JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES
www.journalimcms.org

ISSN (Online) : 2454 -7190 Vol.-15, No.-6, June (2020) pp 158-173 ISSN (Print) 0973-8975

COMPARATIVE ANALYSIS OF SUBDOMAIN ENUMERATION TOOLS AND STATIC CODE ANALYSIS

G. Jaspher Kathrine[1], Ronnie T. Baby[2], V. Ebenzer[3]

[1,2,3] Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, Tamil Nadu, India.

[1]kathrine@karunya.edu, [2]ronnietbaby.work@gmail.com,
[3]ebenezerv@karunya.edu

https://doi.org/10.26782/jmcms.2020.06.00013

Abstract

Reconnaissance or footprinting is the technique used for gathering information about computer systems and the entities they belong to. To exploit any system, a hacker might use various tools and technologies. This information is very useful to a hacker who is trying to crack a whole system. Subdomain enumeration plays a vital role in reconnaissance. Enumeration of subdomains provide an important insight towards the various underlying architecture and enable to find hidden user interfaces and admin panels. The less infrequent and unknown the domain name, the less visitors will visit the site. This enables a blindspot for the easy finding of low hanging vulnerabilities. Some of the most popular various tools used for recon on domains are Amass, Subfinder, KnockPy, altdns, sublis3r. We have done a comparative study and analysis of various functions of these tools on parameters like uniqueness, accuracy, complexity and conclude which works in certain scenarios along with static code analysis to find weak spots within the code infrastructure of each of the tools.

Keywords: reconnaissance, web security, application security.

I. Introduction

Reconnaissance is a way of “pre-attack” tactic employed by attackers to gain both passive and active information gathering of large amount of both informational and sensitive data from the target infrastructure. The main objective of recon is to collect data as fast as possible without any detection alarms; but not to exploit the system infrastructure. In web infrastructure, subdomain enumeration plays a major role in finding the unknown underlying and unvisited internal domains, which may have the chance to better, be vulnerable to 0 day attacks. Various tools like Amass, Subfinder, KnockPy, altdns, sublis3r exists to perform subdomain enumeration. All these tools employ various features for gathering information like web scraping, API etc. [V].

A comparative analysis of various usages of these tools and which will be more efficient under certain parameters; along with static code analysis for finding security vulnerabilities.

Classification of Reconnaissance Methods-

Footprinting also known as reconnaissance or recon in short is commonly divided into two common categories viz.

I. Active information Gathering

II. Passive Information Gathering

Both the tools enforce the characteristics of both active and passive recon methods which is discussed below-

I. Active information Gathering

In active information gathering, the attacker directly interacts with the targeted system and try to gain system information and analyse about the software, backend etc. The only drawback of this type of information gathering is that the system owner may get notified about an active information gathering process and could take appropriatemeasure to stop it and prevent future intrusions[VII].

An enumeration tool altdns uses a pattern to form a mutated set of subdomain list and then tries to confirm if the host gets resolved.

II. Passive Information Gathering

Passive information gathering as the name suggests employs more passive techniques and evades the direct contact with the target system infrastructure. Passive information gathering technique greatly reduces the detection of imminent attack coming against its system by the victim[I]. It really does support the saying "*The* ***quieter you become****, the more you are able to hear.*" Passive recon technique is prominently used as a recon method by attackers as it is not always necessary for the presence of targeted systems within reach. Passive footprinting can be done using search engines, Google dorks and other OSINT techniques.

The only major disadvantage of passive recon is that there may not much information gathered in comparisons with active recon. Speed is also a factor in passive recon.

Sublit3r, Amass, sub finder knock primarily use OSINT, bruteforce and DNS zone transfers and likewise techniques for information gathering[II], [XVI].

II. Subdomain Enumeration

As already discussed, recon is the most common methodology used to analyse and map the potential targets of the victim's system. Since most systems are connected to the external world via internet though respective web infrastructuresystems, it become very important to analyse the web assets. One of the prominent ways to run recon on web assets is by enforcing subdomain enumeration techniques.

Subdomain enumeration allows to find and enumerate a list of subdomains other than the main domain of any organisation. Usually, the main domain of any infrastructureis highlysecured and would always adhere to latest securitystandards and patches [XV]. However it may not be a just a blatant guess to suggest that all other subdomains may not follow the strict security protocols set for the main domain.

Subdomain enumeration technique involves the finding of huge list os subdomains of a main domain. Subdomains tend to be more vulnerableto security and information disclosure issues. Since the amount of visitors to subdomains are less compared to the main domain, it usually gets ignored of having the same prioritised attention compared with the main domain.

An example of domain: google.com

Some examples of subdomains of the main domain of google.com are:

images.google.com

drive.google.com

accounts.google.com

mail.google.com

forms.google.com and so on.

