

Assignment 1

NTI – Ethical Hacking

Assignment 1 Report

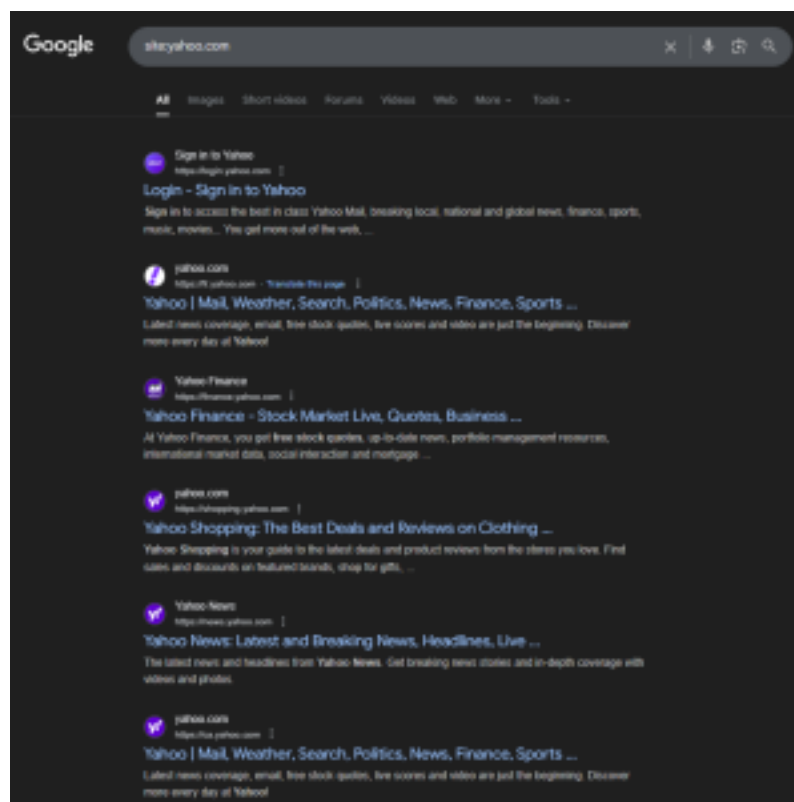
Information gathering “reconnaissance”

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



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
```

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

```
> cd /root/.kali/kali1/~Downloads/tools/sublist3r[
~$ python sublist3r.py -d
/home/kali/.Downloads/tools/sublist3r/sublist3r.py:78: SyntaxWarning: invalid escape sequence '\.'
  MTTTTT.MTTTTT.MTTTTT
/home/kali/.Downloads/tools/sublist3r/sublist3r.py:266: SyntaxWarning: invalid escape sequence '\.'
  link_regex = re.compile('^(http|https)://.*')
/home/kali/.Downloads/tools/sublist3r/sublist3r.py:343: SyntaxWarning: invalid escape sequence '\.'
  link = re.compile('^(http|https)://.*').link()
/home/kali/.Downloads/tools/sublist3r/sublist3r.py:490: SyntaxWarning: invalid escape sequence '\.'
  link = re.compile('^(http|https)://.*').link()
/home/kali/.Downloads/tools/sublist3r/sublist3r.py:556: SyntaxWarning: invalid escape sequence '\.'
  url_regex = re.compile('^(http|https)://.*')
/home/kali/.Downloads/tools/sublist3r/sublist3r.py:686: SyntaxWarning: invalid escape sequence '\.'
  domain_check = re.compile('^(http|https)://.*')
/home/kali/.Downloads/tools/sublist3r/sublist3r.py:712: SyntaxWarning: invalid escape sequence '\.'
  engine = sublist3r.py [-h] -d DOMAIN [-k BRUTEFORCE] [-p PORTS] [-v [VERBOSE]] [-t THREADS] [-e ENGINES] [-o OUTPUT] [-c]

