NETDEFENDER TOOLKIT

A Thesis

Submitted in partial fulfilment of the requirements

for the award of the Degree of B. Tech

IN

Computer Science & Engineering

BIRLA INSTITUTE OF TECHNOLOGY

MESRA-Off Campus
Deoghar-814142

APPROVAL OF THE GUIDE

Recommended that the thesis entitled “**NETDEFENDER TOOLKIT**” presented by **Harsh Vardhan Singh (BTECH/60304/21), Aaditya Kumar (BTECH/60308/21), Ranvir Ranvijay Singh (BTECH/60301/21)** under my supervision and guidance be accepted as fulfilling this part of the requirements for the award of Degree of Bachelors in Technology. To the best of my knowledge, the content of this thesis did not form a basis for the award of any previous degree to anyone else.

Date: Signature of Guide

DECLARATION CERTIFICATE

I certify that

a) The work contained in the thesis is original and has been done by myself under the general supervision of my supervisor.

b) The work has not been submitted to any other Institute for any other degree or diploma.

c) I have followed the guidelines provided by the Institute in writing the thesis.

d) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e) Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

f) Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Harsh Vardhan Singh	(BTECH/60304/21)
Aaditya Kumar	(BTECH/60308/21)
Ranvir Ranvijay Singh	(BTECH/60301/21)

CERTIFICATE OF APPROVAL

This is to certify that the work embodied in this thesis entitled **"NETDEFENDER TOOLKIT"**, is carried out by **Harsh Vardhan Singh (BTECH/60304/21), Aaditya Kumar (BTECH/60308/21), Ranvir Ranvijay Singh (BTECH/60301/21)** has been approved for the degree of **Bachelors in Technology** of Birla Institute of Technology, Mesra, Ranchi.

Date:

Place:

Internal Examiner **External Examiner**

ABSTRACT

The **NetDefender Toolkit** is a complete solution developed in order to bring protection through improving the security of cloud environments by implementing the most important functionalities in order to address emerging threats. The toolkit consists of three crucial modules: **IP Address Visualization Tool**, **Password Generator Tool** and **Nmap Scanner Tool.**

The IP Address Visualization Tool analyses network traffic, identifies source and destination IPs and maps them geographically, providing real-time monitoring through Wireshark and Google My Maps. This helps the network administrator easily identify traffic patterns and possible risks.

The Password Generator Tool allows for the generation of highly secure, random passwords tailored to user-defined specifications, bringing them resistance against brute-force and dictionary attacks. Finally, **The Nmap Scanner Tool**, with the powerful capabilities of Nmap, scans networks and host discovery, in addition to vulnerability assessment, through an easy-to-use interface suitable for users with varied skill levels.

The methodology integrates real-time data processing, geolocation mapping, and advanced security protocols for usability, performance, and reliability. Designed for cross-platform compatibility and scalability, the toolkit addresses the increasing demand for proactive security solutions. Its modular design with an innovative approach brings the gap between complex cloud-based cybersecurity tools and user-friendly functionality to fill a gap for both organizations and individuals.

ACKNOWLEDGEMENT

First, I would like to thank **Dr. Soumya Ray, Assistant Professor, CSE**. my guide for their mentorship, support, and valuable insights. Their guidance was instrumental in shaping the outcome of my Minor Project.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I would like to thank my Head of the Department **Dr. Sounak Paul** for his constructive criticism throughout my Minor Project.

I am highly indebted to Director **Dr. Aruna Jain** for the facilities provided to accomplish this Minor Project.

I am extremely grateful to my department staff members and friends who helped me in successful completion of this minor project.

DATE:

Harsh Vardhan Singh(BTECH/60304/21)

Aaditya Kumar (BTECH/60308/21)

Ranvir Ranvijay Singh(BTECH/60301/21)

CONTENTS

INTRODUCTION

In the digital age, where data drives modern enterprises and connectivity underpins daily operations, the need for sure and robust cloud based cybersecurity measures has never been more acute. With increasing reliance on cloud computing and interconnected systems within organizations, the expanding threat landscape and vulnerabilities being introduced continue to put sensitive information at risk and threaten to disrupt critical operations. The NetDefender Toolkit is specifically designed to solve this problem as a custom solution, with its modular and user-friendly platform that combines must-have tools for improving network security.