Subdomains can go one step more deeper within a subdomain’s context.

Like: *. *. google.com

We will be analysing 5 open source tools and do an analysis of theresults and also conduct basic static code analysis on the codebase .

For the generation of an equal playground of comparison, we will be analysing and enumerating the subdomains of the domain yahoo.com and run all the 5 tools on it.

Selecting yahoo.com as the domain for analysis can be justified by the huge list of subdomains (>1000) available under its main domain. Yahoo has been a old player in the internet industry and once was regarded as the pioneer and go to search engines; before finally being replaced by Google.

The TLD (top level domain) of Yahoo is yahoo.com and it’s a number of assets under its many subdomains. In fact, there is a huge probabilityof finding bugs in subdomains of yahoo and it is highlypopular among security researchers.

III. Tools

Each of the five tools has several common as well as distinct methods for enumeration of the subdomains. We will focus on the generation of the results from the very abstract form and usage of each of them without using integrated devices within including the usage of API’s of third party sources. The tools may enforce both active and passive recon methods for finding the subdomain assets.

At the end, we evaluate all the five tools on different usability parameters.

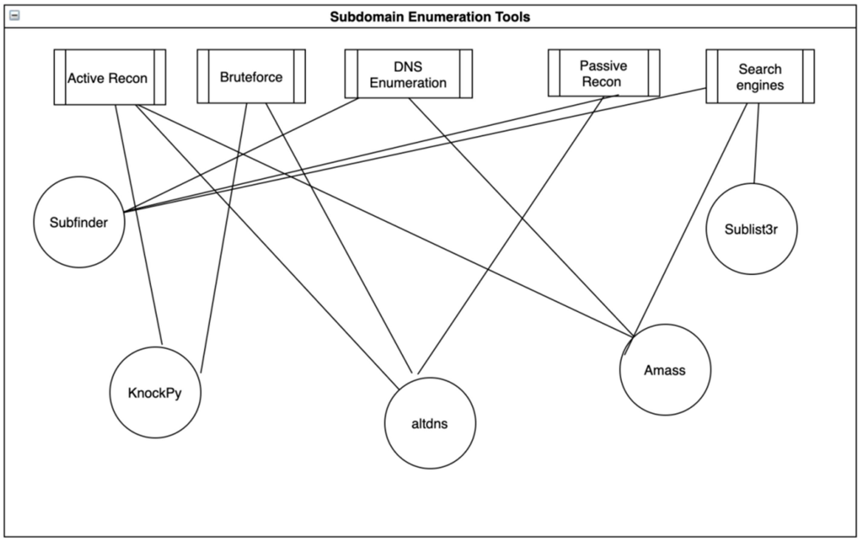


Fig. 1: Architectural Diagram of all 5 subdomain enumeration tools

At the end, we evaluate all the five tools on different usability parameters.

Sublister

Sublis3r is coded in python and mainly takes use of OSINT techniques to enumerate the subdomain list via search engines like google, baidu, bingetc[IV]. It has been also integrated with another tool sub rock to find subdomains using brute force from a list of wordlists.

Usage Commands

Basic usage-

python sublist3r.py -d yahoo.com

To find specific ports

python sublist3r.py -d yahoo.com -p 80,443

To enumerate subdomains and enable the brute force module:

python sublist3r.py -b -d yahoo.com

Amass

Amass is an asset discovery tool which utilises open source information gathering along with active recon methods to enumerate the subdomain list[VIII]. It is completely written in Golang.

Amass is maintained by The Open Web Application Security Project (OWASP) which is a non-profit foundation for improving the security of software.

Recon techniques used are-

Web scraping - Google, Yahoo, Reddit, Baidu etc.

Dns- zone transfer, brute force methods, reverse dns sweeping,

Certificates= censys,certspotter, Google CT

APIs: AlienVault, BinaryEdge, BufferOver, CIRCL, CommonCrawl, DNSDB, GitHub, HackerTarget, NetworksDB, PassiveTotal, Pastebin..

Web Archives: ArchiveIt, ArchiveToday, Arquivo, Wayback and others.

Sample Usage Commands:

Basic usage-

amassenum -d yahoo.com

The command line usage of Amass consists of various subcommands with its respective arguments.

Intel Subcommand

This command uses passive intelligence techniques to find other root domains associatedwiththe same organisation.

ex. *amass intel -active -addr 192.168.2.1-64 -p 80,443,8080*

'enum' Subcommand

It performs DNS enumeration and network mapping to populate the selected graph database.

ex*amassenum -active -d example.com -p 80,443,8080*

'viz' Subcommand

It helps for visualizations and adds a proper structure to the collected information.

ex. *amass viz -maltego -d example.com*

'track' Subcommand

It shows the differences between enumerations of the same target.

