EasyChair Preprint

№ 10035

Streamlining Network Penetration Testing: Overcoming the Challenges of Manual Effort and Inefficient Tools (NetPenTest)

Priyanka Gupta Gupta, Md Mahbub Hasan, Manish Kumar, Abu Sams Md Nakib, Mohammad Ashad Khan and Hasna Hena

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

May 9, 2023

Streamlining Network Penetration Testing: Overcoming the Challenges of Manual Effort and Inefficient Tools (NetPenTest)

Priyanka Gupta[1,a)], MD Mahbub Hasan[2,b)], Abu Sams MD Nakib[2,c)], Mohammd Ashad Khan[2,d)], Manish[2,e)] and Hasna Hena[2,f)]

[1]Assistant professor,Lovely Professional University,Phagwara, Punjab
[2]School of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab

[a)] Corresponding author: apriyanka.21789@lpu.co.in
[b)] mahbub.11900179@lpu.in
[c)] samsnakib4@gmail.com
[d)] ashad.ak786@gmail.com
[e)] manishsamrat2905@gmail.com
[f)] hheena1999@gmail.com

Abstract. Network security professionals face significant challenges in their daily work, from time-consuming repetitive tasks to the difficulty of keeping track of different tools and Information. These challenges can lead miss security threats and vulnerabilities. Inefficient processes can hamper effective network security testing. Addressing this problem requires a tool that can streamline the work of network security professionals. This paper presents an automated security testing service that uses Nuclei to scan web applications and generate detailed security reports. Through the utilization of Docker containerization, the system can execute dynamic security scans that accurately identify potential security threats without generating any false positives, helping end-users from the burden of manual security testing. Additionally, the system provides a user-friendly interface that allows users to select or create workflows and suggests different modules for creating their own workflow. The automated security testing service is designed to help network security professionals improve the security of their web applications by simplifying and automating the security testing process, saving time and reducing the risk of missed security threats and vulnerabilities.

INTRODUCTION

In today’s growing and connected world, organizations face a constant threat from cyber-attack. Increasing technology and the internet make the living style of people simple but on the other hand, it led to an increase in cyber-attacks, and increasing the internet makes the system more vulnerable to cyber-attack [1]. To overcome this organizations should focus on the security of their system. However, maintaining network security can be a complex and time taking task and require a significant amount of effort and resources. In this context, there is an urgent need for effective and user-friendly network penetration tools that can help an organization identify and mitigate vulnerabilities [2].

The network penetration tool described in this research paper has been designed to identify vulnerabilities by incorporating advanced features that make it easy to use and time efficient. This tool has multitasking capabilities, task scheduling capabilities, user-friendly interface, and report generation features making it an ideal solution for organizations of all sizes. Furthermore, the tool incorporates artificial intelligence to provide critical alerts and generate an assistant, allowing users to respond to security incidents in real time. With the ability of all these the tool offers a comprehensive solution for network security, providing organizations with the tools they need to secure their systems and protect their data.

BACKGROUND

A. Penetration testing

Penetration testing is the practice of examining a computer system, network, or web application to uncover any security vulnerabilities that may exist and could be exploited by an attacker. This process is essential for ensuring the security of organizations' digital assets and mitigating potential cyber threats.This process is essential for ensuring the security of organizations' digital assets and mitigating potential cyber threats. The primary aim of conducting penetration testing is to detect and address security weaknesses before they can be taken advantage of by malicious hackers. This proactive

approach to cybersecurity helps organizations to enhance their overall security posture and prevent cyber attacks and data breaches. Penetration testing can be conducted using various techniques and methodologies. [3] [4].
Automation can be used in various stages of the penetration testing process, including planning, reconnaissance, vulnerability scanning, exploitation, and post-exploitation. Automation can help improve the efficiency and effectiveness of penetration testing by automating repetitive tasks, providing consistent results, and reducing the risk of human error [5] [6].

B. Process of Automate Pen-Testing

Plan and define the scope: Defining the test's scope and identifying the apps or systems to be evaluated is the first stage in automating a penetration test. Establishing the goals, the ground rules, and the test limitations entails collaboration with the client [5] [6] [7].
Choose the automation tools: Choosing the appropriate tools for the various pen-test phases is the next step. They might include attack frameworks, vulnerability scanners, and tools for password cracking, among other things. Choosing the best tool for the job is essential when testing systems and applications [5] [6] [7].
Develop automation scripts: The creation of automation scripts to execute the chosen tools is the next stage. The execution of the tools and the gathering of the results will be automated by these scripts. Automation features included into some pen-testing frameworks, like Axiom, Metasploit, can be used to speed up the testing procedure [5] [6] [7].
Run the automation scripts: The various phases of the pen-test may be carried out using the automation scripts after they have been created. This might entail doing vulnerability scans, exploitation, and trying to access systems or applications.
Analyze the results: The outputs of the automation scripts need to be examined in order to find vulnerabilities and potential attack routes. This might entail undertaking further manual testing to confirm the outcomes as well as carefully evaluating the output of the tools and scripts [6] [7].
Generate a report: The pen-test results, including the vulnerabilities found, the possible effect of these vulnerabilities, and remedial suggestions, should all be summarized in a thorough report. This report should be easy to read, brief, and appropriate for the intended audience (e.g., technical vs. non-technical). These reports may be produced using the built-in reporting features of several pen-testing frameworks, such Metasploit [5] [6] [7].