The NetDefender Toolkit consists of three main components, which include: **IP Address Visualization Tool**, **Password Generator Tool**, and **Nmap Scanner Tool**. Together, these modules provide comprehensive functionalities, allowing users to monitor, protect, and assess the security of their systems effectively.

Module 1: IP Address Visualization Tool

Network security can be initiated with the understanding of traffic flows and identification of potential risks. The IP Address Visualization Tool displays network activity as mapping source and destination IP addresses to their geographical locations using capture libraries like Wireshark and Python scripts to process the capture in real time. This functionality assists in the process of suspicious connections and understanding of data flow patterns as well as that of identifying possible origins of attacks. This data is represented through Google My Maps to present an intuitive and actionable view of a network's geographical footprint, which facilitates better decision-making in the mitigation and threat analyses.

Module 2: Password Generator Tool

Passwords will forever be the first point of defence in the digital world. Low strength or easily predictable credentials remain the primary pathway through which security breaches are executed. The tool here, the Password Generator Tool, bridges this vulnerability by creating highly secure customizable passwords. With Python as a core construction material, the tool guarantees resistance to brute-force and dictionary attacks. It allows users to define various parameters, including length, character types, and complexity, which means passwords will meet specific security requirements. The tool is lightweight and efficient and ensures privacy by generating passwords without transmitting any sensitive data, thus it can be used safely at both personal and organizational levels.

Module 3: Nmap Scanner Tool

Maintaining secure networks is heavily reliant on vulnerability assessment. The Nmap Scanner Tool extends the powerful capabilities of the open-source Nmap utility and makes it possible to perform network scans and analysis in a relatively simplified interface. This module supports various operations, such as host discovery, port scanning, and service detection. Users can choose from predefined scan options or input custom commands for tailored use cases. This utility delivers real-time output and provides the ability to download scan results, so usability varies from beginner to advanced users. It gives a user the power of automation to secure their network through identifying open ports, running services, and potential vulnerabilities.

Unified Solution to Modern Security Challenges:

The three modules combined in this toolkit constitute more than the sum of the parts because they guarantee a cloud based cybersecurity-focused, holistic approach. This modular design ensures scalability and flexibility toward covering different needs according to the type of user involved, whether network administrator or individual user. In doing so, the tool brings together functionalities aimed at monitoring, protection, and analysis.

In Concludes, the NetDefender Toolkit by filling the gap between advanced cloud based cybersecurity functionalities and user accessibility. It has made powerful tools intuitive enough to be used by an average user for the protection of his/her systems from ever-evolving threats, thus finding a place in modern digital security as a guiding light to navigate these complex waters.

MODULE 1: IP ADDRESS VISUALIZATION TOOL:

Introduction:-

The IP Address Visualization tool is effective both in analyzing network traffic as well as providing representation of the source of IP addresses in a geographic mode by providing this information by combining data processing, mapping as well as source and destination network packet analysis for network connections.

The Tool captures network packets with Wireshark and processes gathered info using Python with the help of open-source dataset of IP Addresses. Then, Google My Maps will display the output, which would have been almost a pictorial representation of IP address distribution. A user would be able to track network traffic patterns, discover possible risks, or even find out the direction of data trends using this method.

Key Features:

- **Live Network Traffic Capture** is the mechanism where packets are captured and analyzed at runtime with the help of Wireshark (tshark). It Captures all incoming and outgoing packets, allowing real-time monitoring of network activities. Use Case: Immediate detection of active connections as well as potential security threats.
- **Support for Pre-Saved Packet Files** Apart from live capturing of traffic, the toolkit also supports loading pre-saved .pcap files for analysis. The user is able to revisit previous network activities without necessarily needing a working internet connection. Can be used for forensic analysis or reviewing historical data during the audit.
- **Geographical Mapping:** This tool visualizes the IP address on the interactive map created using Google My Maps. It can map every IPs to their corresponding latitude and longitude. It identifies geographical sources of traffic that will help pinpoint potential security threats.
- **Security and Integrity of Data:** It maintains data privacy and integrity, like the handling of the .pcap files in a secure way. It Limits the exposure of sensitive network data during processing.