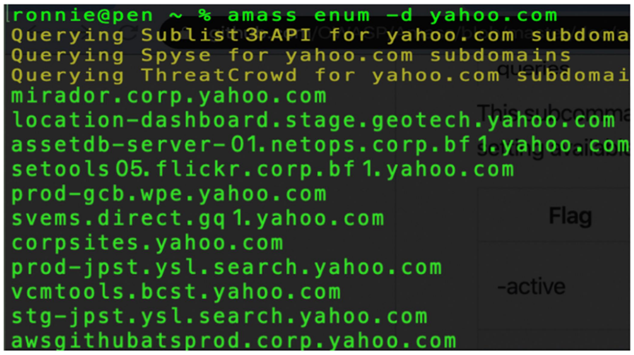
ex. *amass track -history*

'db' Subcommand

It helps for the manipulation of the graph database.

ex. *amassdb -show*

All the enumeration data gets stored in the graph database which gets separated in each new execution.It uses the format known as Cayley Graph Schema for storing the details of the domains.



```
ronnie@pen ~ % amass enum -d yahoo.com
Querying Sublist3rAPI for yahoo.com subdoma
Querying Spyse for yahoo.com subdomains
Querying ThreatCrowd for yahoo.com subdomai
mirador.corp.yahoo.com
location-dashboard.stage.geotech.yahoo.com
assetdb-server-01.netops.corp.bf1.yahoo.com
setools05.flickr.corp.bf1.yahoo.com
prod-gcb.wpe.yahoo.com
svems.direct.gq1.yahoo.com
corpsites.yahoo.com
prod-jpst.ysl.search.yahoo.com
vcmtools.bcst.yahoo.com
stg-jpst.ysl.search.yahoo.com
awsgithubatsprod.corp.yahoo.com
```

Fig.2: BasicEnum command for Amass Tool

Subfinder

Subfinder is a subdomain enumeration tool which uses passive information gathering techniques to gather data. It is written in Go lang for effective speed optimization. It is highly efficient to gather passive recon data and is actively maintained by Project Discovery[IX]. The main claim of sub finder as an effective enumeration tool is its optimisation towards the speed.

Some features of Subfinder include-

I. modular code structure.

II. Integrated with fast wildcard elimination module.

III. Multiple output formats like json , file and stdout

IV. Huge list of passive sources.

V. Helps in integrated workflows via stdin and stdout.

Subfinder works with third party sources like shodan, census, virustotal which will not be discussed as such here as they need api keys. Further, it also supports running as a container under Docker.

Some sample usage commands-

Basic usage-

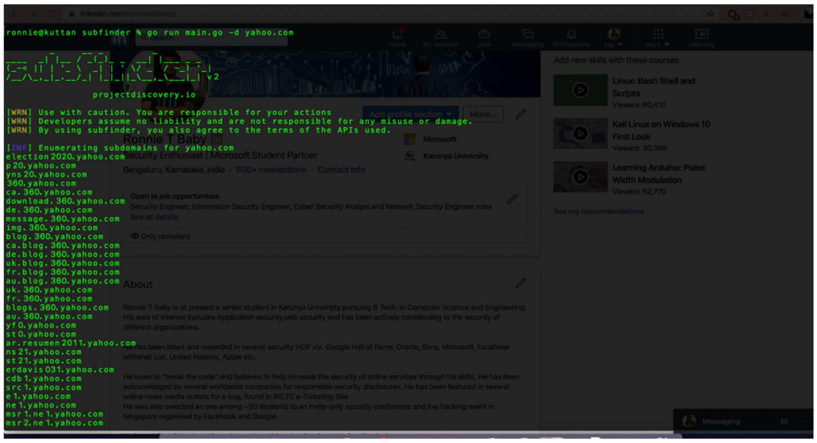
subfinder -d yahoo.com

Domain list for Enumeration:

subfinder -dL yahoo-hosts.txt

Change the Number of Concurrent Go Routines:

subfinder -t 100



```
ronnie@kuttan subfinder % go run main.go -d yahoo.com

                                        v2
              projectdiscovery.io

[WRN] Use with caution. You are responsible for your actions
[WRN] Developers assume no liability and are not responsible for any misuse or damage.
[WRN] By using subfinder, you also agree to the terms of the APIs used.

[INF] Enumerating subdomains for yahoo.com
election2020.yahoo.com
p20.yahoo.com
yns20.yahoo.com
360.yahoo.com
ca.360.yahoo.com
download.360.yahoo.com
de.360.yahoo.com
message.360.yahoo.com
img.360.yahoo.com
blog.360.yahoo.com
ca.blog.360.yahoo.com
de.blog.360.yahoo.com
uk.blog.360.yahoo.com
fr.blog.360.yahoo.com
au.blog.360.yahoo.com
uk.360.yahoo.com
fr.360.yahoo.com
blogs.360.yahoo.com
au.360.yahoo.com
yf0.yahoo.com
st0.yahoo.com
ar.resumen2011.yahoo.com
ns21.yahoo.com
st21.yahoo.com
erdavis031.yahoo.com
cdb1.yahoo.com
src1.yahoo.com
e1.yahoo.com
ne1.yahoo.com
msr1.ne1.yahoo.com
msr2.ne1.yahoo.com
```