C. Motivation and Problem Statement

Penetration tester, network security Professionals encounter several difficulties in their everyday job, including the need for a more effective and user-friendly solution, the requirement for manual testing for routine activities, and the problem of keeping track of various tools and information. These difficulties cause lengthy, ineffective processes that might miss security risks and vulnerabilities. A tool that simplifies network security experts' tasks is required in order to address these problems in a more effective and user-friendly manner [5].

ALGORITHM

1. **User Input:** The system will take task as input from the user. It can be a single task or multiple tasks.
2. **Task Distribution:** Once the task is received, the system will distribute the tasks to different worker computers based on the type of task.
3. **Task Scheduling:** There will be a task scheduling system that will schedule the tasks to be performed by the worker computers after a specific time period.
4. **Task Output:** The output of each task performed by the worker computers will be saved in a database using Apache Kafka.
5. **Monitor System:** There will be a monitor computer that will keep track of all the tasks and their progress.
6. **Report Generation:** Once the task is completed, the monitor computer will generate a report for each task using AI.
7. **Critical Findings:** If any critical findings are found during the task, the monitor computer will trigger a SMTP mail to the end user.

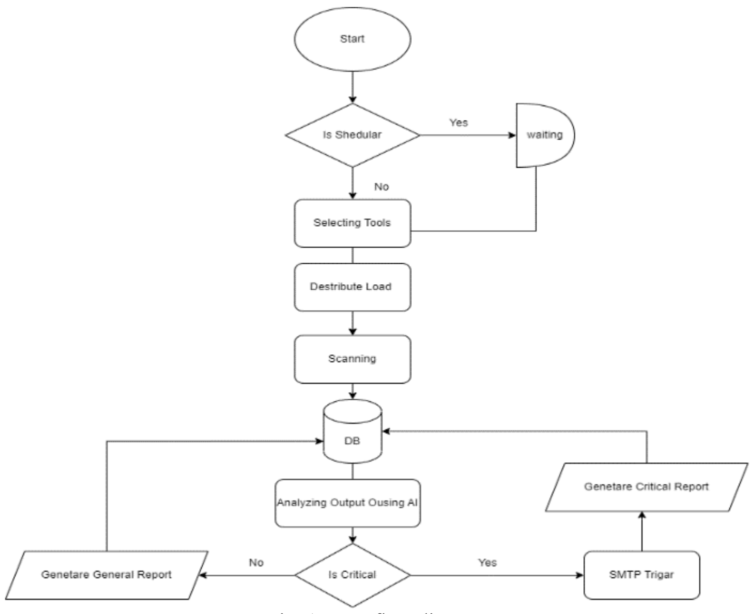


Fig. 1. Data flow diagram

SYSTEM OVERVIEW

In a microservice architecture, the system is broken down into multiple small, independently deployable services that work together to provide the overall functionality of the system [8].In the given Fig 2, the end user interacts with the website that is hosted on the web server. The web server validates user authentication related middleware and sends the request data to the master node through an API. The master node is responsible for decision-making and task management. It receives the request and decides on the appropriate task or group of tasks to be performed. The workload is then distributed to multiple worker nodes (hosted on Kubernetes) to perform the assigned task [9].The worker nodes communicate with the master node to get the necessary data from the database to complete the task. All the logs and data generated by the worker nodes are stored in the database. To handle the continuous generation of data, the system uses Apache Kafka, which is a distributed streaming platform that can handle large amounts of data [10] [11].Additionally, there is a monitor server that acts as a producer and has multiple subscriptions to produce different logs and outputs for the end-user, depending on their subscription. The monitor server is responsible for generating scan reports, triggering critical alerts, scheduling tasks, and other events. It is connected to the master node to perform these operations [6] [7].Overall, this microservice architecture allows for scalability, reliability, and flexibility in handling large amounts of data and user requests. Each service can be independently deployed, updated, and scaled to meet the system's changing demands [8].