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

Example: python sublist3r.py -d google.com

> cd /root/.kali/kali1/~Downloads/tools/sublist3r[
~$
```

[illegible]

And those are my steps running the commands

[illegible]

```
[kali@kali:~]$ theharvester -d yahoo.com -b all -l 250 > theharvesterYahoo_Sub.txt
2025-09-10 06:22:55,867 - api_endpoints - INFO - Starting API endpoint scan for yahoo.com
2025-09-10 06:22:55,867 - api_endpoints - WARNING - No endpoints found in wordlist: /usr/lib/python3/dist-packages/thefarvester/data/wordlists/api_endpoints.txt
2025-09-10 06:22:55,867 - api_endpoints - INFO - Detected schema for yahoo.com: http
2025-09-10 06:22:55,867 - api_endpoints - INFO - Prepared 242 endpoints to scan with concurrency 20
2025-09-10 06:22:55,869 - api_endpoints - INFO - API endpoint scan completed. Found 0 endpoints.
```

theHarvester

Assignment 1

```
[kali@kali] ~/Downloads/tools/Sublist3r/sublist3r.py:78: SyntaxWarning: invalid escape sequence '\_'  
$ python sublist3r.py -d yahoo.com > Sublist3r_subs.txt  
  
~/kali/Downloads/tools/Sublist3r/sublist3r.py:78: SyntaxWarning: invalid escape sequence '\_'  
\_ \_\_ \\ |\_| _\| / \| | \| \|\_ \| \_\_  
~/kali/Downloads/tools/Sublist3r/sublist3r.py:286: SyntaxWarning: invalid escape sequence '\_'  
link_regex = re.compile('cite.*?</?.*></?.*>')  
~/kali/Downloads/tools/Sublist3r/sublist3r.py:343: SyntaxWarning: invalid escape sequence '\_'  
link = re.sub('<(\/)?b>', '', link)  
~/kali/Downloads/tools/Sublist3r/sublist3r.py:439: SyntaxWarning: invalid escape sequence '\_'  
Link = re.sub('<(\/)?strong>[open.*?] @ ', '', Link)  
~/kali/Downloads/tools/Sublist3r/sublist3r.py:658: SyntaxWarning: invalid escape sequence '\_'  
tbl_regex = re.compile('a name="hostanalyzer">\w+test Records.<?table.*?(.*)</table?', re.S)  
~/kali/Downloads/tools/Sublist3r/sublist3r.py:898: SyntaxWarning: invalid escape sequence '\_'  
domain_check + re.compile("([httphttps]?[a-zA-Z0-9]{0}(<{[\-\.,]}{1}[a-zA-Z0-9]+)\.\w{.[a-zA-Z]{2,$})")  
Process MMSnapshotter-B:  
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_csrf_token(resp)  
File "/home/kali/Downloads/tools/Sublist3r/sublist3r.py", line 641, in get_csrf_token  
token = csrf_regex.findall(resp)[0]  
          ~~~~~~  
IndexError: list index out of range
```

```

1  [ ] Generating Subdomains for yahoo.com
2  [ ] Searching now in Root -
3  [ ] Searching now in Subnet -
4  [ ] Searching now in Google -
5  [ ] Searching now in Bing -
6  [ ] Searching now in MSN -
7  [ ] Searching now in Netcraft -
8  [ ] Searching now in Whoisoperator -
9  [ ] Searching now in SiteTarget -
10 [ ] Searching now in ThreatCloud -
11 [ ] Searching now in Shodan -
12 [ ] Searching now in PassiveDNS -
13 [ ] Done. Generated probably now in blocking our requests
14
15 Process: Whoisoperator-0:
16
17 THOUGHT: CMD: curl -s http://
18
19 File "/usr/lib/python3.11/site-packages/parsers.py", line 111, in _extract
20     url=url,
21     ...
22
23 File "/home/ali/Downloads/scan/whois/whois.py", line 148, in run
24     domain_list = self.extracturl()
25
26 File "/home/ali/Downloads/scan/whois/whois.py", line 149, in extracturl
27     return self.get_urls(url)
28
29 File "/home/ali/Downloads/scan/whois/whois.py", line 149, in get_urls
30     return self._get_urls(url)
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
9
```

