Sign in

projectdiscovery / naabu Public

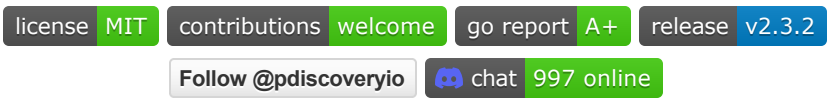
Notifications | Fork 548 | Star 4.7k

Code | Issues 12 | Pull requests 2 | Discussions | Actions | Projects | Security

dev | Go to file | Code

- .github
- integration_tests
- static
- v2
- .gitignore
- Dockerfile
- LICENSE.md
- README.md
- THANKS.md

README | Code of conduct | MIT license | Security



About

A fast port scanner written in go with a focus on reliability and simplicity. Designed to be used in combination with other tools for attack surface discovery in bug bounties and pentests

projectdiscovery.io

nmap | scan-ports | hacktoberfest | portscanner | port-enumeration | cdn-exclusion

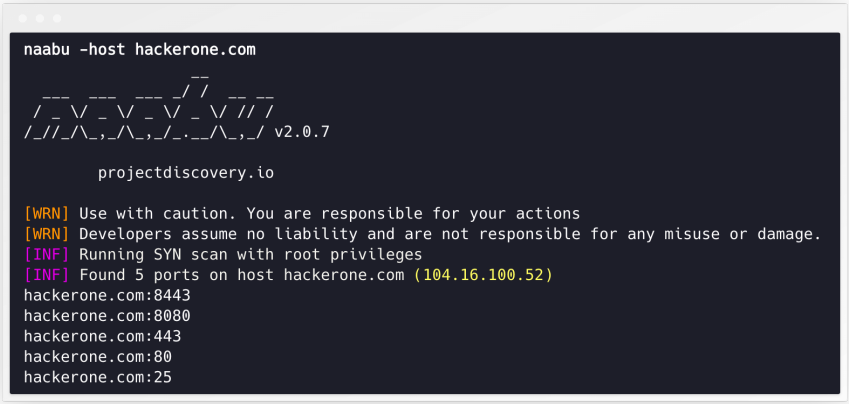
Readme
MIT license
Code of conduct
Security policy
Activity
Custom properties
4.7k stars
69 watching
548 forks
Report repository

Releases 32

Features • Installation • Usage • Running naabu • Config • NMAP integration • CDN/WAF Exclusion • Discord

Naabu is a port scanning tool written in Go that allows you to enumerate valid ports for hosts in a fast and reliable manner. It is a really simple tool that does fast SYN/CONNECT/UDP scans on the host/list of hosts and lists all ports that return a reply.

Features



- Fast And Simple **SYN/CONNECT/UDP** probe based scanning
- Optimized for ease of use and **lightweight** on resources
- **DNS** Port scan
- **Automatic IP Deduplication** for DNS port scan
- **IPv4/IPv6** Port scan (**experimental**)
- **Passive** Port enumeration using Shodan Internetdb
- **Host Discovery** scan (**experimental**)
- **NMAP** integration for service discovery
- Multiple input support - **STDIN/HOST/IP/CIDR/ASN**
- Multiple output format support - **JSON/TXT/STDOUT**

