dev ▾ | Branches | Tags | Go to file | Code ▾ | ···

- .github
- .run
- cmd
- examples
- helm
- integration_tests
- internal
- lib
- pkg
- static
- .gitignore
- .goreleaser.yml
- CONTRIBUTING.md
- DEBUG.md
- DESIGN.md
- Dockerfile
- LICENSE.md
- Makefile
- README.md
- README_CN.md
- README_ES.md
- README_ID.md
- README_JP.md
- README_KR.md
- SYNTAX-REFERENCE.md
- THANKS.md
- gh_retry.sh
- go.mod
- go.sum
- nuclei-jsonschema.json

About

Fast and customizable vulnerability scanner based on simple YAML based DSL.

docs.projectdiscovery.io/tools/nucl...

#security #vulnerability-detection #hacktoberfest #security-scanner #vulnerability-assessment #vulnerability-scanner #attack-surface #subdomain-takeover #cve-scanner #nuclei-engine

- Readme
- MIT license
- Code of conduct
- Security policy
- Activity
- Custom properties
- **18.1k** stars
- **218** watching
- **2.3k** forks

Report repository

Releases 113

v3.2.9 Latest
2 weeks ago

+ 112 releases

Packages

No packages published

Contributors 150

+ 136 contributors

Languages

● **Go** 96.4%	● **TypeScript** 2.9%
● **JavaScript** 0.3%	● **Shell** 0.2%
● **Smarty** 0.1%	● **Makefile** 0.1%

README | Code of conduct | More ▾

nuclei

Fast and customisable vulnerability scanner based on simple YAML based DSL.

Go v1.21 downloads 1.7M contributors 163 release v3.2.9 open issues 266 discussions 578 total chat 835 online Follow @pdnuclei

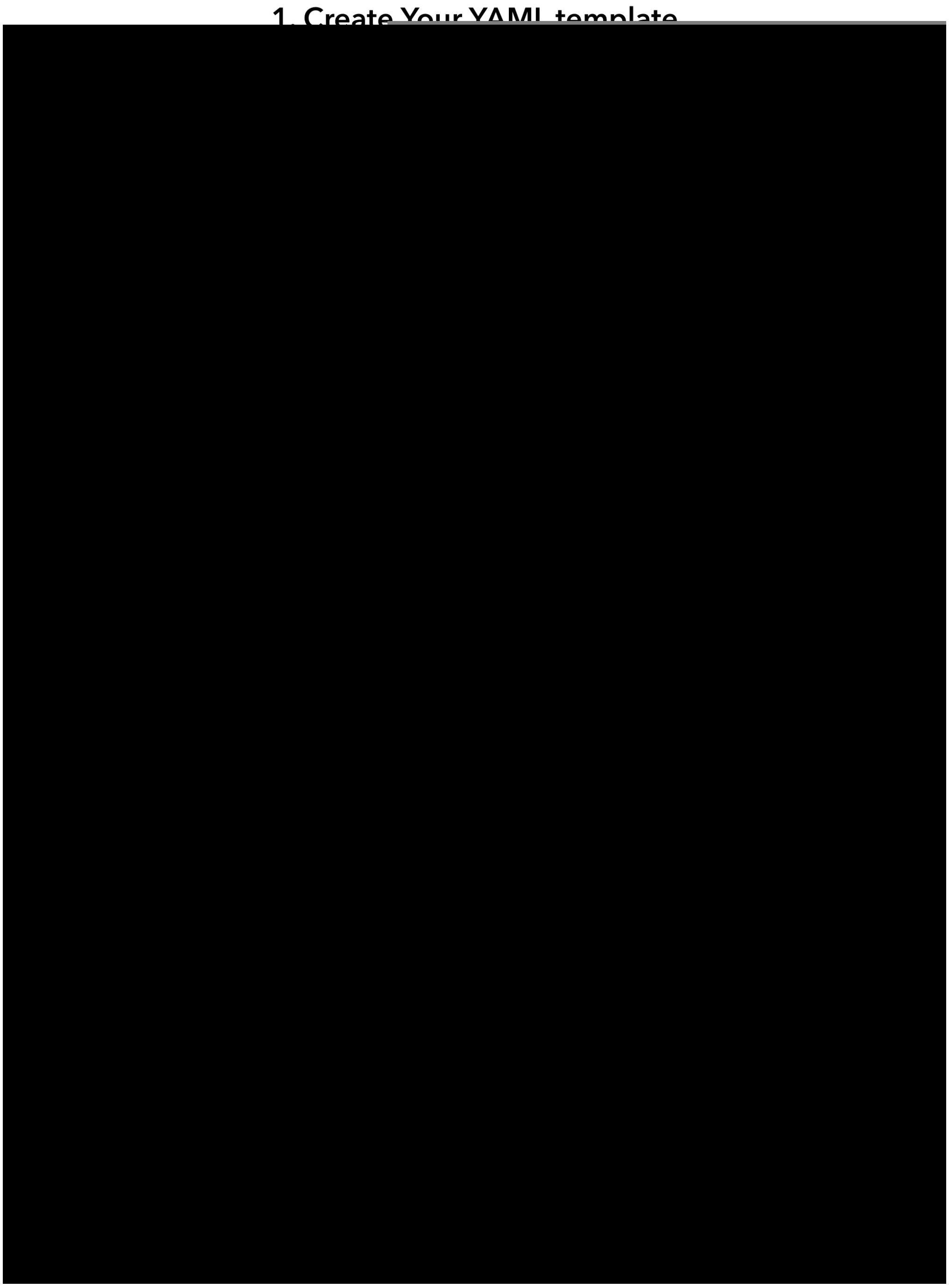
How • Install • Documentation • Credits • FAQs • Join Discord