```
(kali@kali) ~$ thewarvester -d yahoo.com -o urlscan --l 200
read proxies from /home/kali/.thewarvester/proxies.txt
*****
the war vester
*****
+ thewarvester v.0.2
+ Coded by Christian Martello
+ Edge-Security Research
+ christian@edge-security.com
*****

[*] target: yahoo.com
[*] searching urllscn.
[*] scan found: IP
-----
681931D
621A81B
62A089F
621A825
621B31B
621B96D
621C711
622A96D
622A7
622C81
622E3
622F24D
622FA92
6267983
626A113
626C54
626F79F
627523
628824

[*] Drivenetwz twls found: 0
```

```
(kali@kali)-[~]  
$ amass enum -passive -d yahoo.com > amassyahoo.txt
```

```

root@kali:~# sudo gmap -mmap yahoo.com
yahoo.com (Pmap) -> mx_record -> mx25.amb.yahoo.com (Pmap)
yahoo.com (Pmap) -> mx_record -> mx26.amb.yahoo.com (Pmap)
yahoo.com (Pmap) -> mx_record -> mx27.amb.yahoo.com (Pmap)
yahoo.com (Pmap) -> mx_record -> mx2.yahoo.com (Pmap)
yahoo.com (Pmap) -> mx_record -> mx4.yahoo.com (Pmap)
yahoo.com (Pmap) -> mx_record -> mx5.yahoo.com (Pmap)
yahoo.com (Pmap) -> mx_record -> mx1.yahoo.com (Pmap)
mx2.yahoo.com (Pmap) -> cname_record -> global-accelerator.dns-rc.amb.yahoo.com (Pmap)
10.10.10.254-p4.gsl.yahoo.com (Pmap) -> cname_record -> 10.10.10.254-p4.gsl.yahoo.com (Pmap)
10.10.10.254-p4.gsl.yahoo.com (Pmap) -> cname_record -> 10.10.10.254-p4.gsl.yahoo.com (Pmap)
search.yahoo.com (Pmap) -> cname_record -> ds-global3.ltr.search.ystg1.b.yahoo.com (Pmap)
shopping.yahoo.com (Pmap) -> cname_record -> oem-lbcl-router.gsl.yahoo.com (Pmap)
msn.search.yahoo.com (Pmap) -> cname_record -> ds-global3.ltr.search.ystg1.b.yahoo.com (Pmap)
search.yahoo.com (Pmap) -> cname_record -> ds-global3.ltr.search.ystg1.b.yahoo.com (Pmap)
sg.finance.yahoo.com (Pmap) -> cname_record -> edge-gypd.b.yahoo.com (Pmap)
bfi.yahoo.com (Pmap) -> mx_record -> mx1.yahoo.com (Pmap)
bfi.yahoo.com (Pmap) -> mx_record -> mx1.yahoo.com (Pmap)
bfi.yahoo.com (Pmap) -> mx_record -> mx1.yahoo.com (Pmap)

```

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 Assignment 1

