NetblockTool: The Easy Way to Find IP Addresses Owned by a Company

written by Alex Poorman | October 29, 2020

TL;DR

Use NetblockTool to easily dump a unique list of IP addresses belonging to a company and its subsidiaries.

Download the tool here: https://github.com/NetSPI/NetblockTool

The Problem

A problem that I was frequently running into for both offensive and defensive roles is determining the IP addresses that a company owns and uses. Traditionally, gathering a list of IP addresses a company owns is a long and very manual process. Various sources need to be used like Google, ARIN, WHOIS, IPinfo, Censys, and Shodan. The list goes on.

Thankfully, there are some automated tools that exist that make this process a bit easier. Recon-ng is one of these tools but it isn't perfect, and while it does a lot of things well, easily gathering a complete list of netblocks for a company is not one of those things. This is where NetblockTool comes in.

The Solution: NetblockTool

Written as a standalone Python script, NetblockTool is designed to fill in this tooling gap.

For blue team users, simply provide the name of your company and receive a list of unique netblocks, ranked by the likelihood that the returned netblock belongs to your company.

For red team users, use NetblockTool to gather IP ranges, points of contact, and even netblocks belonging to your target’s subsidiaries.

```
root@kali-ec2:~# ./NetblockTool.py
usage:
 _   _      _   _     _            _    _____           _ 
| \ | | ___| |_| |__ | | ___   ___| | _|_   _|__   ___ | |
|  \| |/ _ \ __| '_ \| |/ _ \ / __| |/ / | |/ _ \ / _ \| |
| |\  |  __/ |_| |_) | | (_) | (__|   <  | | (_) | (_) | |
|_| \_|\___|\__|_.__/|_|\___/ \___|_|\_\ |_|\___/ \___/|_|

./NetblockTool.py [options] {target company}
    Find netblocks owned by a company

Positional arguments:
    {target} Target company (exclude "Inc", "Corp", etc.)

Optional arguments:
    Common Options:
    -l          List mode; argument is a file with list of companies, one per line
    -o          File name to write data to (no extension, default is target name)
    -v          Verbose mode
    -q          Quiet mode
    -h          Print this help message

    Data Retrieval & Processing:
    -n          Don't perform thorough wildcard queries (query = target)
    -ng         Don't perform Google Dorking queries
    -w          Perform more thorough complete wildcard queries (query = *target*). Note
                    that this option may return significantly more false positives.
    -c          Company name if different than target (may affect accuracy of confidence
                    scores, use carefully; exclude "Inc", "Corp", etc.)
    -e          Only return results greater than a given confidence score
    -p          Retrieve and write point of contact information to a text file. Note that
                    retrieval of PoC information will likely take some time.
    -4          Only return IPv4 netblocks
    -6          Only return IPv6 netblocks

    Company Subsidiaries:
    -s          Fetch subsidiary information and return netblocks of all subsidiaries in
                    addition to initial target
    -sn         Company name to use when fetching subsidiaries
    -sp         Use alternate parsing method when fetching subsidiary information; use
                    if the default method isn't working as expected
    -so         Write subsidiary information to a text file (CompanyName_subsidiaries.txt)

    Physical Location:
    -g          Retrieve geolocation data (if available)
    -a          Write netblock address information to output
    -ag         Write netblock address information to output but only if it contains a
                    given string

Examples:
    python NetblockTool.py -v Google
    python NetblockTool.py -so -wv Facebook -o Results
    python NetblockTool.py -gavl companies.txt
```

Getting Started

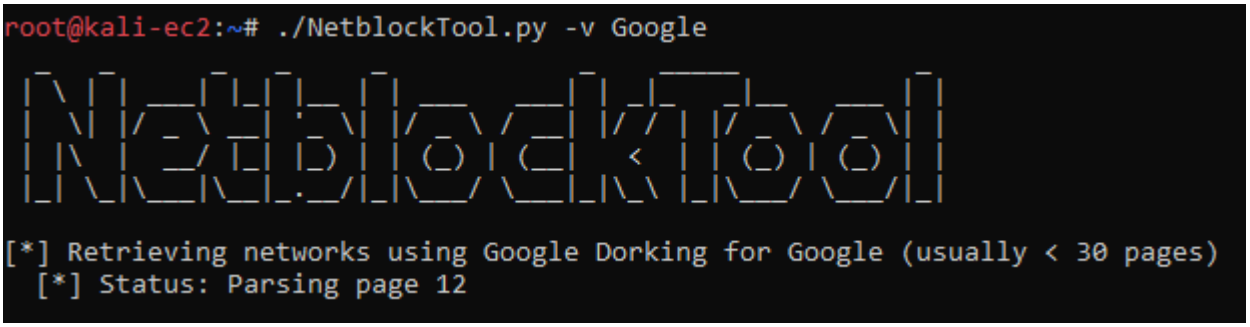
Getting started is easy. Simply clone the repository, install the requirements, and you're ready to start using NetblockTool.

```
git clone https://github.com/NetSPI/NetblockTool.git
cd NetblockTool && pip3 install -r requirements.txt
python3 NetblockTool.py -v Company
```