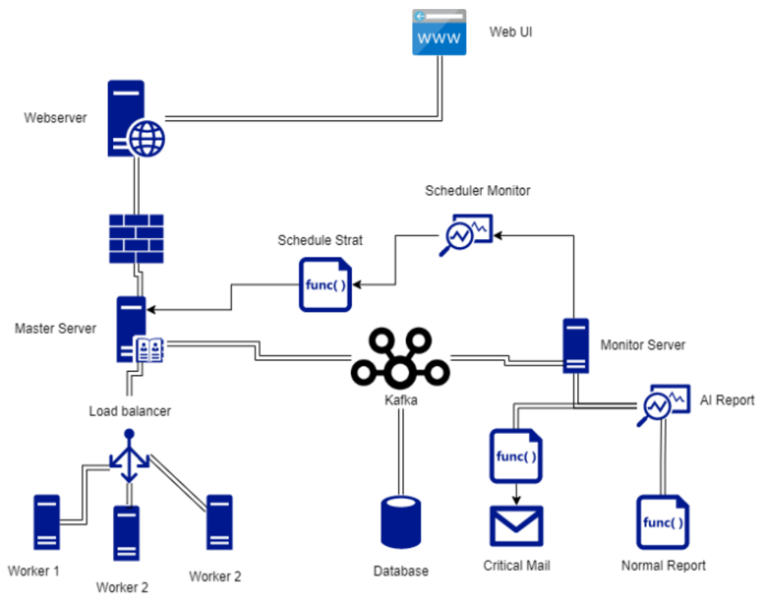


Fig. 2. System Overview Diagram

A. *Nuclei*

Nuclei is a powerful tool for security scanning and testing. Nuclei is used to scan across targets based on a template, its provide zero false positives result and 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 etc. The powerful templating system also makes it easy to customize Nuclei to suit specific testing needs, allowing users to model a wide range of security checks [12].

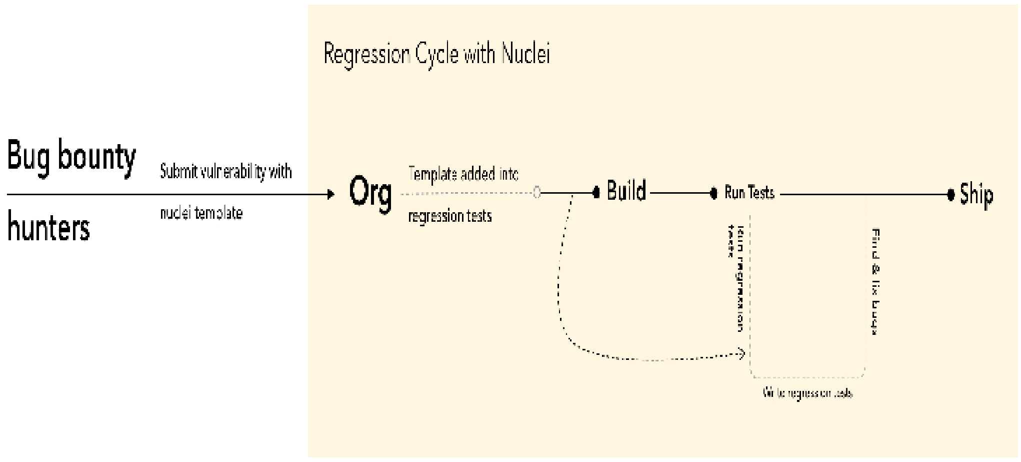

Regression Cycle with Nuclei
Bug bounty hunters
Submit vulnerability with nuclei template
Org
Template added into regression tests
Build
Run Tests
Ship
Find & fix bugs
Write regression tests

Fig. 3. Regression Cycle Nuclei

B. *Axiom*

Axiom is a framework that allows efficiently work with multiple cloud environments and build repeatable infrastructure for offensive and defensive security purposes. It achieves this by pre-installing the desired tools onto a base image, which can then be used to deploy new instances quickly. Axiom enables us to distribute scans of many different tools simultaneously by using that we can perform scans of a large set of targets across instances within minutes, and obtain results extremely quickly.

Overall, Axiom provides a powerful and flexible platform for security professionals to build, deploy, and manage their infrastructure across multiple cloud environments, with a particular emphasis on offensive and defensive security. [13].

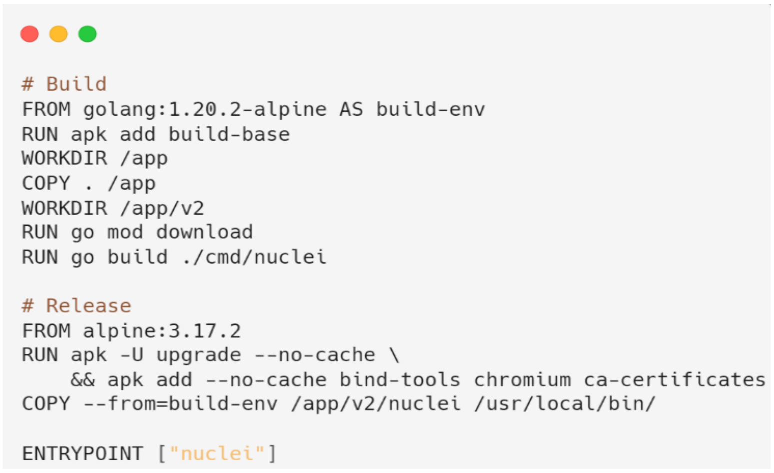
USED METHODOLOGIES

A. *Docker Containerization*

Docker containerization is a method of packaging and deploying software applications into lightweight, portable containers. It provide a consistent runtime environment for applications, regardless of the underlying operating system or infrastructure. We have created an images, which is essentially snapshots of a file system and a set of instructions for how to run the application. The Docker engine uses these images to create containers, which are then run on our kubernetes hosts worker pod. Containerization provides several benefits, including improved portability, scalability, and resource utilization. With Docker, we can build applications that can run anywhere, without worrying about dependencies or compatibility issues. Additionally, containers can be easily scaled up or down depending on demand. Docker container are isolated from one another, so it can make more efficient use of system resources [14].