Related Work:-

A number of tools and frameworks have been developed for network traffic analysis and visualization. The IP Address Visualization Tool extends those already existing technologies and combines their strength to provide a comprehensive solution for network data visualization. Below are some related works that have inspired or contributed to the development of this project:

1. Wireshark: (tshark)

- Wireshark is a widely used open-source network protocol analyzer that allows users to capture and interactively browse the traffic running on a computer network.
- It provides a detailed, real-time view of network activity by capturing data packets as they travel across the network, which can then be analyzed in depth.

2. Google My Maps:

- Google My Maps is a service whereby you can create and share custom maps.
- Not exactly a network analysis tool, it is a very powerful tool for representing geospatial data, which comprises locating an IP address.

3.MaxMind GeoIP Dataset:

- GeoIP databases and tools are offered by MaxMind to map IP addresses to their geographical locations.
- Numerous applications, such as network analysis, content personalization, and fraud detection, make extensive use of it.

Requirement Specification for IP Address Visualization Tool:-

Introduction:

The IP Address Visualization Tool project analyzes network traffic, extracts IP addresses, and maps their geographical locations. Through the tool, network administrators or cybersecurity professionals can visualize the origin and destination of network connections and better understand traffic patterns and possible security threats in perspective.

Project Background:

Network traffic and cyber threats are increasing with each passing day. The integration of tools that provide analysis of traffic patterns with regard to potential threats is required. Traditionally, old packet analysis tools don't support the geographical visualization of sources. This project fills the gap by integrating the extraction of IP addresses, geolocation, and map-based visualization. It helps network administrators as well as cyber security teams monitor and analyze traffic effectively.

Purpose:

- **Improved Network Monitoring:** To assist cybersecurity experts and network managers in locating the sources and destinations of traffic.
- **Threat Detection:** To help identify malicious or suspicious traffic by emphasizing connections from odd places.
- **Improved Decision Making**: To support better decision-making in network security and optimization through geographical insights.

Functional Requirements:

1) **Packet Capture**:

   The tool must capture live network traffic using Wireshark (tshark) or load pre-saved `.pcap` files for analysis.

2) **Data Processing**:

   The tool should extract relevant IP addresses information from the captured network traffic and process it using a *Python script.*

3) **Geolocation Mapping**:

   Convert the processed data into *KML format* to visualize IP locations on Google My Maps.

4) **File Management**:

   Save captured network data as `.pcap` files for later use and allow loading previously saved files.

5) **Interactive Map Visualization**:

Enable users to import the *KML file* into Google My Maps and visualize destination IPs geographically.

Non-Functional Requirements

1. **Performance:**

   The tool should be able to handle huge files in the form of .pcap without too much time consumption.

2. **Usability:**

   It must be user-friendly along with step-by-step guidance for the non-expert user on how to use it.

3. **Security:**

   The captured and processed sensitive network data should be secure, and it must not access the packets by unauthorized people.

4. **Reliability:**

   It ensures successful geolocation and data visualization without numerous errors in the case of unstable networks.

5. **Compatibility:**

   It is easily deployable on a broad range of platforms, including Linux, Windows, and macOS

System Requirements

- **Hardware**:
  - Minimum 4 GB RAM
  - 10 GB available storage
  - Network interface capable of packet capture
- **Software**:
  - Operating System: Linux, Windows and MacOS

  - Wireshark installed for packet capturing
  - Python 3 with necessary libraries (e.g., pyshark, GeoIP, simplekml)
  - Google My Maps for visualization

User Requirement:

- Basic Knowledge: Users should have basic knowledge of network traffic, IP addresses, and Linux command-line usage.
- Access Privileges: Users need administrator/root privileges to capture live network traffic.

