CodeNinja a.k.a. Aakash Choudhary

Home About Vulnerability Newsletter Welcome to my website

Bug-Hunting-Day-7

Apr 2, 2019

Bug Hunting Day 7

Today, I am continue from last post all about Subdomain Enumeration and tools and analyzing them.

Topic on Subfinder , Altdns and Massdns from https://poc-server.com/blog/2019/01/18/advanced-recon-subdomains/

- First we bruteforce subdomains with massdns using one or more custom subdomain wordlists.
- Then we run subfinder to get subdomains from various sources.
- When thats done we run altdns to see if we can get some altered versions of the subdomains.
- And finally we add recursion to get subdomains which are more levels deep.

Great

1. Massdns

I didn’t understand much what he doing there So, I google more to clear my doubts and confusion

After google I came to know jhaddix wordlist “all.txt” with massdns for subdomain bruteforce recommended https://twitter.com/aaronhnatiw/status/974682458561097728

By @_alicelebi

He said -> combination of subbrute+massdns gave incorrect result But Jhaddix replied -> Probably a bad resolver, identify it and remove it or grep out all its output

So, we need a better resolver.txt file

Detection wise and Time wise massdns is best bit problem mostly faced in bad resolver

I googled about it and got https://n0where.net/high-performance-dns-stub-resolver-massdns

The repository includes the file resolvers.txt consisting of a filtered subset of the resolvers provided by the subbrute project. Please note that the usage of MassDNS may cause a significant load on the used resolvers and result in abuse complaints being sent to your ISP. Also note that the provided resolvers are not guaranteed to be trustworthy. If you detect a bad resolver that is still included within MassDNS, please file an issue

:O Shit

I now read this ->http://offsecbyautomation.com/Use-MassDNS/

He said in Final Thoughts -> `Overall, to get use out of the information MassDNS provides, you have to write a script to parse it and interact with the output. In my opinion this is the biggest barrier for everyone to use MassDNS. With this said, if you know enough programming to parse the output and relay into your automation, you will have a great subdomain enumeration process. Try it out and compare to your previous enumeration methods, think in terms of results, reliability, and speed.`

Hey but after read this -> https://medium.com/@europa_/recoinnassance-7840824b9ef2

I learned many things from there

``` python ./bin/subbrute.py ./bin/lists/jhaddix.txt ${domains} \ | massdns -r ./bin/lists/resolvers.txt -t A –verify-ip -o S \ | cut -f1 -d’ ‘ \ | sed -r "s/.(${domains// /|}).//g" \ > massdns.out

```
As far as fake results go, instead of blindly trusting what the public resolvers tell me I pref
er to parse the returned results, strip the main domain away, and prepare a sub-wordlist with a
ll the returned entries, to be subsequently fed to Aquatone’s dictionary module.

So he used aquatone after get result of massdns.out
```

# retrieve the resolved entries from the remote droplet
```

scp … … massdns.out .

add cloudflare results to the list

grep –no-filename -E ‘,(A|AAAA|CNAME),’ *.csv \ | cut -f1 -d’,’ \ | rev \ | cut -f3- -d’.’ \ | rev \ | sort -u \ » massdns.out

add fdns results to the list

rev fdns.lst \ | cut -f3- -d’.’ \ | rev \ | grep -Fv ‘*’ \ | sort -u \ » massdns.out

run aquatone-discover on each target

sort -u massdns.out \ | awk NF \ > x && \ mv x massdns.out

run locally

aquatone-discover \ –ignore-private \ –wordlist=massdns.out \ -d target target …

~~~~
Cloudflare result from https://github.com/appsecco/bugcrowd-levelup-subdomain-enumeration/blob/master/cloudflare_enum.py

Nice

2. Subfinder

```subfinder -d $domain -nW -o "subfinder-online.txt" -rL ~/wordlists/resolvers.txt > /dev/null 2>&1```

Massdns + Subfinder -> ```cat wordlist-online.txt subfinder-online.txt > subdomains.txt```

Making the file unique -> ```sort -u "subdomains.txt" -o "subdomains.txt"
~~~~

1. Altdns

Altdns alters the subdomains with a list of given words. For example it can find staging-dev.poc-server.com if you give it dev.poc-server.com.

Hmm interesting

Using this technique we can discover subdomains others wouldn’t have found.