Fig. 3: Basicenum command for subfinder tool

Knockpy

Knockpy is written in python and run subdomain scan from a list of wordlists. It also supports scanning for DNS zone transfer and has the capability to bypass wildcard DNS record if enabled. It also supports queries via VirusTotal using its api keys[X]. It is currentlymaintained by Gianni Amato (IT Security Researcher).

Sample Usage Commands-

Scan with internal wordlist

knockpyyahoo.com

Scan with external wordlist

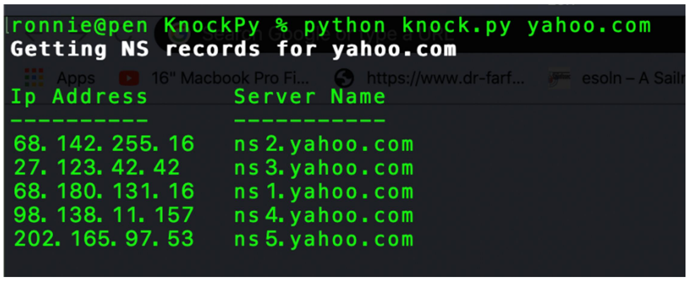
knockpy yahoo.com -w wordlist.txt

Get response headers while resaving domains

knockpy -r yahoo.com

Export report in JSON-

knockpy -j yahoo.com



```
ronnie@pen KnockPy % python knock.py yahoo.com
Getting NS records for yahoo.com

Ip Address          Server Name
----------          -----------
68.142.255.16       ns2.yahoo.com
27.123.42.42        ns3.yahoo.com
68.180.131.16       ns1.yahoo.com
98.138.11.157       ns4.yahoo.com
202.165.97.53       ns5.yahoo.com
```

Fig.4: Basicenum command for Knockpy tool

Altdns

Altdns is a totally different subdomain discovery tool when compared all other tools as it uses a combo of permutations and alterations and enforces a certain standard for pattern generation to form a huge list of mutated subdomain list. This mutated list is then analysed to find the number of resolving hosts via dns brute forcing techniques[XI]. It supports multi-threaded subdomain resolves and is coded in python. It is maintained by security researcher @infosec-au.

The only disadvantage of Altdns as an efficient tool is its stringent requirement for a huge list of initial dataset (~200) containing already known subdomain list. The more initial data we have, the more efficient will be the list of altered subdomains.

Sample Usage Commands-

altdns -i subdomains.txt -o data_output -w words.txt -r -s results_output.txt

where subdomains.txt contains the known subdomains

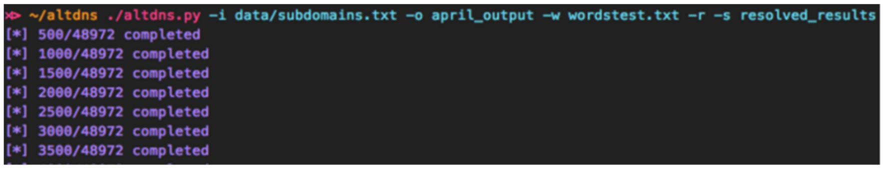
data_output contains list of altered subdomains.

words.txt contains list of words for permutations

-r tries to resolve the mutated subdomain

-s location for the saved output.

-t number of threads



```
⋗ ~/altdns ./altdns.py -i data/subdomains.txt -o april_output -w wordstest.txt -r -s resolved_results
[*] 500/48972 completed
[*] 1000/48972 completed
[*] 1500/48972 completed
[*] 2000/48972 completed
[*] 2500/48972 completed
[*] 3000/48972 completed
[*] 3500/48972 completed
```

Fig. 5: Basicenum command for altdns tool

IV. Results

Alternative Subdomain Enumeration Way

Apart from these 5 tools, we can also scrape and find the subdomains using newly issued TLS Logs.

Crt.sh has available database to make this possible. Some of the tools have already integrated the crt.sh functionality within them. Certificate transparency is an internet security standard for the monitoring and auditing of digital certificates[XII]. Idea of creating crt.sh started after Comodo was attacked and CA Diginotar was compromised in the year 2011.