B. *Configuring Tools*

In order to perform effective security testing, it is essential to configure security tools properly. This is particularly true for dynamic security attacking tools, as improperly configured tools may not be able to detect vulnerabilities with the necessary level of accuracy and precision. To address this issue, Docker images are preconfigured with all the necessary configurations, shell scripts, and other scripts that are required to pass relevant parameters to configure these tools correctly. By preconfiguring Docker images in this way, users can be confident that the security testing tools they use are properly configured, and that they are obtaining accurate and reliable results.

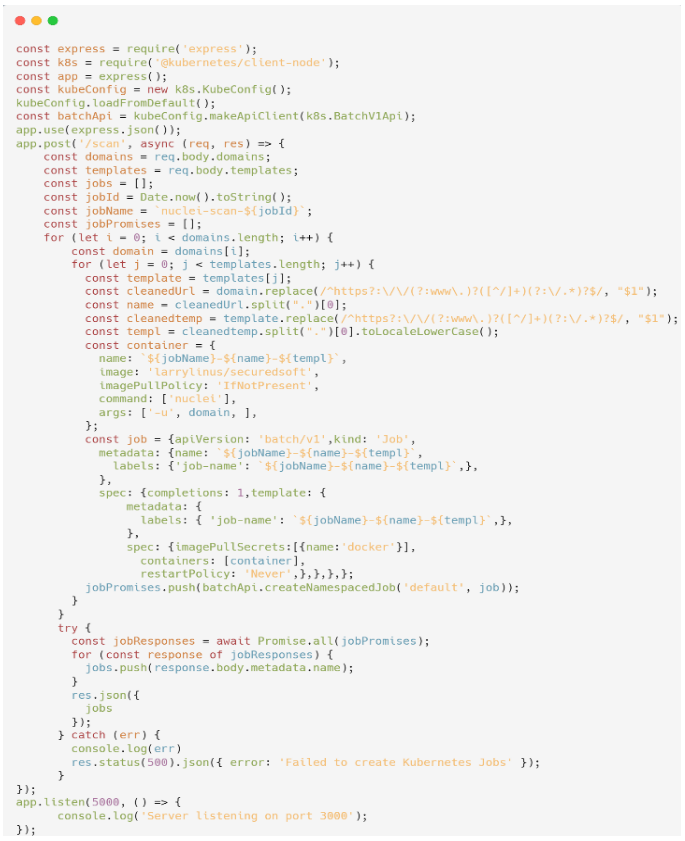


```
# Build
FROM golang:1.20.2-alpine AS build-env
RUN apk add build-base
WORKDIR /app
COPY . /app
WORKDIR /app/v2
RUN go mod download
RUN go build ./cmd/nuclei

# Release
FROM alpine:3.17.2
RUN apk -U upgrade --no-cache \
    && apk add --no-cache bind-tools chromium ca-certificates
COPY --from=build-env /app/v2/nuclei /usr/local/bin/

ENTRYPOINT ["nuclei"]
```

Fig. 4. Docker File for Nuclei

A. *Configuring Nuclei Tool*

We will build a Docker file containing instructions to build and release a Docker image that includes the Nuclei tool for vulnerability scanning's.The first part of the Docker file, marked as "build", uses the Golang 1.20.1-alpine image as the base image and sets it up as a build environment. The Alpine package manager is used to add the build-base package, which is a collection of tools and libraries needed for building packages in Alpine Linux. The working directory is set to /app, and the contents of the current directory are copied to the /app directory in the image. The working directory is then changed to /app/v2, and the go mod download command is run to download the dependencies specified in the go.mod file. Finally, the go build command is run to build the nuclei binary in the /app/v2/cmd/nuclei directories.The second part of the Docker file, marked as "release", uses the Alpine 3.17.2 image as the base image and installs the bind-tools, chromium, and ca-certificates packages. The bind-tools package provides DNS utilities, chromium provides a headless browser for scanning web applications, and ca-certificates provides SSL certificates. The nuclei binary that was built in the previous stage is then copied from the build environment to the /usr/local/bin directory in the release environment.The ENTRYPOINT command specifies that the default command to run when the container starts is the nuclei tool.



```
const express = require('express');
const k8s = require('@kubernetes/client-node');
const app = express();
const kubeConfig = new k8s.KubeConfig();
kubeConfig.loadFromDefault();
const batchApi = kubeConfig.makeApiClient(k8s.BatchV1Api);
app.use(express.json());
app.post('/scan', async (req, res) => {
    const domains = req.body.domains;
    const templates = req.body.templates;
    const jobs = [];
    const jobId = Date.now().toString();
    const jobName = `nuclei-scan-${jobId}`;
    const jobPromises = [];
    for (let i = 0; i < domains.length; i++) {
        const domain = domains[i];
        for (let j = 0; j < templates.length; j++) {
          const template = templates[j];
          const cleanedUrl = domain.replace(/^https?:\/\/(?:www\.)?([^/]+)(?:\/.*)?$/, "$1");
          const name = cleanedUrl.split(".")[0];
          const cleanedtemp = template.replace(/^https?:\/\/(?:www\.)?([^/]+)(?:\/.*)?$/, "$1");
          const templ = cleanedtemp.split(".")[0].toLocaleLowerCase();
          const container = {
            name: `${jobName}-${name}-${templ}`,
            image: 'larrylinus/securedsoft',
            imagePullPolicy: 'IfNotPresent',
            command: ['nuclei'],
            args: ['-u', domain, ],
          };
          const job = {apiVersion: 'batch/v1',kind: 'Job',
            metadata: {name: `${jobName}-${name}-${templ}`,
              labels: {'job-name': `${jobName}-${name}-${templ}`,},
            },
            spec: {completions: 1,template: {
                metadata: {
                  labels: { 'job-name': `${jobName}-${name}-${templ}`,},
                },
                spec: {imagePullSecrets:[{name:'docker'}],
                  containers: [container],
                  restartPolicy: 'Never',},},},};
          jobPromises.push(batchApi.createNamespacedJob('default', job));
        }
      }
      try {
        const jobResponses = await Promise.all(jobPromises);
        for (const response of jobResponses) {
          jobs.push(response.body.metadata.name);
        }
        res.json({
          jobs
        });
      } catch (err) {
        console.log(err)
        res.status(500).json({ error: 'Failed to create Kubernetes Jobs' });
      }
});
app.listen(5000, () => {
      console.log('Server listening on port 3000');
});
```