```
python ~/tools/altdns/altdns.py -i "subdomains.txt" -o "altdns-wordlist.txt" -w ~/tools/altdns/words.txt
```

Things to remember There are somethings that are not in this post, and I hope you will find them on your own as well, so you can learn from them.

A couple of examples are:

```
- There might be a wildcard for subdomains. This means that you would get a huge list of false
 positives. Below this list I have inserted a code snippet to detect wildcards.
- Giving your own resolvers list to your tools might increase the speed.
- Clean your -online.txt files when you have all output in one big file.
- Give the script some extra parameters like -verbose to toggle the verbose version of masscan,
 to keep track of the resolved items and how long it will still take.
- FDNS
```

```
if [[ "$(dig @1.1.1.1 A,CNAME {test321123,testingforwildcard,plsdontgimmearesult}.$domain +short | wc -l)" -gt "1" ]]; then echo "[!] Possible wildcard detected." fi
```

Lots of cofusion still in using that specially for beginner like me

Lets now move to Jhaddix methodologies

1. amass.sh ->

```
mkdir $1
touch $1/$1.txt
amass -active -d $1 |tee /root/tools/amass/$1/$1.txt
```

1. subfinder.sh ->

```
mkdir $1
touch $1/$1.txt
```

```
subfinder -d $1 |tee /root/tools/subfinder/$1/$1.txt
```

1. Dont Use Anymore

   Enumall / Recon-NG (Not great on sources or speed) Aquatone (Not Great on Sources) but Aquatone-scan is useful Sublist3r (Same as Above) Anything else for scraping Cloudflare enum (Although sometimes I Think about it)

   - https://github.com/mandatoryprogrammer/cloudflar_enum
2. massdns is best Jhaddix said
3. Eyewitness

Really so much so much information and guide for beginners and is it really helpful for beginners ?

In Infosec Community KISS [Keep It Simple Stupid/Stuff] is really good Advice needing

So i have read many sources and even talked some bug hunters on this topic and conclude that they are follow the KISS thing that is they are doing recon in simple and easy manner not confusion type

Like some using amass+subfinder+altdns sublister+knockpy+aquatone subbrute+amass+aquatone amass+aquatone amass+eyewitness massdns+amass+aquatone/eyewitness

So there are many combination of like this

Also some using their own personal stuffs

So what is common in all of that

Amass [Quick] but sometime not recommended and sometime yes. Pouplarity is high on this tool Subfinder successor of sublist3r Aquatone is too mixely recommended Massdns+amass+eyewitness/aquatone+subfinder [Much highely recommended specially massdns]

But Massdns having problem of resolver issue and false assumption

Some adviced to use our own

Hey but i seen other githubs which using all of these as a combination Like -> Domained+chomp-scan+003Recon+Osmedus+Blackbird

Playing with them is good and then make own then much will be good for us.

But for beginner this is not recommend, First learn to use tools manually and then in free time play with above github repos and make own tool.

So, I Finally make my own Methodology of using such all tools ,hey not all but some with personal choice or can say like that ;)

My Own Methodology with Guide

1. First step i think which is very fast is just use GOOGLE DORKS and grab subdomains using that and start testing on those subdomains which got from Google Dorks

Also shodan+Bing+DuckDuckGo [Search Engines] better alongwith Google to start with searching of subdomains

Also while doing this step follow upcoming steps beside this number 1 step so no time waste and can start testing

Why?

Suppose I am familiar with SQLI or SSRF or File Upload Testing or Open Redirection or XSS/CSRF etc etc

So the fastest way of testing a target domain is

- ** site:domain.com filetype:pdf allinurl:upload -www [like that for exclude subdomains]
- ** site:domain.com ?id=

So, what the result come from that step directly testing of vulnerability of our own choice of testing

So, Browser+Burp is enough for such quick easy steps

1. Second step is subdomain scanning [scrape+bruteforce]

Here are top commands :->

- amass -active -v -d target.com -o /path/to/save/result.txt
- subfinder -d target.com -b -w /path/to/wordlist.txt -v -o /path/to/save/file
- Sublister -b -d target.com -v -t 20 -p 21,22,443,80,8080 -o /path/to/save/file
- aquatone-discover -d target.com [or can use latest aquatoe which have written in go language]
- Eyewitness -f subdomainlist.txt -t 15 -d /directory/name/to/save/output/from/eyewitness/result –web
  - Best helper guide -> https://www.christophertruncer.com/eyewitness-usage-guide/
  - https://github.com/FortyNorthSecurity/EyeWitness/issues/310
  - also can use options -> –prepend-https and User-Agent options

- webscreenshot -i subdomainlit.txt -o /save/output/target-name-folder -v -m [http & https]
-