PROBLEM DESIGN:

METHODOLOGY

1. Packet Capture and Preprocessing:

Objective : To collect raw network traffic data.

Tool: Wireshark(tshark) for real-time network monitoring

Process:

- Users initiate packet capture on selected network interface
- Traffic is captured and saved in *pcap* format, so the user can make further analysis

Output: ".*pcap*" file contains captured network packets.

2. IP Geolocation:

Objective: Match the IP address extracted to geolocation information

Tools used: GeoIP databases (such as MaxMind GeoLite2).

Process:

- Look up each IP address for their GeoIP service.

- Extract the geolocation data for each of them, including country, city, latitude, and longitude.

Output: Dataset of IP address with corresponding geolocation information.

3. KML File Generation:

Objective: File format that is specifically for visualization on the map.

Tools: Using Python's *simplekml* library

Process:

- Geolocation data that is converted into a KML file format.
- The KML file has placemarks with metadata having location and relevant IP.

Output: Preparing a ready-to-be-used KML file for the map.

4. Data Visualization

Objective: To provide an intuitive geographical view of network traffic.

Tools Used: Google My Maps or Equivalent GIS tools.

Process:

- Import the KML file into Google My Maps.
- The respective IP addresses appear as clickable markers on a world-wide map.

Outcome: An interactive map to indicate the geographic spread of IP traffic.

5. Feedback and Error Handling

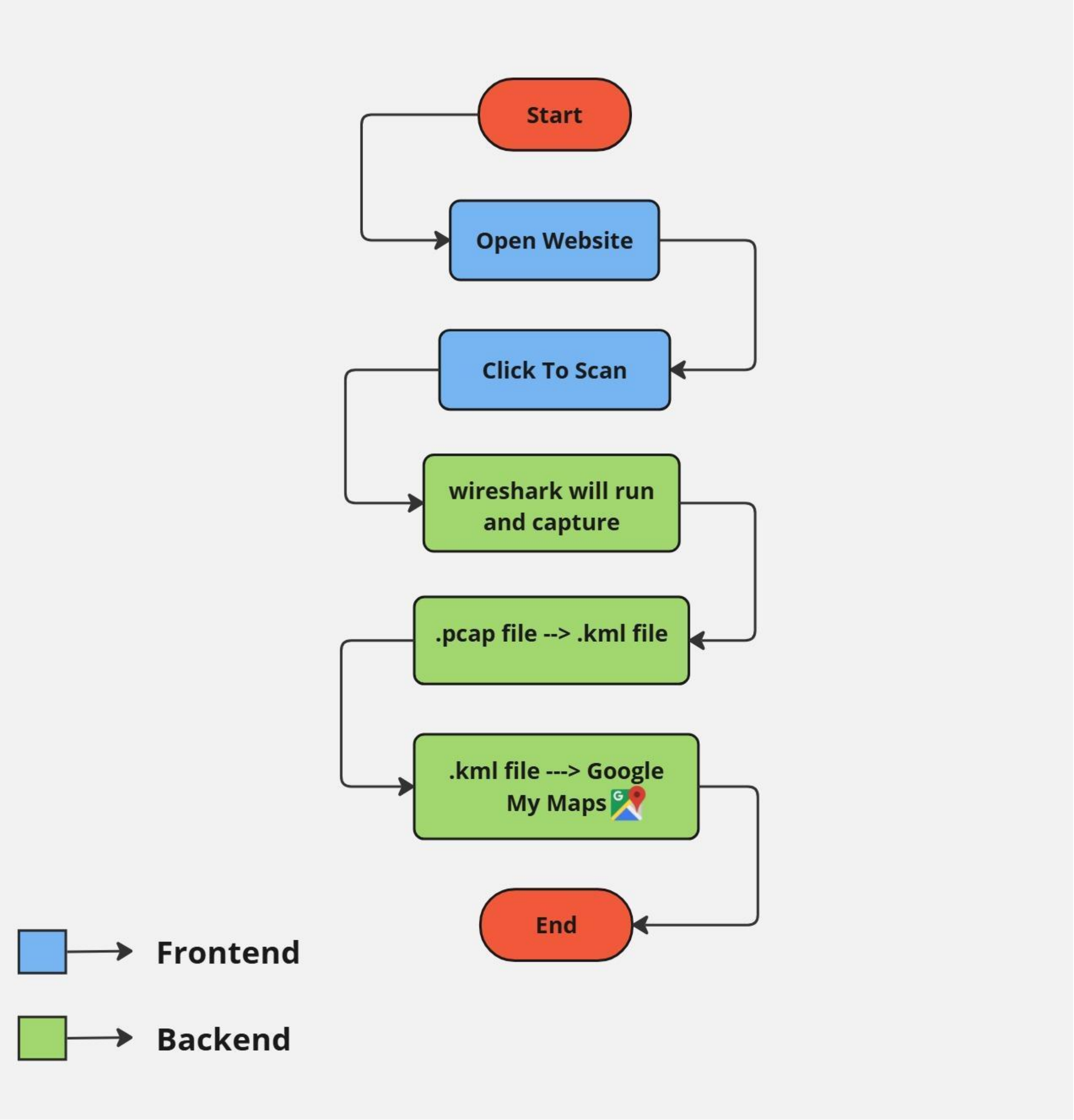
Objective: Ensure Accurate data, robust against errors.