Fig. 5. Main Job Distribution Program

C. *Configuring Kubernetes & Job Distribution*

The program [fig 4] first imports the required dependencies, including the Express.js framework for building web applications and the Kubernetes Node.js client library. It also loads the Kubernetes configuration from the default configuration file.

Then, it defines a POST endpoint at /scan to handle incoming HTTP requests. The endpoint expects two query parameters: domains and templates, which are used to specify the domains to scan and the templates to use for the scans, respectively.

For each domain and template combination, the program creates a Kubernetes Job using the BatchV1Api and the provided configuration. The jobid of each created job is returned to the client as a JSON response [15].

D. *Outputs*

Job id to Pod id

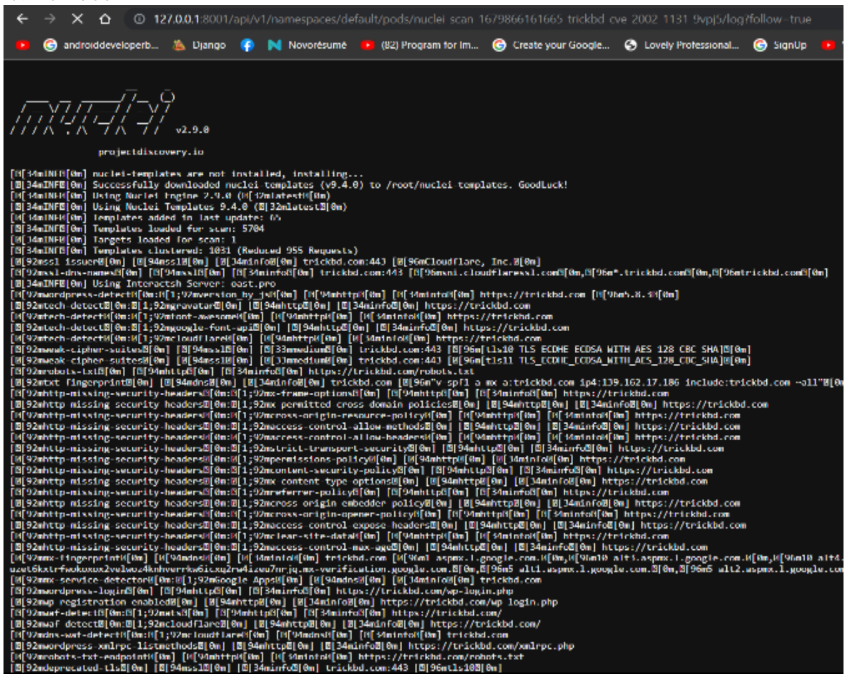
End points : /api/v1/namespaces/default/pods?labelSelector=job-name=xxx-xxx-xxx
Method : GET Content Type: Application Json

Fig. 6. Reaponse Containing Pod Id
API endpoint for a GET request that retrieves information about Kubernetes pods running in the default namespace with a specific label selector. The endpoint URL is "/api/v1/namespaces/default/pods", and the label selector is "job-name=xxx-xxx-xxx".

The request is expected to have a content type of "Application/Json", which means that the response will be in JSON format. The response will contain information about the pods that match the label selector, such as their name, status, and metadata.