English • 中文 • Korean • Indonesia • Spanish • 日本語

Nuclei is used to send requests across targets based on a template, leading to zero false positives and providing fast scanning on a large number of hosts. Nuclei offers scanning for a variety of protocols, including TCP, DNS, HTTP, SSL, File, Whois, Websocket, Headless, Code etc. With powerful and flexible templating, Nuclei can be used to model all kinds of security checks.

We have a dedicated repository that houses various type of vulnerability templates contributed by **more than 300** security researchers and engineers.

How it works



❗ **Disclaimer**
This project is in active development. Expect breaking changes with releases. Review the release changelog before updating.
This project was primarily built to be used as a standalone CLI tool. **Running nuclei as a service may pose security risks.** It's recommended to use with caution and additional security measures.

Install Nuclei

Nuclei requires **go1.21** to install successfully. Run the following command to install the latest version -

```
go install -v github.com/projectdiscovery/nuclei/v3/cmd/nuclei@latest
```

▶ Brew

▶ Docker

More installation methods can be found here.

Nuclei Templates

Nuclei has built-in support for automatic template download/update as default since version v2.5.2. **Nuclei-Templates** project provides a community-contributed list of ready-to-use templates that is constantly updated.

You may still use the `update-templates` flag to update the nuclei templates at any time; You can write your own checks for your individual workflow and needs following Nuclei's templating guide.

The YAML DSL reference syntax is available here.