We have written a Python script to retrieve the subdomains by scraping the results from crt.sh is-

```
#!/usr/bin/env python3
import sys
importurllib.request
importurllib.parse
import re
iflen(sys.argv) == 1:
    print("Usage: " + sys.argv[0] + " [domain] ...")
    sys.exit(1)
for i, arg in enumerate(sys.argv, 1):
    domains = set()
    withurllib.request.urlopen('https://crt.sh/?q=' + urllib.parse.quote('%.' + arg)) as r:
        code = r.read().decode('utf-8')
        for cert, domain in re.findall('<tr>(?:\s|\S)*?href="\?id=([0-9]+?)"(?:\s|\S)*?<td>([*_a-zA-Z0-9.-]+?\.' + re.escape(arg) + ')</td>(?:\s|\S)*?</tr>', code, re.IGNORECASE):
            domain = domain.split('@')[-1]
            if not domain in domains:
                domains.add(domain)
                print(domain)
```

Since its not feasible to list all the subdomains generated by each tool, we have listed it here for easy access [VI].Please do note that that the results may differ for others having varied system configuration and internet speed.

The configuration used here are-

Internet Speed- 300 kbps/sec

System- MacBook Pro 16 2019 16 GB ram

Time:

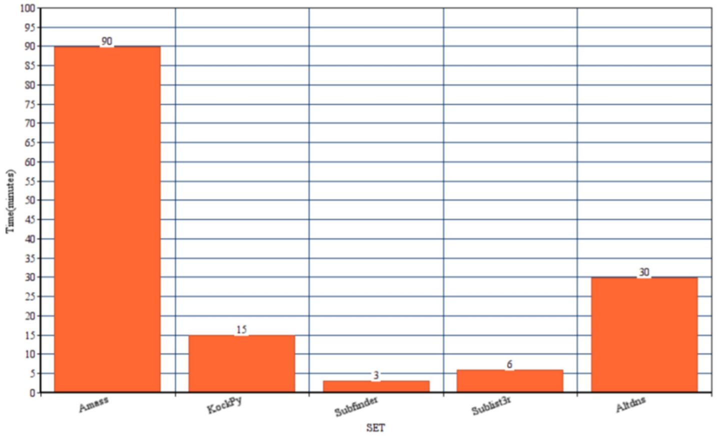


Fig. 6: Time Analysis

As we observe, we ran tests with the basic usage command of each of the 5 tool and got the aforementioned results.

Amass took a total of 90 minutes for the results to end showing and Subfinder was the fastest among all. As discussed above, Subfinder main advantage was its optimization for speed and it does well to stand up to the expectations.

From the least time taken-

Subfinder< Sublist3r <Knockpy<Altdns< Amass

Features:

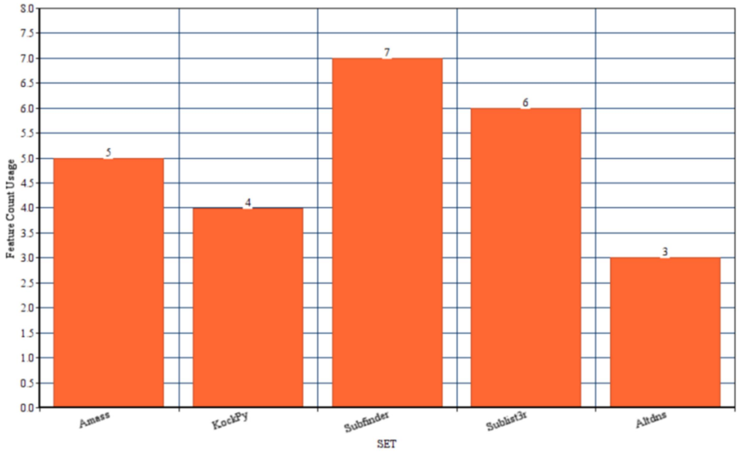


Fig. 7: Feature Analysis

It is debatable to compare to compare and analyse each of its features as each of these

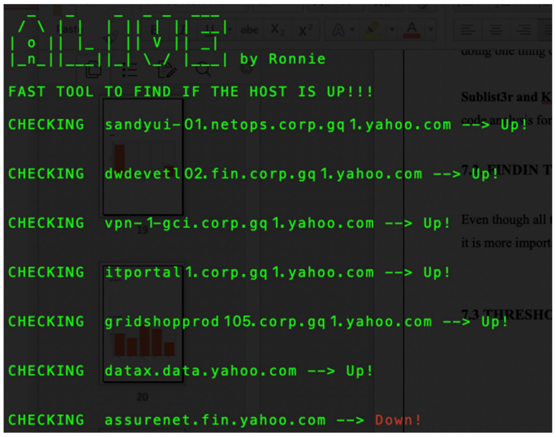
5 tools work well in certain conditions and situations.

Again in terms of feature count and advanced usage techniques, subfinder is the frontrunner with ~7 and altdns has the least features. Having said that, altdns performs well in its own fact that it does a job of mutating the subdomains, which is not included in any of the other 4 tools.