Pod id to Pod log

End points : /apis/batch/v1/namespaces/default/jobs/podid-xxx-xxx-xxx /log
Method : GET
Content Type: Application Json


127.0.0.1:8001/api/v1/namespaces/default/pods/nuclei scan 1679866161665 trickbd cve 2002 1131 9vpj5/log?follow=true
androiddeveloperb...
Django
Novorésumé
(82) Program for Im...
Create your Google...
Lovely Professional...
SignUp

```
                                v2.9.0

            projectdiscovery.io

[▯[34mINF▯[0m] nuclei-templates are not installed, installing...
[▯[34mINF▯[0m] Successfully downloaded nuclei templates (v9.4.0) to /root/nuclei-templates. GoodLuck!
[▯[34mINF▯[0m] Using Nuclei Engine 2.9.0 (▯[92mlatest▯[0m)
[▯[34mINF▯[0m] Using Nuclei Templates 9.4.0 (▯[32mlatest▯[0m)
[▯[34mINF▯[0m] Templates added in last update: 65
[▯[34mINF▯[0m] Templates loaded for scan: 5704
[▯[34mINF▯[0m] Targets loaded for scan: 1
[▯[34mINF▯[0m] Templates clustered: 1031 (Reduced 955 Requests)
[▯[92mssl issuer▯[0m] [▯[94mssl▯[0m] [▯[34minfo▯[0m] trickbd.com:443 [▯[96mCloudflare, Inc.▯[0m]
[▯[92mssl-dns-names▯[0m] [▯[94mssl▯[0m] [▯[34minfo▯[0m] trickbd.com:443 [▯[96msni.cloudflaressl.com▯[0m,▯[96m*.trickbd.com▯[0m,▯[96mtrickbd.com▯[0m]
[▯[34mINF▯[0m] Using Interactsh Server: oast.pro
[▯[92mwordpress-detect▯[0m:▯[1;92mversion_by_js▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com [▯[96m5.8.4▯[0m]
[▯[92mtech-detect▯[0m:▯[1;92mgravatar▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mtech-detect▯[0m:▯[1;92mfont-awesome▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mtech-detect▯[0m:▯[1;92mgoogle-font-api▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mtech-detect▯[0m:▯[1;92mcloudflare▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mweak-cipher-suites▯[0m] [▯[94mssl▯[0m] [▯[33mmedium▯[0m] trickbd.com:443 [▯[96m[tls10 TLS ECDHE ECDSA WITH AES 128 CBC SHA]▯[0m]
[▯[92mweak-cipher-suites▯[0m] [▯[94mssl▯[0m] [▯[33mmedium▯[0m] trickbd.com:443 [▯[96m[tls11 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA]▯[0m]
[▯[92mrobots-txt▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com/robots.txt
[▯[92mtxt fingerprint▯[0m] [▯[94mdns▯[0m] [▯[34minfo▯[0m] trickbd.com [▯[96m"v spf1 a mx a:trickbd.com ip4:139.162.17.186 include:trickbd.com ~all"▯[0
[▯[92mhttp-missing-security-headers▯[0m:▯[1;92mx-frame-options▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mhttp missing security headers▯[0m:▯[1;92mx permitted cross domain policies▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mhttp-missing-security-headers▯[0m:▯[1;92mcross-origin-resource-policy▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mhttp-missing-security-headers▯[0m:▯[1;92maccess-control-allow-methods▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mhttp-missing-security-headers▯[0m:▯[1;92maccess-control-allow-headers▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mhttp-missing-security-headers▯[0m:▯[1;92mstrict-transport-security▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mhttp-missing-security-headers▯[0m:▯[1;92mpermissions-policy▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mhttp-missing-security-headers▯[0m:▯[1;92mcontent-security-policy▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mhttp missing security headers▯[0m:▯[1;92mx content type options▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mhttp-missing-security-headers▯[0m:▯[1;92mreferrer-policy▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mhttp missing security headers▯[0m:▯[1;92mcross origin embedder policy▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mhttp-missing-security-headers▯[0m:▯[1;92mcross-origin-opener-policy▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mhttp missing security headers▯[0m:▯[1;92maccess control expose headers▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mhttp-missing-security-headers▯[0m:▯[1;92mclear-site-data▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mhttp-missing-security-headers▯[0m:▯[1;92maccess-control-max-age▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com
[▯[92mmx-fingerprint▯[0m] [▯[94mdns▯[0m] [▯[34minfo▯[0m] trickbd.com [▯[96m1 aspmx.l.google.com.▯[0m,▯[96m10 alt1.aspmx.l.google.com.▯[0m,▯[96m10 alt4.
uzel6kxtrfwokoxox2velwoz4knhverrkw6icxq2rw4izeu7nrjq.mx-verification.google.com.▯[0m,▯[96m5 alt1.aspmx.l.google.com.▯[0m,▯[96m5 alt2.aspmx.l.google.com
[▯[92mmx-service-detector▯[0m:▯[1;92mGoogle Apps▯[0m] [▯[94mdns▯[0m] [▯[34minfo▯[0m] trickbd.com
[▯[92mwordpress-login▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com/wp-login.php
[▯[92mwp registration enabled▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com/wp login.php
[▯[92mwaf-detect▯[0m:▯[1;92mats▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com/
[▯[92mwaf detect▯[0m:▯[1;92mcloudflare▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com/
[▯[92mdns-waf-detect▯[0m:▯[1;92mcloudflare▯[0m] [▯[94mdns▯[0m] [▯[34minfo▯[0m] trickbd.com
[▯[92mwordpress xmlrpc listmethods▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com/xmlrpc.php
[▯[92mrobots-txt-endpoint▯[0m] [▯[94mhttp▯[0m] [▯[34minfo▯[0m] https://trickbd.com/robots.txt
[▯[92mdeprecated-tls▯[0m] [▯[94mssl▯[0m] [▯[34minfo▯[0m] trickbd.com:443 [▯[96mtls10▯[0m]
```