Usage

```
nuclei -h
```

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

```
Nuclei is a fast, template based vulnerability scanner focusing
on extensive configurability, massive extensibility and ease of use.

Usage:
  ./nuclei [flags]

Flags:
TARGET:
   -u, -target string[]          target URLs/hosts to scan
   -l, -list string              path to file containing a list of target URLs/hosts to scan (one per line)
   -eh, -exclude-hosts string[]  hosts to exclude to scan from the input list (ip, cidr, hostname)
   -resume string                resume scan using resume.cfg (clustering will be disabled)
   -sa, -scan-all-ips            scan all the IP's associated with dns record
   -iv, -ip-version string[]     IP version to scan of hostname (4,6) - (default 4)

TARGET-FORMAT:
   -im, -input-mode string        mode of input file (list, burp, jsonl, yaml, openapi, swagger) (default "list")
   -ro, -required-only            use only required fields in input format when generating requests
   -sfv, -skip-format-validation  skip format validation (like missing vars) when parsing input file

TEMPLATES:
   -nt, -new-templates                    run only new templates added in latest nuclei-templates release
   -ntv, -new-templates-version string[]  run new templates added in specific version
   -as, -automatic-scan                   automatic web scan using wappalyzer technology detection to tags mapping
   -t, -templates string[]                list of template or template directory to run (comma-separated, file)
   -turl, -template-url string[]          template url or list containing template urls to run (comma-separated, file
   -w, -workflows string[]                list of workflow or workflow directory to run (comma-separated, file)
   -wurl, -workflow-url string[]          workflow url or list containing workflow urls to run (comma-separated, file
   -validate                              validate the passed templates to nuclei
   -nss, -no-strict-syntax                disable strict syntax check on templates
   -td, -template-display                 displays the templates content
   -tl                                    list all available templates
   -tgl                                   list all available tags
   -sign                                  signs the templates with the private key defined in NUCLEI_SIGNATURE_PRIVAT
   -code                                  enable loading code protocol-based templates
   -dut, -disable-unsigned-templates      disable running unsigned templates or templates with mismatched signature

FILTERING:
   -a, -author string[]               templates to run based on authors (comma-separated, file)
   -tags string[]                     templates to run based on tags (comma-separated, file)
   -etags, -exclude-tags string[]     templates to exclude based on tags (comma-separated, file)
   -itags, -include-tags string[]     tags to be executed even if they are excluded either by default or configuratic
   -id, -template-id string[]         templates to run based on template ids (comma-separated, file, allow-wildcard)
   -eid, -exclude-id string[]         templates to exclude based on template ids (comma-separated, file)
   -it, -include-templates string[]   path to template file or directory to be executed even if they are excluded eit
   -et, -exclude-templates string[]   path to template file or directory to exclude (comma-separated, file)
   -em, -exclude-matchers string[]    template matchers to exclude in result
   -s, -severity value[]              templates to run based on severity. Possible values: info, low, medium, high, c
   -es, -exclude-severity value[]     templates to exclude based on severity. Possible values: info, low, medium, hig
   -pt, -type value[]                 templates to run based on protocol type. Possible values: dns, file, http, heac
   -ept, -exclude-type value[]        templates to exclude based on protocol type. Possible values: dns, file, http,
   -tc, -template-condition string[]  templates to run based on expression condition

OUTPUT:
   -o, -output string            output file to write found issues/vulnerabilities
   -sresp, -store-resp           store all request/response passed through nuclei to output directory
   -srd, -store-resp-dir string  store all request/response passed through nuclei to custom directory (default "outpu
   -silent                       display findings only
   -nc, -no-color                disable output content coloring (ANSI escape codes)
   -j, -jsonl                    write output in JSONL(ines) format
   -irr, -include-rr -omit-raw   include request/response pairs in the JSON, JSONL, and Markdown outputs (for finding
   -or, -omit-raw                omit request/response pairs in the JSON, JSONL, and Markdown outputs (for findings c
   -ot, -omit-template           omit encoded template in the JSON, JSONL output
   -nm, -no-meta                 disable printing result metadata in cli output
   -ts, -timestamp               enables printing timestamp in cli output
   -rdb, -report-db string       nuclei reporting database (always use this to persist report data)
   -ms, -matcher-status          display match failure status
   -me, -markdown-export string  directory to export results in markdown format
   -se, -sarif-export string     file to export results in SARIF format
   -je, -json-export string      file to export results in JSON format
   -jle, -jsonl-export string    file to export results in JSONL(ine) format

CONFIGURATIONS:
   -config string                        path to the nuclei configuration file
   -tp, -profile string                  template profile config file to run