Accuracy

Since each of the five subdomain enumeration tools generate a large set of results and output a big list of subdomains, it becomes important to find the accuracy of the results and determine if the hosts are getting resolved or not.

To fulfil the requirement, we have written a tool; “Alive” in python which takes a list of subdomains and determines of the subdomain is actually resolving at present.



```
  _   _       _   _ _   ___
 / \ | |     | | | | | |_ _|
| o || |_   | | | V | | _|
|_n_||___|  |_|  \_/  |___| by Ronnie

FAST TOOL TO FIND IF THE HOST IS UP!!!

CHECKING  sandyui-01.netops.corp.gq1.yahoo.com --> Up!

CHECKING  dwdevetl02.fin.corp.gq1.yahoo.com --> Up!

CHECKING  vpn-1-gci.corp.gq1.yahoo.com --> Up!

CHECKING  itportal1.corp.gq1.yahoo.com --> Up!

CHECKING  gridshopprod105.corp.gq1.yahoo.com --> Up!

CHECKING  datax.data.yahoo.com --> Up!

CHECKING  assurenet.fin.yahoo.com --> Down!
```

Fig. 8: Tool to find UP hosts

Here is a python script developed to check if the resultant domains are getting resolved.

```
import socket

def main():

  f=open("read.txt","r")

count=0

  f1=f.readlines()

forsd in f1:

aas=sd.strip()

print("CHECKING  "+ aas+" --> ",end='')

try:

        w=socket.gethostbyname(aas)
```

```
print("Up!",sep='')
count=count+1
exceptsocket.error:
print("\033[31m" + "Down!"+"\033[0m",sep='')
print("\n")
print("\n Total hosts FOUND to be UP ARE :  "+str(count))
if __name__=="__main__" :main()
```

Mapping accuracy is important because some tools use OSINT and web scraping to find the subdomains, but it is not necessary, that they remain active now.

Hosts which are up:

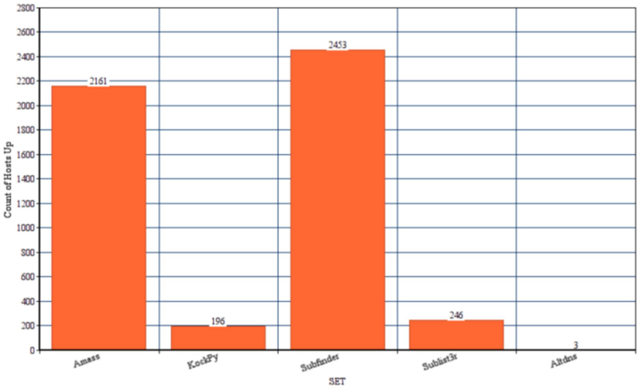


Fig. 9: Up Host result from Alive Tool

Again, we find that Subfinder is the frontrunner with 2453 alive hosts.

Amass follows closely with around 2161 and the least no. of resolving hosts is by Altdns.

Altdns uses a set of permutations to guess a new subdomain along with the combination of wordlist which makes it difficult to have more number of live hosts when compared to others and it also requires a big initial list of known subdomains; which we have compiled from the script used to scrape data from crt.sh.

V. Static Code Analysis

Code Analysis can be done in Two Ways

I. Dynamic Code Analysis

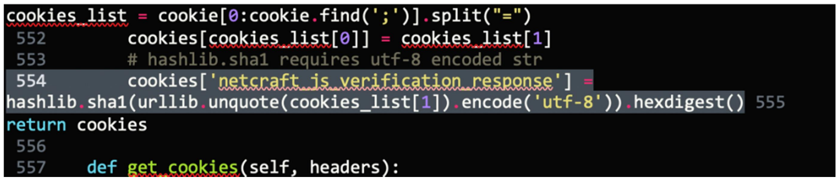
II. Static Code Analysis

Static code analysis involves the debugging of a computer application without executing the program, and only by just examining the code structure and detecting flaws[XIV], [III]. Static code analysis has several advantages like its efficient detecting of the flaw in code in the exact spot, fast running in case automated tools are used. It also is faster. It has started to play a crucial role in Devops and quality assurance early in the software product lifecycle.

We have done static code analysis on the tools and have concluded to give only certain security recommendations about the code quality[XIII]. Given are some of the several security risks associated with each of the five tools.

