ProjectDiscovery

Getting Started with Nuclei DAST and Global Templates

Dhiyaneshwaran
Template Engineer

WHO WE ARE

ProjectDiscovery: Forged in the open with a **global, offensive community**

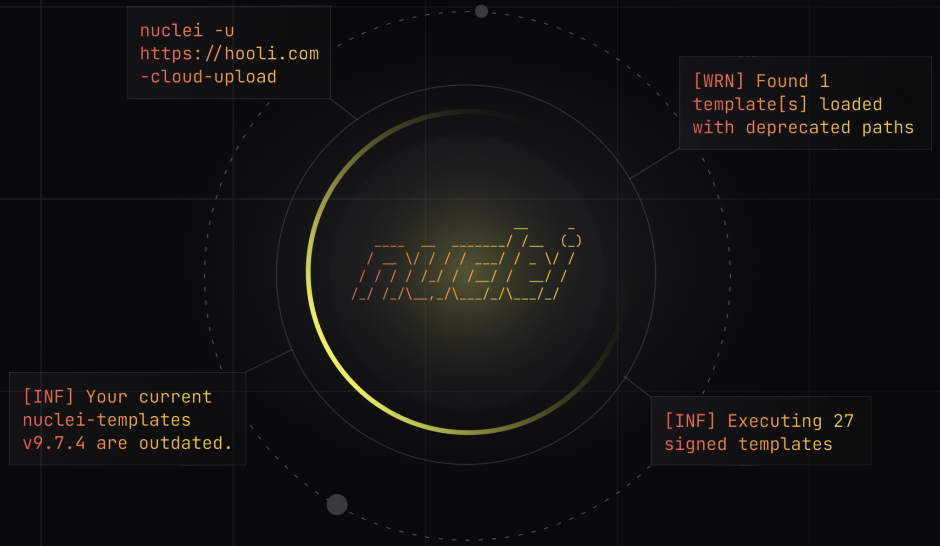
2020
Founded by the world's top bug bounty hunters

20+
Open source tools

90k+
GitHub stars

100k+
Global community members

Nuclei: a modern scanner for **exploitable vulnerabilities**



Run at Scale

Scale any checks to thousands of hosts with Nuclei Engine

Modular & Transparent

Nuclei's YAML-based template architecture offers **maximum flexibility**

Community Powered

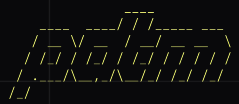
Hundreds of new templates contributed every month from our global community

Prerequisites

PREREQUISITES

Install GO language and set up GOPATH

- Download and Install Go (Linux, macOS, Windows)
- Set up GOPATH environment variable
- Update shell configuration (bashrc, zshrc)



Install ProjectDiscovery Tool Manager

- Install PDTM using Go
- Verify installation with pdtm command

v0.0.9

Download the PDTM Binary Directly

- https://github.com/projectdiscovery/pdtm/releases

What are DAST Nuclei Templates?

- DAST Nuclei Templates are YAML-based configuration files used with Nuclei to dynamically test live applications and services for vulnerabilities.
- They define requests, payloads, and conditions to detect security issues such as XSS, SQLi, misconfigurations, and more during runtime, enabling efficient and automated dynamic security testing.

```yaml
id: fuzz-reflection-xss

info:
  name: Basic Reflection Potential XSS Detection
  author: pdteam
  severity: low

http:
  - pre-condition:
      - type: dsl
        dsl:
          - 'method == "GET"'       # only run if method is
GET payloads:
      reflection:
        - "6842'\"><9967"

    stop-at-first-match: true
    fuzzing:
      - part: query
        type: postfix
        mode: single
        fuzz:
          - "{{reflection}}"

    matchers-condition: and
    matchers:
      - type: word
        part: body
        words:
          - "{{reflection}}"

      - type: word
        part: header
        words:
          - "text/html"
```

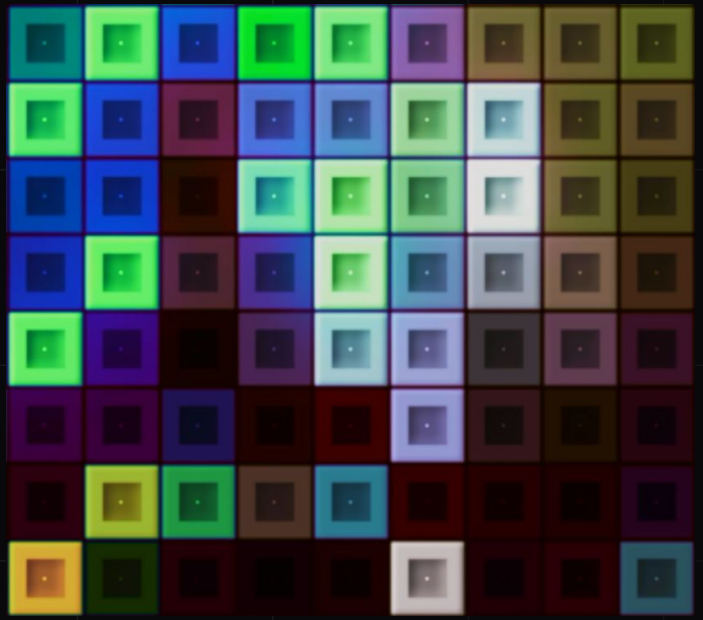
DAST : FUZZING FOR UNKNOWN VULNERABILITIES

Automate the boring parts of manual bug bounty hunting and speed up the overall process by 100x.

Fuzz every part of an HTTP request (Cookie, Header, Path, Body) with an abstracted Key-Value interface.

Supports XML, JSON, Multipart, and URLEncoded request bodies, including nested fields.

Supported inputs include Burp Suite saved items, Swagger, OpenAPI, Postman, Katana, and Proxify.



What Exactly are Global Matchers?

- Global matchers are basically matchers that operate on a global level. Instead of being tied to a specific request in a single template, they automatically apply to all HTTP responses received during a scan.

- Whether you're scanning for misconfigurations, secrets, or vulnerabilities, global matchers let you define your logic once and reuse it across all templates.

```yaml
http:

  - global-matchers: true

    matchers-condition: or
    matchers:
      - type: regex
        name: asymmetric_private_key
        regex:
          - '-----BEGIN ((EC|PGP|DSA|RSA|OPENSSH) )?PRIVATE KEY( BLOCK)?-----'
        part: body
```

Any Questions?

Thank You