Usage

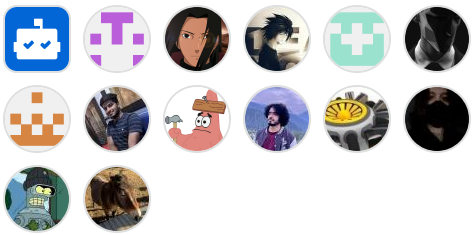
v2.3.2 Latest
yesterday

+ 31 releases

Packages

No packages published

Contributors 49



+ 35 contributors

Languages

Go 99.2% Other 0.8%

```
naabu -h
```

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

```
Usage:
  ./naabu [flags]

INPUT:
   -host string[]              hosts to scan por
   -list, -l string            list of hosts to
   -exclude-hosts, -eh string  hosts to exclude
   -exclude-file, -ef string   list of hosts to

PORT:
   -port, -p string            ports to scan (8(
   -top-ports, -tp string      top ports to scar
   -exclude-ports, -ep string  ports to exclude
   -ports-file, -pf string     list of ports to
   -port-threshold, -pts int   port threshold t(
   -exclude-cdn, -ec           skip full port s(
   -display-cdn, -cdn          display cdn in us

RATE-LIMIT:
   -c int     general internal worker threads (d
   -rate int  packets to send per second (defau

UPDATE:
   -up, -update                 update naabu to
   -duc, -disable-update-check  disable automat

OUTPUT:
   -o, -output string  file to write output to
   -j, -json           write output in JSON lin(
   -csv                write output in csv form

CONFIGURATION:
   -config string                  path to the
   -scan-all-ips, -sa              scan all th(
   -ip-version, -iv string[]       ip version
   -scan-type, -s string           type of por
   -source-ip string               source ip a
   -interface-list, -il            list availal
```

```
   -interface, -i string            network Int
   -nmap                            invoke nmap
   -nmap-cli string                 nmap comman
   -r string                        list of cus
   -proxy string                    socks5 prox
   -proxy-auth string               socks5 prox
   -resume                          resume scan
   -stream                          stream mode
   -passive                         display pas
   -irt, -input-read-timeout value  timeout on
   -no-stdin                        Disable Std

HOST-DISCOVERY:
   -sn, -host-discovery           Perform Only
   -Pn, -skip-host-discovery      Skip Host dis
   -ps, -probe-tcp-syn string[]   TCP SYN Ping
   -pa, -probe-tcp-ack string[]   TCP ACK Ping
   -pe, -probe-icmp-echo          ICMP echo req
   -pp, -probe-icmp-timestamp     ICMP timestam
   -pm, -probe-icmp-address-mask  ICMP address
   -arp, -arp-ping                ARP ping (hos
   -nd, -nd-ping                  IPv6 Neighbor
   -rev-ptr                       Reverse PTR l

OPTIMIZATION:
   -retries int       number of retries for the
   -timeout int       millisecond to wait befor
   -warm-up-time int  time in seconds between s
   -ping              ping probes for verificat
   -verify            validate the ports again

DEBUG:
   -health-check, -hc        run diagnostic che
   -debug                    display debugging
   -verbose, -v              display verbose ou
   -no-color, -nc            disable colors in
   -silent                   display only resul
   -version                  display version of
   -stats                    display stats of t
   -si, -stats-interval int  number of seconds
   -mp, -metrics-port int    port to expose naa
```

Installation Instructions

Download the ready to run binary / docker or install with GO

Prerequisite

> **Note**: before installing naabu, make sure to install `libpcap` library for packet capturing.

To install libcap on **Linux**: `sudo apt install -y libpcap-dev`, on **Mac**: `brew install libpcap`

Installing Naabu

```
go install -v github.com/projectdiscovery/naabu/
```

Running Naabu

To run the tool on a target, just use the following command.

```
naabu -host hackerone.com
```

This will run the tool against hackerone.com. There are a number of configuration options that you can pass along with this command. The verbose switch `-v` can be used to display verbose information.

```
naabu -host hackerone.com

                 __
  ___  ___  ___ _/ /  __ __
 / _ \/ _ \/ _ \ _ \ // /
/_//_/\_,_/\_,_/_.__/\_,_/ v2.0.3

    projectdiscovery.io

[WRN] Use with caution. You are responsible for
[WRN] Developers assume no liability and are no
[INF] Running SYN scan with root privileges
```

```
[INF] Found 4 ports on host hackerone.com (104.

hackerone.com:80
hackerone.com:443
hackerone.com:8443
hackerone.com:8080
```

The ports to scan for on the host can be specified via `-p` parameter (udp ports must be expressed as `u:port`). It takes nmap format ports and runs enumeration on them.

```
naabu -p 80,443,21-23,u:53 -host hackerone.com
```

By default, the Naabu checks for nmap's `Top 100` ports. It supports the following in-built port lists -

Flag	Description
`-top-ports 100`	Scan for nmap top **100** port
`-top-ports 1000`	Scan for nmap top **1000** port
`-p -`	Scan for full ports from **1-65535**

You can also specify specific ports which you would like to exclude from the scan.

```
naabu -p - -exclude-ports 80,443
```

To run the naabu on a list of hosts, `-list` option can be used.

```
naabu -list hosts.txt
```

To run the naabu on a ASN, AS input can be used. It takes the IP address available for given ASN and runs the enumeration on them.

```
echo AS14421 | naabu -p 80,443

216.101.17.249:80
216.101.17.249:443
216.101.17.248:443
216.101.17.252:443
216.101.17.251:80
216.101.17.251:443
216.101.17.250:443
216.101.17.250:80
```

You can also get output in json format using `-json` switch. This switch saves the output in the JSON lines format.

```
naabu -host 104.16.99.52 -json

{"ip":"104.16.99.52","port":443}
{"ip":"104.16.99.52","port":80}
```

The ports discovered can be piped to other tools too. For example, you can pipe the ports discovered by naabu to httpx which will then find running http servers on the host.

```
echo hackerone.com | naabu -silent | httpx -sil

http://hackerone.com:8443
http://hackerone.com:443
http://hackerone.com:8080
http://hackerone.com:80
```

The speed can be controlled by changing the value of `rate` flag that represent the number of packets per second. Increasing it while processing hosts may lead to increased false-positive rates. So it is recommended to keep it to a reasonable amount.