```
cat domains.txt    httprobe > live.txt; webscreenshot -i live.txt
```

Note: For starting i don’t want to play with massdns so i think this is enough for starting for beginners And for usage of commands check my earlier blog post Day 6 And didn’t use altdns+goaltdns till now but it can be usefull if trying this too

Give it a try too when having time gograbber[https://github.com/swarley7/gograbber] [scan to portscan, dirbust for directory,screenshot]

Can use subdomain takeover tools now if wish to use like subzy,tko-subover,aquatone-takeover But yeah this time i don’t want to use it or can be use side by side ;)

1. Google Dorks: Done, Subdomain bruteforcing:done, screenshot:done, subdomain takeover:partially done Now left is directory brutefocing, link finding, parameter finding

When choosing subdomain target from Step 2 or even in Step 1 try to find parameters and all links,hidden links and all of these in side by side i.e from quicker testing to time taking step testing

For directory bruteforcing -> gobuster,dirsearch,BurpSmartBuster, recursebuster[https://github.com/C-Sto/recursebuster],

Arjun for Parameter Discover : https://github.com/s0md3v/Arjun Parameth for Parameter Discover : https://github.com/maK-/parameth

Endpoint Discovery -> Linkfinder Target Tab > Right Click Target.com > Save Selected Items python linkfinder.py -o cli -i burpfile

Link Finder Target Tab > Right Click Target.com > Engagement Tools > Find Scripts Ctrl A > Copy Selected URLs (Paste to textfile linkfinder.txt) cat linkfinder.txt | grep .js > linkfinder2.txt python linkfinder.py -o cli -i http://target.com/everylink.js OR copy and paste into JSParser: python handler.py (visit localhost:8008)

Wayback Machine on burp with domains to get endpoints

Resouces Till Now ->

1. https://poc-server.com/blog/2019/01/18/advanced-recon-subdomains/
2. https://twitter.com/aaronhnatiw/status/974682458561097728
3. https://n0where.net/high-performance-dns-stub-resolver-massdns

4. http://offsecbyautomation.com/Use-MassDNS/
5. https://medium.com/@europa_/recoinnassance-7840824b9ef2
6. https://ethical-hacking.securitynewspaper.com/index.php/2019/03/22/chomp-scan-tool-used-by-bug-bounty-hackers/
7. https://github.com/0xhelloworld/public/blob/master/recon%20cheatsheet
8. https://zseano.com/tutorials/6.html
9. https://www.bugbountynotes.com/training/tutorial?id=6
10. https://pentester.land/conference-notes/2018/08/02/levelup-2018-the-bug-hunters-methodology-v3.html
11. https://sylarsec.wordpress.com/2019/01/11/100-ways-to-discover-part-1/

I will continue more in upcoming post and days

✉ choudharyaakash68@gmail.com · · View source · New post · RSS · © 2015 LiXizhi