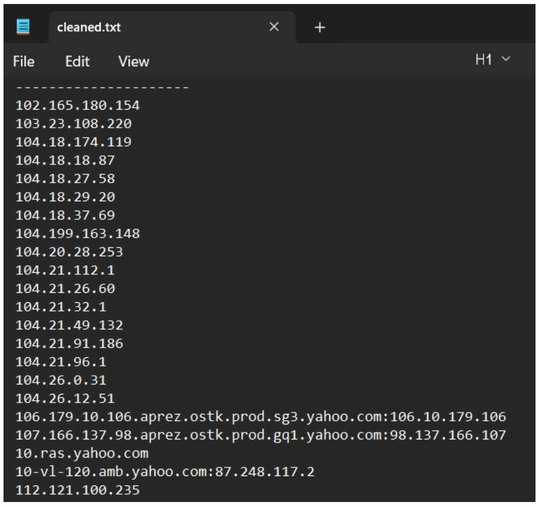
or use this ***awk -F',' '!seen[$2]++' dns_records.csv > cleaned.csv*** also I can use feature in excel to remove the duplicates and I can gather all the information from different tools and engines and but in one file then perform the cleaning to make results clearer.



```
┌──(kali㉿kali)-[~]
└─$ cat  theharvesterYahoo_Sub.txt | sort -u > cleaned.txt

┌──(kali㉿kali)-[~]
└─$ awk -F',' '!seen[$2]++' dns_records.csv > cleanedamassresualt.csv
```



1	Record Type	Domain	Target
2	mx_record	yahoo.com	mta5.am0.yahoodns.net
3	cname_record	rc.yahoo.com	global-accelerator.dns-rc.aws.oath.cloud
4	cname_record	tor170-264-pda.gq1.yahoo.com	lo0.tor170-264-pda.gq1.yahoo.com
5	cname_record	media-router-fp1.prod.media.yahoo.cor	atsv2-fp.wg1.b.yahoo.com
6	cname_record	r.search.yahoo.com	ds-global3.l7.search.ystg1.b.yahoo.com
7	cname_record	shopping.yahoo.com	oob-intl-router.g03.yahoodns.net
8	cname_record	mage.search.yahoo.com	ds-global3.l7.search.ystg1.b.yahoo.com
9	cname_record	search.yahoo.com	ds-global3.l7.search.ystg1.b.yahoo.com
10	cname_record	sg.finance.yahoo.com	edge.gycpi.b.yahoodns.net
11	ns_record	bf1.yahoo.com	ns3.yahoo.com
12	cname_record	tor27-100-pda.bf1.yahoo.com	lo0.tor27-100-pda.bf1.yahoo.com
13	cname_record	tw.search.mall.yahoo.com	rc.yahoo.com
14	cname_record	fr-ca.actualites.yahoo.com	rc.yahoo.com
15	cname_record	tor38-54-pdb.sg3.yahoo.com	lo0.tor38-54-pdb.sg3.yahoo.com
16	cname_record	tor203-45-pdc.bf1.yahoo.com	lo0.tor203-45-pdc.bf1.yahoo.com
17	ns_record	sg3.yahoo.com	ns4.yahoo.com
18	cname_record	maktoob.ac2.qa.search.yahoo.com	release.l7.search.yahoo.com
19	cname_record	sar2.spv.yahoo.com	eth-1-1-p8.sar2.spv.yahoo.com
20	ns_record	spv.yahoo.com	ns4.yahoo.com



```
---------------------
102.165.180.154
103.23.108.220
104.18.174.119
104.18.18.87
104.18.27.58
104.18.29.20
104.18.37.69
104.199.163.148
104.20.28.253
104.21.112.1
104.21.26.60
104.21.32.1
104.21.49.132
104.21.91.186
104.21.96.1
104.26.0.31
104.26.12.51
106.179.10.106.aprez.ostk.prod.sg3.yahoo.com:106.10.179.106
107.166.137.98.aprez.ostk.prod.gq1.yahoo.com:98.137.166.107
10.ras.yahoo.com
10-vl-120.amb.yahoo.com:87.248.117.2
112.121.100.235
```

Based on the data I have found an attacker can use some of this data in phishing or social engineering attacks as the four e-mails I have found using theHarvester and the Email-related Hosts I found using amass tool or he could find LinkedIn of employees could help in getting more information to use in phishing or social engineering attacks or any attacks related to what he could found also based on the subdomain he can perform some searching to find vulnerabilities and got more access

To sum up what I have learned from this task I have learned that using different tools in reconnaissance is very important because each one shows different parts of the target. For example, Sublist3r was very good at finding many subdomains quickly, Amass gave me a deeper view of the infrastructure, and theHarvester helped with OSINT, like e-mails. I also learned that some data was repeated across the tools, but this was useful because it confirmed the information and made it more reliable. At the same time, I found unique results that only one tool was able to discover. Through this task, I understood better how domains, subdomains, FQDNs, PTR records, and netblocks are all connected, and how they reveal the structure of a company's online presence. Even though theHarvester gave me only four emails and no employee names, I realized that even a small amount of data can still be used in attacks like phishing.

Another thing I noticed is how big companies like Yahoo manage very complex infrastructures. I found more than 200 subdomains, and this showed me the importance of organizing and cleaning the data so it can be used effectively.

Finally, this task gave me a strong foundation for the next steps in penetration testing, such as vulnerability scanning and exploring the attack surface in more detail.

Disclaimer

This report is for educational purposes only.