main Branches Tags Go to file Code ···

- .github
- cmd/asnmap
- examples
- libs
- runner
- .dockerignore
- .gitignore
- .goreleaser.yml
- Dockerfile
- LICENSE
- Makefile
- README.md
- go.mod
- go.sum

About

Go CLI and Library for quickly mapping organization network ranges using ASN information.

#osint #asn #recon #asn-lookup #cidr-range

- Readme
- MIT license
- Code of conduct
- Security policy
- Activity
- Custom properties
- **689** stars
- **17** watching
- **65** forks

Report repository

Releases 10

v1.1.1 Latest
2 days ago

+ 9 releases

Packages

Used by 209

+ 201

Contributors 11

Languages

● **Go** 97.9% ● **Makefile** 1.2% ● **Dockerfile** 0.9%

README Code of conduct More

ASNmap

Go CLI and Library for quickly mapping organization network ranges using ASN information.

go report A | contributions welcome | release v1.1.1 | Follow @pdiscoveryio | chat 833 online

Features • Installation • Usage • Running asnmap • Join Discord

Features

- **ASN to CIDR** Lookup
- **ORG to CIDR** Lookup
- **DNS to CIDR** Lookup
- **IP to CIDR** Lookup
- **ASN/DNS/IP/ORG** input
- **JSON/CSV/TEXT** output
- STD **IN/OUT** support

Installation

asnmap requires **Go 1.21** to install successfully. To install, just run the below command or download pre-compiled binary from release page.

```
go install github.com/projectdiscovery/asnmap/cmd/asnmap@latest
```

Usage

```
asnmap -h
```

This will display help for the tool. Here are all the flag it supports.

```
Usage:
  ./asnmap [flags]

Flags:
INPUT:
   -a, -asn string[]     target asn to lookup, example: -a AS5650
   -i, -ip string[]      target ip to lookup, example: -i 100.19.12.21, -i 2a10:ad40::
   -d, -domain string[]  target domain to lookup, example: -d google.com, -d facebook.com
   -org string[]         target organization to lookup, example: -org GOOGLE
   -f, -file string[]    targets to lookup from file

CONFIGURATIONS:
   -config string           path to the asnmap configuration file
   -r, -resolvers string[]  list of resolvers to use

UPDATE:
   -up, -update                 update asnmap to latest version
   -duc, -disable-update-check  disable automatic asnmap update check
```

```
OUTPUT:
   -o, -output string  file to write output to
   -j, -json           display json format output
   -c, -csv            display csv format output
   -v6                 display ipv6 cidr ranges in cli output
   -v, -verbose        display verbose output
   -silent             display silent output
   -version            show version of the project
```

Configuring ASNMap CLI

ASNMap CLI is built on top of the ASNMap API that requires API Token from ProjectDiscovery Cloud Platform that can be configured using environment variable or using interactive `-auth` option as shown below.

Using environment variable

```
export PDCP_API_KEY=*************
```

Using auth option

```
asnmap -auth


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

    projectdiscovery.io

[INF] Get your free api key by signing up at https://cloud.projectdiscovery.io
[*] Enter PDCP API Key (exit to abort): *************
[INF] Successfully logged in as (@user)
```

Running asnmap

asnmap support multiple inputs including **ASN**, **IP**, **DNS** and **ORG** name to query ASN/CIDR information.

Input	ASN	DNS	IP	ORG
Example	`AS14421`	`example.com`	`93.184.216.34`	`GOOGLE`

Input can be provided either using specific options or STDIN which accepts all the supported formats. Single, multiple (comma-separated) and file input is supported for all the options.

```
echo GOOGLE | ./asnmap -silent
```

Example input for asnmap:

```
asnmap -a AS45596 -silent
asnmap -i 100.19.12.21 -silent
asnmap -d hackerone.com -silent
asnmap -org GOOGLE -silent
```

Default Run

asnmap by default returns the CIDR range for given input.

```
echo GOOGLE | ./asnmap


   ___   _____  __              
  / _ | / __/ |/ /_ _  ___ ____
 / __ |_\ \/    /  ' \/ _  / _ \
/_/ |_/___/_/|_/_/_/_/\_,_/ .__/
                         /_/    v0.0.1
```

```
                  projectdiscovery.io

Use with caution. You are responsible for your actions
Developers assume no liability and are not responsible for any misuse or damage.

8.8.4.0/24
8.8.8.0/24
8.35.200.0/21
34.3.3.0/24
34.4.4.0/24
34.96.0.0/20
34.96.32.0/19
34.96.64.0/18
34.98.64.0/18
34.98.136.0/21
34.98.144.0/21
```

JSON Output

asnmap by default displays CIDR range, and all the information is always available in JSON format, for automation and post processing using `-json` output is most convenient option to use.

```
echo hackerone.com | ./asnmap -json -silent | jq
```

```
{
  "timestamp": "2022-09-19 12:14:33.267339314 +0530 IST",
  "input": "hackerone.com",
  "as_number": "AS13335",
  "as_name": "CLOUDFLARENET",
  "as_country": "US",
  "as_range": [
    "104.16.0.0/14",
    "104.20.0.0/16",
    "104.21.0.0/17"
  ]
}
{
  "timestamp": "2022-09-19 12:14:33.457401266 +0530 IST",
  "input": "hackerone.com",
  "as_number": "AS13335",
  "as_name": "CLOUDFLARENET",
  "as_country": "US",
  "as_range": [
    "2606:4700:8390::/44"
  ]
}
```

CSV Output

asnmap also support csv format output which has all the information just like JSON output

```
echo hackerone.com | ./asnmap -csv -silent
```

```
timestamp|input|as_number|as_name|as_country|as_range
2022-09-19 12:15:04.906664007 +0530 IST|hackerone.com|AS13335|CLOUDFLARENET|US|104.16.0.0/14,104.20.0.0/16,104.21.0.0
2022-09-19 12:15:05.201328136 +0530 IST|hackerone.com|AS13335|CLOUDFLARENET|US|2606:4700:9760::/44
```

Using with other PD projects

Output of asnmap can be directly piped into other projects in workflow accepting stdin as input, for example:

- `echo AS54115 | asnmap | tlsx`
- `echo AS54115 | asnmap | dnsx -ptr`
- `echo AS54115 | asnmap | naabu -p 443`
- `echo AS54115 | asnmap | naabu -p 443 | httpx`
- `echo AS54115 | asnmap | naabu -p 443 | httpx | nuclei -id tech-detect`

Use asnmap as a library