Process:

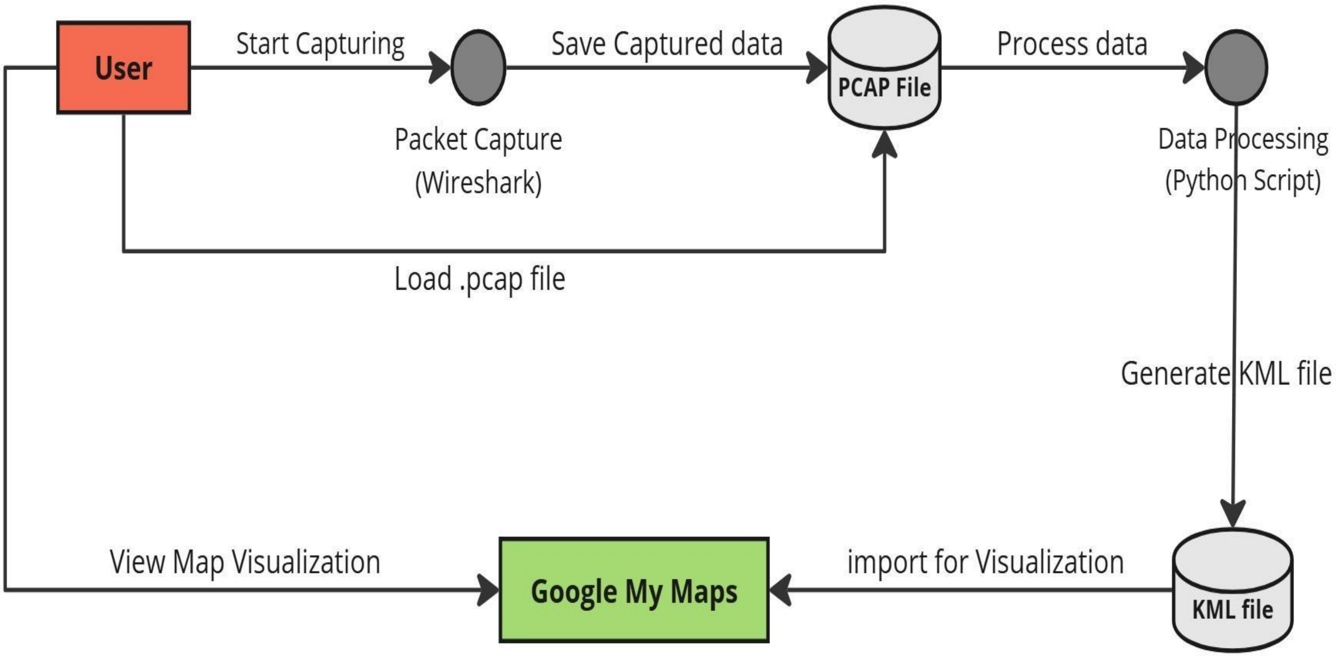
- Validate against known IP address ranges to catch private or invalid IP addresses.
- Prove feedback regarding errors (such as a failed geolocation request) and suggest corrections.