   -tpl, -profile-list                   list community template profiles
   -fr, -follow-redirects                enable following redirects for http templates
   -fhr, -follow-host-redirects          follow redirects on the same host
   -mr, -max-redirects int               max number of redirects to follow for http templates (default 10)
   -dr, -disable-redirects               disable redirects for http templates
   -rc, -report-config string            nuclei reporting module configuration file
   -H, -header string[]                  custom header/cookie to include in all http request in header:value format (
   -V, -var value                        custom vars in key=value format
   -r, -resolvers string                 file containing resolver list for nuclei
   -sr, -system-resolvers                use system DNS resolving as error fallback
   -dc, -disable-clustering              disable clustering of requests
   -passive                              enable passive HTTP response processing mode
   -fh2, -force-http2                    force http2 connection on requests
   -ev, -env-vars                        enable environment variables to be used in template
   -cc, -client-cert string              client certificate file (PEM-encoded) used for authenticating against scanne
   -ck, -client-key string               client key file (PEM-encoded) used for authenticating against scanned hosts
   -ca, -client-ca string                client certificate authority file (PEM-encoded) used for authenticating agai
   -sml, -show-match-line                show match lines for file templates, works with extractors only
   -ztls                                 use ztls library with autofallback to standard one for tls13 [Deprecated] au
   -sni string                           tls sni hostname to use (default: input domain name)
   -dt, -dialer-timeout value            timeout for network requests.
   -dka, -dialer-keep-alive value        keep-alive duration for network requests.
   -lfa, -allow-local-file-access        allows file (payload) access anywhere on the system
   -lna, -restrict-local-network-access  blocks connections to the local / private network
   -i, -interface string                 network interface to use for network scan
   -at, -attack-type string              type of payload combinations to perform (batteringram,pitchfork,clusterbomb)
   -sip, -source-ip string               source ip address to use for network scan
   -rsr, -response-size-read int         max response size to read in bytes
   -rss, -response-size-save int         max response size to read in bytes (default 1048576)
   -rrt, -response-read-timeout value    response read timeout in seconds (default 5s)
   -reset                                reset removes all nuclei configuration and data files (including nuclei-temp
   -tlsi, -tls-impersonate               enable experimental client hello (ja3) tls randomization
   -hae, -http-api-endpoint string       experimental http api endpoint

INTERACTSH:
   -iserver, -interactsh-server string  interactsh server url for self-hosted instance (default: oast.pro,oast.live,c
   -itoken, -interactsh-token string    authentication token for self-hosted interactsh server
   -interactions-cache-size int         number of requests to keep in the interactions cache (default 5000)
   -interactions-eviction int           number of seconds to wait before evicting requests from cache (default 60)
   -interactions-poll-duration int      number of seconds to wait before each interaction poll request (default 5)
   -interactions-cooldown-period int    extra time for interaction polling before exiting (default 5)
   -ni, -no-interactsh                  disable interactsh server for OAST testing, exclude OAST based templates

FUZZING:
   -ft, -fuzzing-type string     overrides fuzzing type set in template (replace, prefix, postfix, infix)
   -fm, -fuzzing-mode string     overrides fuzzing mode set in template (multiple, single)
   -fuzz                         enable loading fuzzing templates (Deprecated: use -dast instead)
   -dast                         enable / run dast (fuzz) nuclei templates
   -dfp, -display-fuzz-points    display fuzz points in the output for debugging
   -fuzz-param-frequency int     frequency of uninteresting parameters for fuzzing before skipping (default 10)
   -fa, -fuzz-aggression string  fuzzing aggression level controls payload count for fuzz (low, medium, high) (defaul

UNCOVER:
   -uc, -uncover                  enable uncover engine
   -uq, -uncover-query string[]   uncover search query
   -ue, -uncover-engine string[]  uncover search engine (shodan,censys,fofa,shodan-idb,quake,hunter,zoomeye,netlas,cr
   -uf, -uncover-field string     uncover fields to return (ip,port,host) (default "ip:port")
   -ul, -uncover-limit int        uncover results to return (default 100)
   -ur, -uncover-ratelimit int    override ratelimit of engines with unknown ratelimit (default 60 req/min) (default

RATE-LIMIT:
   -rl, -rate-limit int               maximum number of requests to send per second (default 150)
   -rld, -rate-limit-duration value   maximum number of requests to send per second (default 1s)
   -rlm, -rate-limit-minute int       maximum number of requests to send per minute (DEPRECATED)
   -bs, -bulk-size int                maximum number of hosts to be analyzed in parallel per template (default 25)
   -c, -concurrency int               maximum number of templates to be executed in parallel (default 25)
   -hbs, -headless-bulk-size int      maximum number of headless hosts to be analyzed in parallel per template (defau
   -headc, -headless-concurrency int  maximum number of headless templates to be executed in parallel (default 10)