Fig. 5. Report Generated by Nuclei

This API endpoint is part of the Kubernetes Batch API, which is used for running batch jobs on the cluster. The "/log" endpoint is specifically used to retrieve the log data for the pod associated with a particular batch job [Fig 5]. Previously we have created a Service account named api-access and give it a role in different pods and pods logs to watch get and list. If we generate access tokens from that service account, by using this we can retrieve data from any of our pods.

E. *Machine Learning Process*

From the output generated by nuclei It first removes well-known false positives and then passes the remaining security issues through a K-Nearest Neighbors algorithm to classify them as false positives or not. The dataset that was utilized for training the machine learning model underwent a thorough manual verification process to ensure that false positives were eliminated. The system is not 100% accurate but produces correct reports more than 80% of the time. End users can also mark security issues as false positives or exclude them from reports to eliminate recurring overhead [16].

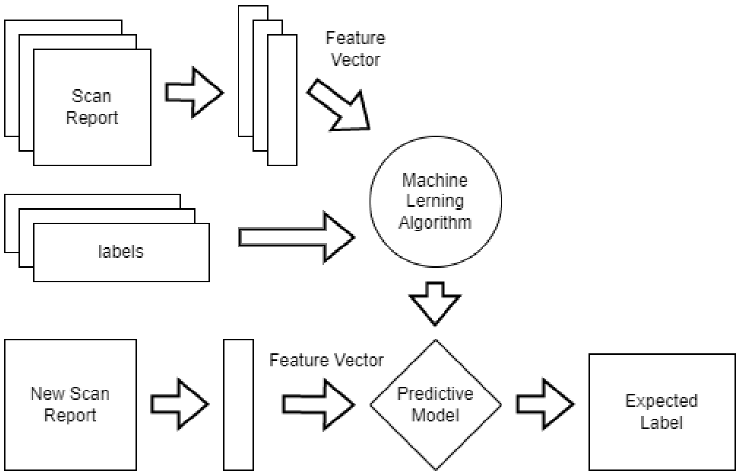


Fig. 7. Supervise Learning Model.

F. *Front End Web Application*

The network scanning and discovery application that we are developing. Our application will provide multiple pre-defined workflows for users to choose from when scanning their network. When a user adds a domain to our front-end, the application will automatically scan using our basic discovery tool or create a map of the network. The AI-powered GPT-3 model will then suggest appropriate workflows based on the initial scan results.

In addition, users can select templates from our user interface dropdown menu to create their own custom workflows for the scan. The AI will assist users by suggesting appropriate templates based on the initial scan results, which may include technology discovery and services inside the domain server. There will also be an option for users to add their own Nuclei template. If a user creates a nuclei template, they can upload it, and our system will check the template code for correctness. Once verified, the template will be added to the workflow creation template menu for use in future scans.

Our system will also have the ability to update Nuclei templates automatically. Overall, our application will provide users with a comprehensive set of tools for scanning their network and discovering information about their systems and services. It will be user-friendly and flexible, with the AI providing assistance where needed.

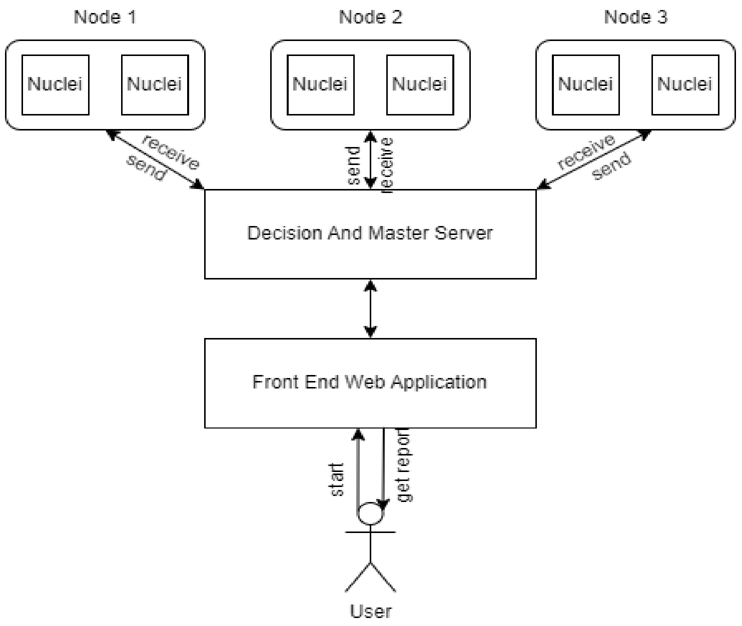


Fig. 8. User Interaction with System

FUTURE PLAN

A. We will create a web browser base plugin. The plugin will collect data from various hidden pages and routes. Then it will send the data to our main scanning server. Our server will then scan the code for potential security vulnerabilities. It might include bad coding practices and hidden routes that could be exploited. The plugin and our proposed tool will work together to ensure our users have the most secure web page possible.