Sublist3r

Insecure hash function

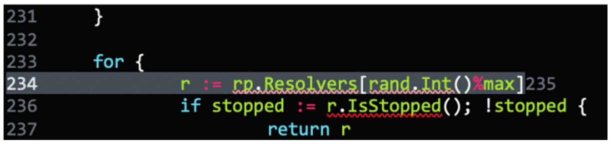


```
cookies_list = cookie[0:cookie.find(';')].split("=")
        cookies[cookies_list[0]] = cookies_list[1]
        # hashlib.sha1 requires utf-8 encoded str
        cookies['netcraft_js_verification_response'] =
hashlib.sha1(urllib.unquote(cookies_list[1]).encode('utf-8')).hexdigest()
return cookies

    def get_cookies(self, headers):
```

Fig. 10: Usage of insecure hash function

SHA1 signature algorithms are known to be vulnerable to collision attacks. Attackers can exploit this to generate another certificate with the same digital signature, allowing them to masquerade as the affected service.

Amass



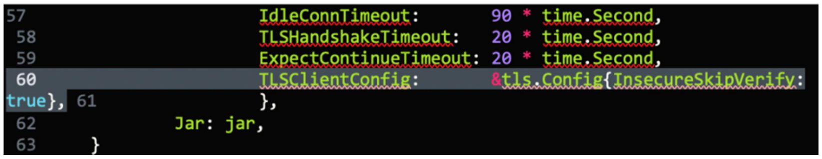
```
    }

    for {
            r := rp.Resolvers[rand.Int()%max]
            if stopped := r.IsStopped(); !stopped {
                    return r
```

Fig. 11: Random number generation source

math/rand is much faster for applications that don't need crypto-level or security-related random data generation. crypto/rand is suited for secure and crypto-ready usage, but it's slower.

It is highly recommended to use crypto/rand when needing to be secure with random numbers such as generating session ID in a web application.

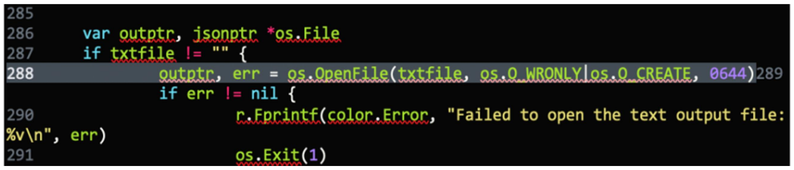
Bad TLS connection settings



```
                        IdleConnTimeout:       90 * time.Second,
                        TLSHandshakeTimeout:   20 * time.Second,
                        ExpectContinueTimeout: 20 * time.Second,
                        TLSClientConfig:       &tls.Config{InsecureSkipVerify:
true},
                        },
                Jar: jar,
        }
```

Fig. 12: InsecureSkipVerify set to true in TLS config.

Poor file permissions used when creation file or using chmod



```
    var outptr, jsonptr *os.File
    if txtfile != "" {
            outptr, err = os.OpenFile(txtfile, os.O_WRONLY|os.O_CREATE, 0644)
            if err != nil {
                    r.Fprintf(color.Error, "Failed to open the text output file: %v\n", err)
                    os.Exit(1)
```

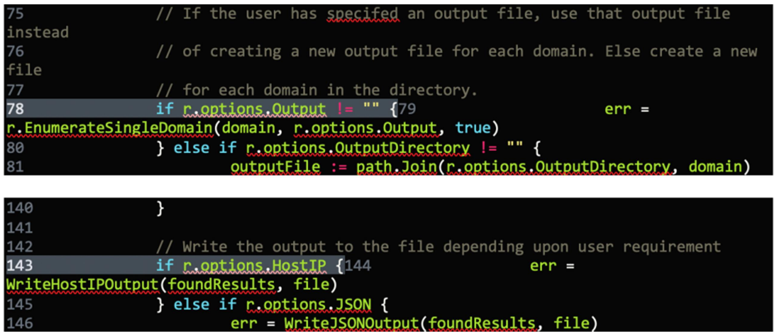
Fig. 13: Misconfigured file permissions

Excessive permissions granted to a file/directory. This warning is triggered whenever permission greater than 0600 is granted.

Subfinder

There was no security risk found while checking code of subfinder; which is a good thing given that it has performed the best among all other tools in all other parameters.

Some recommended performance issues-



```
        // If the user has specifed an output file, use that output file instead
        // of creating a new output file for each domain. Else create a new file
        // for each domain in the directory.
        if r.options.Output != "" {
                err = r.EnumerateSingleDomain(domain, r.options.Output, true)
        } else if r.options.OutputDirectory != "" {
                outputFile := path.Join(r.options.OutputDirectory, domain)
```

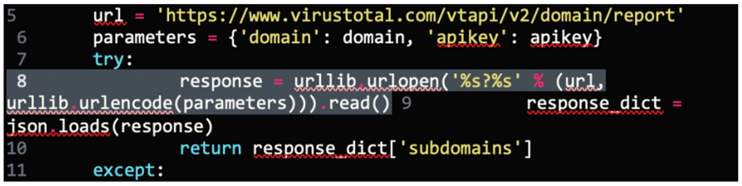
```
        }

        // Write the output to the file depending upon user requirement
        if r.options.HostIP {
                err = WriteHostIPOutput(foundResults, file)
        } else if r.options.JSON {
                err = WriteJSONOutput(foundResults, file)
```