```
(kali@kali)~[~]
$ chmod 777 parse_dns.sh

(kali@kali)~[~]
$ ./parse_dns.sh
[+] Done! Saved to dns_records.csv
```

```
GNU nano 2.9.2 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"
```

| 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 | atv2-lp.wg1.b.yahoo.com |
| 13 | cname_record | r.search.yahoo.com | ds-globa3.17.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
Assignment 1

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 lps and 19 ASNS while -b urlscan gave me 19 ASNS, 2 URLs, 60 lps and 2 hosts and some resources does not gave me any results



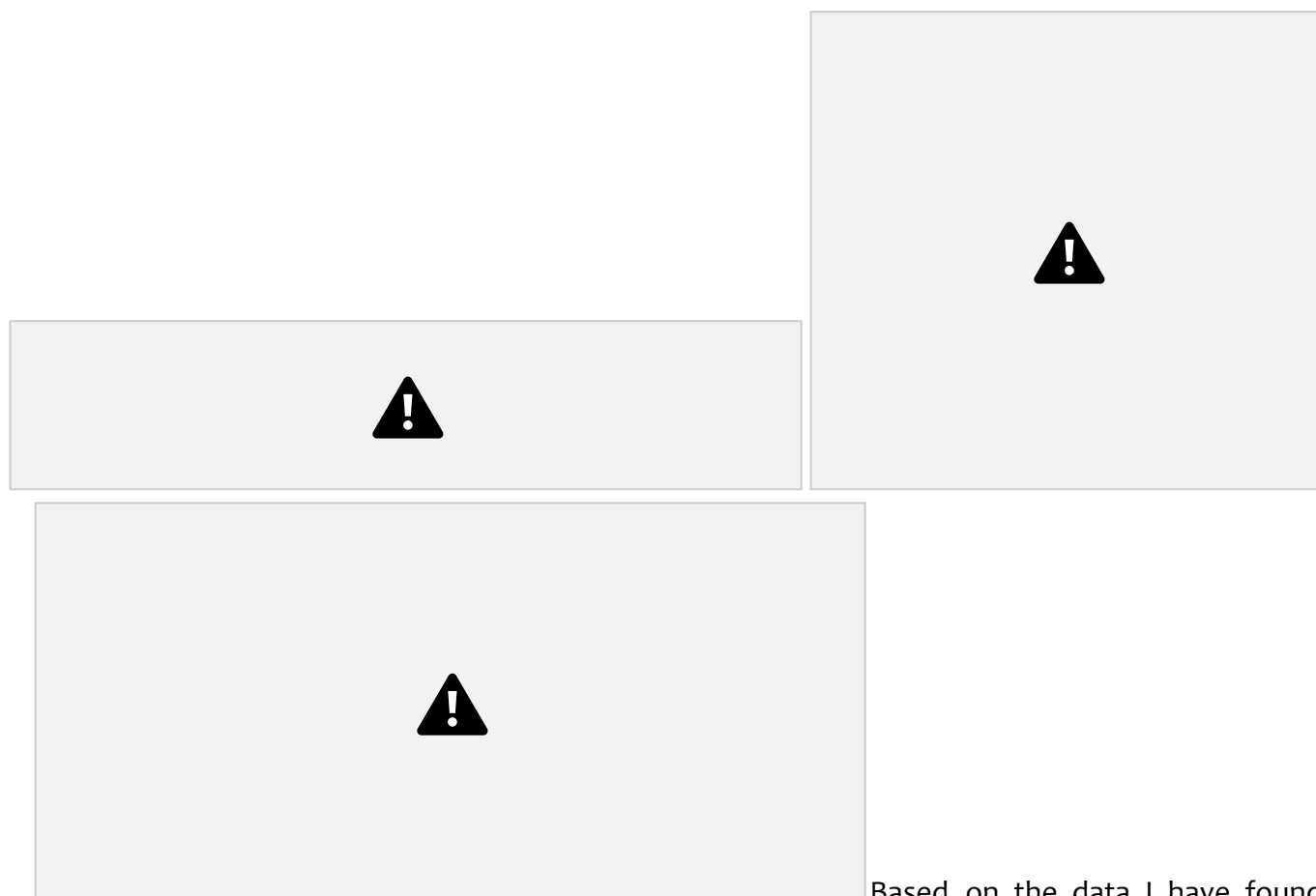
The source engine gave me the most useful results was amass the global default option and theHarvester -b all option gave me wider information and results that could help me more in my upcoming phases, but I thought it will differ based on the target.

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



theHarvester Assignment 1

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.



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.