CONCLUSION

This research paper we presents a system that we have developed to address the challenges faced by network security professionals and pentesters. We have detailed the various technologies and methodologies that were used for the implementation of our system using Nuclei, Docker, and Kubernetes etc. Our system offers a user-friendly front-end interface in which users just need to provide the domain they wish to scan for security vulnerabilities. The system then runs the scans in the background, and it will provide comprehensive security scan reports with very less false positives. The users can track the progress of the scan and receive notifications when the scan is complete. The system is also capable of sending critical mail about findings to the users. It is also possible to chose a custom workflow or create a new workflow for the scan. In summary, our application provides users with a comprehensive set of tools for scanning their network for security vulnerabilities. It is user-friendly and flexible and provide AI assistance where needed. Furthermore, we have examined possible directions for future development aimed at advancing our system's capabilities. Overall, our system helps network security professionals and pen testers to save time and obtain more accurate results.

REFERENCES

[1] "Fortinet," 2023. [Online]. Available: https://global.fortinet.com/lp-en-ap-2023cloudsecurityreport. [Accessed 2023].

[2] I. Yurtseven and S. Bagriyanik, "A Review of Penetration Testing and Vulnerability Assessment in Cloud Environment," in *IEEE*, Istanbul, Turkey, 2020.

[3] H. Li, X. He, Y. Zhang and W. Guan, "Attack Detection in Cyber-Physical Systems Using Particle Filter: An Illustration on Three-Tank System," Tianjin, China, 2019.

[4] G. Jayasuryapal, P. M. Pranay, H. Kaur and Swati, "A Survey on Network Penetration Testing," in *IEEE*, London, United Kingdom, 2021.

[5] "OWASP," [Online]. Available: https://owasp.org/www-project-web-security-testing-guide/latest/3-The_OWASP_Testing_Framework/1-Penetration_Testing_Methodologies. [Accessed 2023].

[6] Y. Stefinko, A. Piskozub and R. Banakh, "Manual and automated penetration testing. Benefits and drawbacks. Modern tendency," in *IEEE*, Lviv, Ukraine, 2016.

[7] S. Semenov, Z. Liqiang and C. Weiling, "Penetration Testing Process Mathematical Model," in *IEEE*, Kharkiv, Ukraine, 2021.

[8] Y. Romani, O. Tibermacine and C. Tibermacine, "Towards Migrating Legacy Software Systems to Microservice-based Architectures: a Data-Centric Process for Microservice Identification," in *IEEE*, Honolulu, HI, USA, 2022.

[9] N. T. Nguyen and Y. Kim, "A Design of Resource Allocation Structure for Multi-Tenant Services in Kubernetes Cluster," in *IEEE*, Jeju Island, Korea, Republic of, 2022.

[10] "Kafka," Apache, [Online]. Available: https://kafka.apache.org/documentation/#implementation. [Accessed 2023].

[11] R. Shree, T. Choudhury, S. C. Gupta and P. Kumar, "KAFKA: The modern platform for data management and analysis in big data domain," in *IEEE*, Noida, India, 2018.

[12] ProjectDiscovery, 2020. [Online]. Available: https://github.com/projectdiscovery/nuclei.

[13] pry0cc, 2020. [Online]. Available: https://github.com/pry0cc/axiom.

[14] "Docker," Docker, [Online]. Available: https://docs.docker.com/get-started/02_our_app/. [Accessed 2023].

[15] "Kubernetes," CNCF, [Online]. Available: https://kubernetes.io/docs/concepts/workloads/controllers/job/.

[16] P. P. W. Pathirathna, V. A. I. Ayesha, W. A. T. Imihira, W. M. J. C. Wasala, N. Kodagoda and E. A. T. D. Edirisinghe, "Security testing as a service with docker containerization," in *2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, Malabe, Sri Lanka, 2018.

[17] M. A. Helmiawan, E. Julian, Y. Cahyan and A. Saeppani, "Experimental Evaluation of Security Monitoring and Notification on Network Intrusion Detection System for Server Security," in *IEEE*, Bengkulu, Indonesia, 2021.

[18] A. Confido, E. V. Ntagiou and M. Wallum, "Reinforcing Penetration Testing Using AI," in *IEEE*, Big Sky, MT, USA, 2022.

[19] J. Huang, Z. Mao, C. Xie and Y. Sun, "Distributed Photovoltaic Scenario Generation Based on Edge-Cloud Collaborative Architecture," in *IEEE*, Shanghai, China, 2022.