IPv4 and IPv6

Naabu supports both IPv4 and IPv6. Both ranges can be piped together as input. If IPv6 is used, connectivity must be correctly configured, and the network interface must have an IPv6 address assigned (`inet6`) and a default gateway.

```
echo hackerone.com | dnsx -resp-only -a -aaaa -s

104.16.99.52:80
104.16.100.52:80
2606:4700::6810:6434:80
2606:4700::6810:6334:80
```

The option `-ip-version 6` makes the tool use IPv6 addresses while resolving domain names.

```
echo hackerone.com | ./naabu -p 80 -ip-version 6

                __
  ___  ___  ___ _/ /  __ __
 / _ \/ _ \/ _ \ _ \// /
/_//_/\_,_/\_,_/_.__/\_,_/ v2.0.8

      projectdiscovery.io

Use with caution. You are responsible for your a
Developers assume no liability and are not respo
[INF] Running CONNECT scan with non root privile
[INF] Found 1 ports on host hackerone.com (2606
hackerone.com:80
```

To scan all the IPs of both version, `ip-version 4,6` can be used along with `-scan-all-ips` flag.

```
echo hackerone.com | ./naabu -iv 4,6 -sa -p 80

[INF] Found 1 ports on host hackerone.com (104.1
hackerone.com:80
[INF] Found 1 ports on host hackerone.com (104.1
hackerone.com:80
[INF] Found 1 ports on host hackerone.com (2606
hackerone.com:80
```

```
[INF] Found 1 ports on host hackerone.com (2606
hackerone.com:80
```

Host Discovery

Naabu optionally supports multiple options to perform host discovery, as outlined below. Host discovery is completed automatically before beginning a connect/syn scan if the process has enough privileges. `-sn` flag instructs the toll to perform host discovery only. `-Pn` flag skips the host discovery phase. Host discovery is completed using multiple internal methods; one can specify the desired approach to perform host discovery by setting available options.

Available options to perform host discovery:

- **ARP** ping (`-arp`)
- TCP **SYN** ping (`-ps 80`)
- TCP **ACK** ping (`-pa 443`)
- ICMP **echo** ping (`-pe`)
- ICMP **timestamp** ping (`-pp`)
- ICMP **address mask** ping (`-pm`)
- IPv6 **neighbor discovery** (`-nd`)

Configuration file

Naabu supports config file as default located at `$HOME/.config/naabu/config.yaml`, It allows you to define any flag in the config file and set default values to include for all scans.

Nmap integration

We have integrated nmap support for service discovery or any additional scans supported by nmap on the found results by

Naabu, make sure you have `nmap` installed to use this feature.

To use, `nmap-cli` flag can be used followed by nmap command, for example:-

```
echo hackerone.com | naabu -nmap-cli 'nmap -sV
                    __
   ___  ___  ___ _/ /  __ __
  / _ \/ _ \/ _ \ _ \ // /
 /_//_/\_,_/\_,_/_.__/\_,_/ v2.0.0

    projectdiscovery.io

[WRN] Use with caution. You are responsible for
[WRN] Developers assume no liability and are no
[INF] Running TCP/ICMP/SYN scan with root privi
[INF] Found 4 ports on host hackerone.com (104.

hackerone.com:443
hackerone.com:80
hackerone.com:8443
hackerone.com:8080

[INF] Running nmap command: nmap -sV -p 80,8443

Starting Nmap 7.01 ( https://nmap.org ) at 2020
Nmap scan report for 104.16.99.52
Host is up (0.0021s latency).
PORT     STATE SERVICE       VERSION
80/tcp   open  http          cloudflare
443/tcp  open  ssl/https     cloudflare
8080/tcp open  http-proxy    cloudflare
8443/tcp open  ssl/https-alt cloudflare
```

CDN/WAF Exclusion

Naabu also supports excluding CDN/WAF IPs being port scanned. If used, only `80` and `443` ports get scanned for those IPs. This feature can be enabled by using `exclude-cdn` flag.

Currently `cloudflare`, `akamai`, `incapsula` and `sucuri` IPs are supported for exclusions.

Scan Status

Naabu exposes json scan info on a local port bound to localhost at `http://localhost:63636/metrics` (the port can be changed via the `-metrics-port` flag)

Using naabu as library

The following sample program scan the port `80` of `scanme.sh`. The results are returned via the `OnResult` callback:

```
package main

import (
        "log"

        "github.com/projectdiscovery/goflags"
        "github.com/projectdiscovery/naabu/v2/pl
        "github.com/projectdiscovery/naabu/v2/pl
)

func main() {
        options := runner.Options{
                Host:      goflags.StringSlice{'
                ScanType: "s",
                OnResult: func(hr *result.HostR
                        log.Println(hr.Host, hr
                },
                Ports: "80",
        }

        naabuRunner, err := runner.NewRunner(&o
        if err != nil {
                log.Fatal(err)
        }
        defer naabuRunner.Close()
```

```
        naabuRunner.RunEnumeration()
}
```

Notes

- Naabu allows arbitrary binary execution as a feature to support nmap integration.
- Naabu is designed to scan ports on multiple hosts / mass port scanning.
- As default naabu is configured with a assumption that you are running it from VPS.
- We suggest tuning the flags / rate if running naabu from local system.
- For best results, run naabu as **root** user.