Output: Improved quality of data and user trust in the tool.

Diagrams: Flow Chart



Diagrams: Data Flow diagram



EXPERIMENTAL SETUP

Steps to Create the Model

1. Requirement Gathering and Analysis

- Determine the main objectives of the tool, (e.g., network traffic capture, IP extraction, geo-location, and mapping of the same on a geospatial map.)
- Functional requirements include performance, compatibility, and scalability

2. Architecture Design

- Outline the design of data flow in terms of diagrams such as DFD and flowcharts.
- Determine the modules that can be used:
  1. Packet Capture
  2. IP Extraction
  3. Geo-location
  4. KML Generation
  5. Map Visualization

3. Tools and Libraries Selection

Based on requirement, determine the software and libraries

- Wireshark for packet capture
- Python with scapy, pyshark, and GeoIP
- Google My Maps for visualization

4. Implementation of Core Components

- Packet Capture: Write scripts for packet capture or importing *.pcap files*
- IP Extraction: Using the extracted .pcap files extract source and destination IPs
- Geolocation: Query geolocation API or make use of the offline databases to fetch location data about IPs
- KML Generation: Convert geolocation data into KML files using Python libraries such as *simplekml.*

5. Module Implementation for Visualization

- Import the generated KML files into Google My Maps to display them in an interactive visual way.
- Test the visualization for correctness and clarity in data representation.

6. Integrate and Test the Modules

- To form an integrated system, combine modules.
- To test every module, do unit testing. Integrate all modules by doing integration testing on the whole system.

7. Optimize the Model

- The program should smoothly handle big .pcap files
- Implement caching of repeated geolocation queries.
- Ease user interface

8. Deploy and Document

- Write down a User manual or guide to run the tool.
- Deploy the tool on relevant platforms, including Linux/Windows/macOS.
- Collect feedback for future improvements.

Tools used

- Wireshark(tshark)
- Google My Maps
- Visual Studio

Technologies used

- Django - 5.1.3
- gunicorn - 23.0.0
- appdirs - 1.4.4
- asgiref - 3.8.1
- lxm - 5.3.0
- packaging - 24.2
- pyshark - 0.6
- simplekml - 1.3.6
- sqlparse - 0.5.2
- termcolor - 2.5.0

MODULE 2: PASSWORD GENERATOR TOOL:-

Introduction:-

Password generator is a tool to produce random passwords with high security. Generally, random passwords have various benefits over user-chosen passwords where it enhances security and confidentiality. The new methodology has been created to generate random passwords which enable users to choose password characteristics such as length, inclusion of uppercase letters, lowercase letters, numbers, and special symbols.

Key Features:

- **Customizable Password Options** Allows users to input specify password length and character types (uppercase letters, lowercase letters, numbers, special symbols). It can assure flexibility toward different security policies, either corporate or personal.
- **Random and Secure Passwords** Generates highly randomized passwords using the *"random"* library and salting to ensure no predictable pattern. Its main advantages are enhancing resistance to brute-force and dictionary attacks. It uses *random.sample* and shuffling to ensure randomness across password characters.
- **Secure Online Processing** All password generation processes are online and secure, that is, no passwords are stored and transmitted to third parties. It protects the user data and maintains trust in its integrity.
- **Fast and Lightweight** Optimized for minimal resource usage to generate passwords quickly on systems of all specifications. It can run efficiently on older devices or limited-resource environments. This uses Python's lightweight libraries and avoids any unnecessary computations.
- **Scalable and Reliable Backend** The online service can support concurrent user requests with regard to performance. Ensures a smooth experience for users, even during peak usage. Scalable server infrastructure, load balancing, and optimized algorithms for instant response.