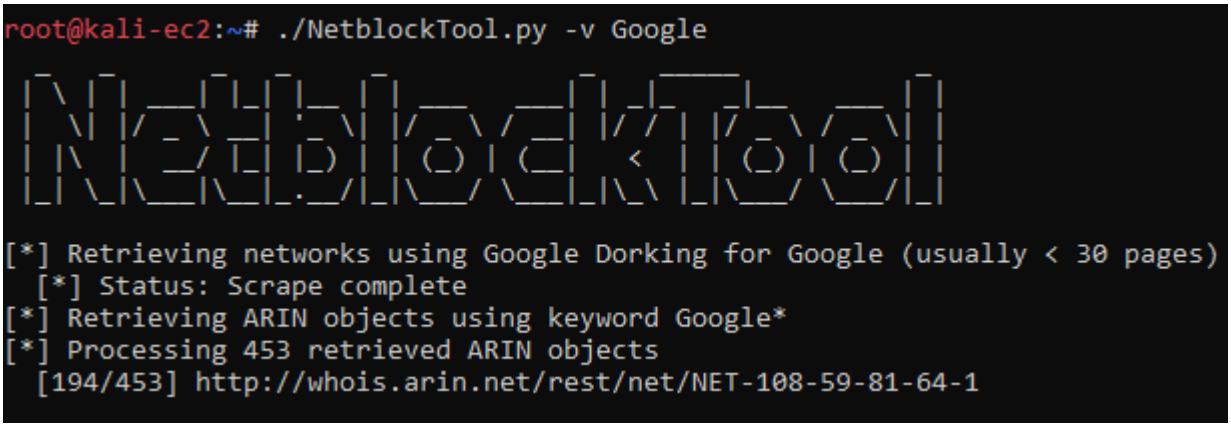
How does it work?

NetblockTool uses several data sources to gather netblocks that a company may own, which include Google dorking, the ARIN database, the ARIN API website, and IPinfo. Since public websites are being scraped, there is no API key needed for any site when using NetblockTool.

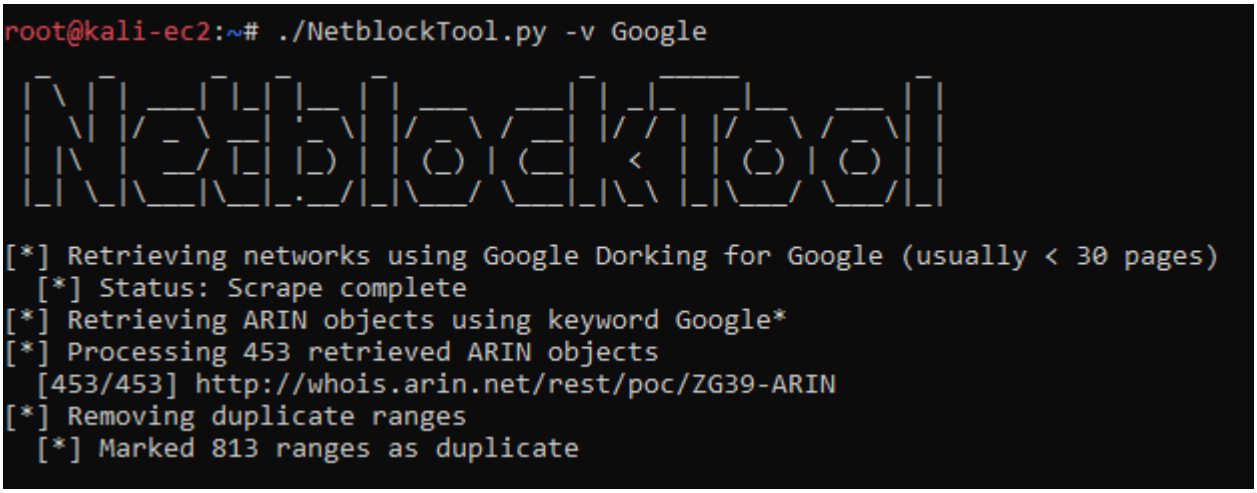
First, the user provides a target company. NetblockTool then scrapes Google using a Google dork to retrieve networks that IPinfo knows about.