   -jsc, -js-concurrency int          maximum number of javascript runtimes to be executed in parallel (default 120)
   -pc, -payload-concurrency int      max payload concurrency for each template (default 25)
   -prc, -probe-concurrency int       http probe concurrency with httpx (default 50)

OPTIMIZATIONS:
   -timeout int                     time to wait in seconds before timeout (default 10)
   -retries int                     number of times to retry a failed request (default 1)
   -ldp, -leave-default-ports       leave default HTTP/HTTPS ports (eg. host:80,host:443)
   -mhe, -max-host-error int        max errors for a host before skipping from scan (default 30)
   -te, -track-error string[]       adds given error to max-host-error watchlist (standard, file)
   -nmhe, -no-mhe                   disable skipping host from scan based on errors
   -project                         use a project folder to avoid sending same request multiple times
   -project-path string             set a specific project path (default "/tmp")
   -spm, -stop-at-first-match       stop processing HTTP requests after the first match (may break template/workflow
   -stream                          stream mode - start elaborating without sorting the input
   -ss, -scan-strategy value        strategy to use while scanning(auto/host-spray/template-spray) (default auto)
   -irt, -input-read-timeout value  timeout on input read (default 3m0s)
   -nh, -no-httpx                   disable httpx probing for non-url input
   -no-stdin                        disable stdin processing

HEADLESS:
   -headless                        enable templates that require headless browser support (root user on Linux will c
   -page-timeout int                seconds to wait for each page in headless mode (default 20)
   -sb, -show-browser               show the browser on the screen when running templates with headless mode
   -ho, -headless-options string[]  start headless chrome with additional options
   -sc, -system-chrome              use local installed Chrome browser instead of nuclei installed
   -lha, -list-headless-action      list available headless actions

DEBUG:
   -debug                    show all requests and responses
   -dreq, -debug-req         show all sent requests
   -dresp, -debug-resp       show all received responses
   -p, -proxy string[]       list of http/socks5 proxy to use (comma separated or file input)
   -pi, -proxy-internal      proxy all internal requests
   -ldf, -list-dsl-function  list all supported DSL function signatures
   -tlog, -trace-log string  file to write sent requests trace log
   -elog, -error-log string  file to write sent requests error log
   -version                  show nuclei version
   -hm, -hang-monitor        enable nuclei hang monitoring
   -v, -verbose              show verbose output
   -profile-mem string       optional nuclei memory profile dump file
   -vv                       display templates loaded for scan
   -svd, -show-var-dump      show variables dump for debugging
   -ep, -enable-pprof        enable pprof debugging server
   -tv, -templates-version   shows the version of the installed nuclei-templates
   -hc, -health-check        run diagnostic check up

UPDATE:
   -up, -update                      update nuclei engine to the latest released version
   -ut, -update-templates            update nuclei-templates to latest released version
   -ud, -update-template-dir string  custom directory to install / update nuclei-templates
   -duc, -disable-update-check       disable automatic nuclei/templates update check

STATISTICS:
   -stats                    display statistics about the running scan
   -sj, -stats-json          display statistics in JSONL(ines) format
   -si, -stats-interval int  number of seconds to wait between showing a statistics update (default 5)
   -mp, -metrics-port int    port to expose nuclei metrics on (default 9092)

CLOUD:
   -auth                  configure projectdiscovery cloud (pdcp) api key
   -cup, -cloud-upload    upload scan results to pdcp dashboard
   -sid, -scan-id string  upload scan results to given scan id

AUTHENTICATION:
   -sf, -secret-file string[]  path to config file containing secrets for nuclei authenticated scan
   -ps, -prefetch-secrets      prefetch secrets from the secrets file


EXAMPLES:
Run nuclei on single host:
   $ nuclei -target example.com
```

```
Run nuclei with specific template directories:
   $ nuclei -target example.com -t http/cves/ -t ssl

Run nuclei against a list of hosts:
   $ nuclei -list hosts.txt

Run nuclei with a JSON output:
   $ nuclei -target example.com -json-export output.json

Run nuclei with sorted Markdown outputs (with environment variables):
   $ MARKDOWN_EXPORT_SORT_MODE=template nuclei -target example.com -markdown-export nuclei_report/

Additional documentation is available at: https://docs.nuclei.sh/getting-started/running
```

Running Nuclei

See https://docs.projectdiscovery.io/tools/nuclei/running for details on running Nuclei

Using Nuclei From Go Code

Complete guide of using Nuclei as Library/SDK is available at godoc

Resources

You can access the main documentation for Nuclei at https://docs.projectdiscovery.io/tools/nuclei/, and learn more about Nuclei in the cloud with ProjectDiscovery Cloud Platform

See https://docs.projectdiscovery.io/tools/nuclei/resources for more resources and videos about Nuclei!

Credits

Thanks to all the amazing community contributors for sending PRs and keeping this project updated. ❤️

If you have an idea or some kind of improvement, you are welcome to contribute and participate in the Project, feel free to send your PR.