Fig. 14: Performance issues

Repeated if-else statements should be replaced with switch

Knockpy

Use of an insecure method method from urllib



```
    url = 'https://www.virustotal.com/vtapi/v2/domain/report'
    parameters = {'domain': domain, 'apikey': apikey}
    try:
        response = urllib.urlopen('%s?%s' % (url, urllib.urlencode(parameters))).read()
        response_dict = json.loads(response)
        return response_dict['subdomains']
    except:
```

Fig. 15: Insecure urllib method

urllib not only opens http:// or https:// URLs, but also ftp:// and file://. With this it might be possible to open local files on the executing machine which might be a security risk if the URL to open can be manipulated by an external user.

Altdns

No security risk was detected in its codebase. However a few performance issues like unused variable, unnecessary calls, and if-else statements were found.

VI. Conclusion

Reconnaissance plays a prominent role in finding both passive and active information about a given target. Some various recon methodologies and techniques used in it are DNS queries, WHOIS queries, operating system identification. Recon maybe done either in active and passive ways. Passive footprinting gathers data by innocuous means and may not easily be detected. Active footprinting uses feature like traceroute which directly pings onto the target infrastructure. The only disadvantage in this is it has higher chance of triggering IDS systems. All 5 of the subdomain enumeration research tools use both active and passive recon methods to gather information about the subdomains.

Doing a proper comparative study between the different recon tools helps to identify which works best under certain parameters like accuracy, time, features, ease of use etc. We did static code analysis and further results show that subfinder has the best codebase among all and is more efficient and speed optimised.

Static code analysis on these recon tools helped to find the embedded weakness in the codebase and propose standards for improvement in the future.

References

I. A. Kothia, B. Swar and F. Jaafar, "Knowledge Extraction and Integration for Information Gathering in Penetration Testing," 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), Sofia, Bulgaria, 2019, pp. 330-335.

II. Adiwal, Sanjay &Rajendran, Balaji&Shetty, Pushparaj.(2018). Domain Name System (DNS) Security: Attacks Identification and Protection Methods.

III. AlkaAgrawal, MamdouhAlenezi, Rajeev Kumar and Raees Ahmad Khan, Securing Web Applications through a Framework of Source Code Analysis, Journal of Computer Science,Volume 15, Issue 12,Pages 1780-1794

IV. https://github.com/aboul3la/Sublist3r

V. https://github.com/guelfoweb/knock

VI. https://gitlab.com/paperrepo/subdomain-enumeratioon

VII. https://github.com/infosec-au/altdns

VIII. https://github.com/OWASP/Amass

IX. https://github.com/projectdiscovery/subfinder

X. K. Nirmal, B. Janet And R. Kumar, "Web Application Vulnerabilities- The Hacker's Treasure," 2018 International Conference On Inventive Research In Computing Applications (Icirca), Coimbatore, India, 2018, Pp. 58-62

XI. P. Harika Reddy SurapaneniGopi Siva SaiTeja,Cyber Security and Ethical Hacking,International Journal for Research in Applied Science & Engineering Technology (IJRASET),Volume 6 Issue VI, June 2018

XII. Richard Roberts and Dave Levin. 2019. When Certificate Transparency Is Too Transparent: Analyzing Information Leakage in HTTPS Domain Names. In Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society (WPES'19).Association for Computing Machinery, New York, NY, USA, 87–92.

XIII. Russell, Rebecca & Kim, Louis & Hamilton, Lei &Lazovich, Tomo&Harer, Jacob &Ozdemir, Onur&Ellingwood, Paul &McConley, Marc. (2018). Automated Vulnerability Detection in Source Code Using Deep Representation Learning. 757-762. 10.1109/ICMLA.2018.00120.

XIV. S. M. Zia Ur Rashid ImtiazKamrulImtiazKamrulAsrafulAlamAsrafulAlam,Understanding the Security Threats of Esoteric Subdomain Takeover and Prevention Scheme, Conference: 2019 International Conference on Electrical,doi: 10.1109/ECACE.2019.8679122

XV. Siavvas M., Gelenbe E., Kehagias D., Tzovaras D. (2018) Static Analysis-Based Approaches for Secure Software Development. In: Gelenbe E. et al. (eds) Security in Computer and Information Sciences. Euro-CYBERSEC 2018.Communications in Computer and Information Science, vol 821. Springer, Cham

XVI. Thomassen, P., Benninger, J., &Margraf, M. (2018).Hijacking DNS Subdomains via Subzone Registration: A Case for Signed Zones. OJWT, 5, 6-13.