Next, the ARIN database is queried by sending the same traffic a normal user would send by visiting their website and manually searching for a company. The results are then scraped for ARIN objects (like networks and company contacts) and the objects are visited and further scraped. The advantage of this method is that more results are provided than just directly querying the database using their APIs.



After all sources have been scraped, each discovered netblock is deduplicated and assigned a confidence score that it belongs to the company. The score is largely based on the name of the netblock, the type of ARIN object it is (either ASN, network, or a leased range known as a customer), and the address linked to the netblock.



From here, further operations are then performed that are based on the user's arguments, such a retrieving geolocation data for each IP.

Finally, the total number of addresses is printed and the results are written to a CSV. The first 15 rows for Google are shown below.

	A	B	C	D	E	F	G
1	Network	Name	ID	Type	Confidence	Score Rationale	Resource URL
2	173.205.116.0/22	Google inc	C06916871	customer	94	Customer, company name is name parameter, address match from ARIN objects	http://whois.arin.net/rest/customer/C06916871
3	199.87.241.32/27	Google	C02765668	customer	94	Customer, company name is name parameter, address match from ARIN objects	http://whois.arin.net/rest/customer/C02765668
4	204.237.189.0/27	Google inc	C06087065	customer	94	Customer, company name is name parameter, address match from ARIN objects	http://whois.arin.net/rest/customer/C06087065
5	204.237.220.0/27	Google inc	C06213948	customer	94	Customer, company name is name parameter, address match from ARIN objects	http://whois.arin.net/rest/customer/C06213948
6	204.237.220.32/27	Google inc	C06217384	customer	94	Customer, company name is name parameter, address match from ARIN objects	http://whois.arin.net/rest/customer/C06217384
7	204.237.220.64/27	Google inc	C06217391	customer	94	Customer, company name is name parameter, address match from ARIN objects	http://whois.arin.net/rest/customer/C06217391
8	204.237.220.96/27	Google inc	C06217392	customer	94	Customer, company name is name parameter, address match from ARIN objects	http://whois.arin.net/rest/customer/C06217392
9	204.237.221.4/30	Google inc	C06238600	customer	94	Customer, company name is name parameter, address match from ARIN objects	http://whois.arin.net/rest/customer/C06238600
10	204.237.223.64/29	Google inc	C06238598	customer	94	Customer, company name is name parameter, address match from ARIN objects	http://whois.arin.net/rest/customer/C06238598
11	64.235.254.0/26	Google inc	C07474745	customer	94	Customer, company name is name parameter, address match from ARIN objects	http://whois.arin.net/rest/customer/C07474745
12	69.174.118.0/24	Google inc	C06885258	customer	94	Customer, company name is name parameter, address match from ARIN objects	http://whois.arin.net/rest/customer/C06885258
13	69.174.119.0/24	Google inc	C06881904	customer	94	Customer, company name is name parameter, address match from ARIN objects	http://whois.arin.net/rest/customer/C06881904
14	128.177.109.0/26	Google Inc.	C05223487	customer	85	Customer, company name is name parameter	http://whois.arin.net/rest/customer/C05223487
15	128.177.109.128/26	Google Inc	C05226420	customer	85	Customer, company name is name parameter	http://whois.arin.net/rest/customer/C05226420

Subsidiaries

What if a company has subsidiaries and has netblocks registered to them? NetblockTool has you covered. It’s able to automatically query the Securities and Exchange Commission’s public database to retrieve a list of possible subsidiaries and then enumerate the subsidiaries’ netblocks.

Common Use Cases

There are many different ways of getting the data you desire from NetblockTool, but the easiest way of running the tool is simply:

python3 NetblockTool.py -v Company

If you want to extract netblocks owned by your target company’s subsidiaries, specify that flag:

```
python3 NetblockTool.py -v Company -s
```

Extracting point of contact information can also be helpful:

```
python3 NetblockTool.py -v Company -p
```

Or, if you want to get as much information as possible, including netblocks found using wildcard queries, points of contact, geolocation data, and physical addresses:

```
python3 NetblockTool.py -wpgav Company -so
```

Conclusion

Whether you need to find the netblocks your employer owns or find the netblocks for your next red team engagement, NetblockTool is your quick and easy solution. Give it a shot and see if you find it useful.

https://github.com/NetSPI/NetblockTool