Requirement Specification for Password Generator Tool:-

Introduction:

The Password Generator Tool is a Python-based application designed to generate strong, secure, and customized passwords. It ensures user-friendly functionality while following best practices in password generation, randomness, and security.

Project Background:

The more frequent occurrence of cyberattacks underscores the need for strong, secure passwords in the protection of digital assets. Most available tools are expensive or dependent on internet connectivity or not open-source. Therefore, this project aims to build an open-source Python-based password generator tool that focuses on user customization and security. In this way, it bridges the gaps related to usability and randomness and empowers users to generate solid credentials pretty easily.

Purpose:

- **Enhance Cyber Security:** A trusted way to create very strong, virtually indecipherable, randomized passwords which prevent brute force or dictionary attacks.
- **Empower Users:** Option for password length and strength so that users can generate based on specified security requirements.
- **Ensure Accessibility:** A lightweight, offline/cloud based application with minimum functional extra features for privacy and usability across any platform by any user.

Functional Requirements:

1) **Password Generation:**

- Generate passwords for any length set by the user (8-32 characters).
- Select character types:
    - It can include uppercase letters from [A-Z]
    - Can include lowercase letters from [a-z]
    - Can include numbers from [0-9]
    - Can include special symbols like !, @, # and many more.

2) **Customization:**

- To choose which character types to exclude or include
- To create more than one password in a session

3) **Password Strength Validation:**

- Evaluate password strength.
- Include suggestions on how to make a weak password strong

4) **Secure Randomization:**

- The tool should apply strong randomization techniques for password generation so that it is unpredictable and conforms to modern security criteria.

Non-Functional Requirements:

1) **Usability:**
    - Simple, guiding interface (GUI).
    - Provide error handling in case of invalid input such as "password length does not satisfy the minimum requirement".

2) **Security:**
    - Generated passwords are not stored anywhere

3) **Performance:**
    - Instantly generates passwords to provide for a fast user interface.
    - Extremely resource-frugal to support all sorts of devices.

4) **Compatibility:**
    - Cross-platform support for Windows, macOS, Linux, etc.
    - Requires Python3 and standard libraries.

System Requirements:

- **Hardware:**
  - Minimum 4 GB RAM,
  - 10 GB Available storage

- **Software Requirements**:
  - Operating System: Linux, Windows and MacOS
  - Python 3 with Libraries: *random, string, tkinter*

PROBLEM DESIGN:

METHODOLOGY / EXPERIMENTAL SETUP

1. Design Phase

System Architecture:

- Modular design by the following component:
  - Input Module: Accepts user preferences (length, character types).
  - Password Generation Module: This uses the random library of Python to generate random passwords using inputs.
  - Output Module: Displays generated passwords to the user.

Algorithm Design:

- **Password Generation Algorithm:**
  1. Collect user input on length and preferences for character usage.
  2. Generate the pool of characters to use based on the options chosen: uppercase, lowercase, numbers, and/or symbols.
  3. Select characters at random from the pool until the desired length is achieved.
  4. Provide the final password.

2. Implementation

- **Importing Required Libraries:**

The code begins with importing the necessary libraries:

To build this project we will use the basic concept of python and libraries – tkinter, secrets, messagebox, os, strings.

- **secrets**: It provides access to secure random numbers suitable for generating passwords.
- **string:** It includes a collection of string constants containing ASCII characters used for generating passwords.
- **tkinter:** It is the standard Python interface to the Tk GUI toolkit.
- **messagebox:** It provides a simple way to display message boxes in Tkinter.
- **os:** It provides a way to use operating system-dependent functionality (in this case, creating directories and files).

3. Development Steps

The development process of Password Generator Tool is divided into clear and stepwise steps that ensure it to be functional, efficient, and user-friendly. Here's the step-by-step break down for each part:

1. Set Up Environment

- Objective: Prepare a system for development.
- Actions: Install Python3 and validate installation using python --version
- Provide a text editor or Integrated Development Environment (IDE) like VS Code or PyCharm to do coding.
- Install any dependencies necessary, random, string and optionally tkinter for GUI support.

2. Code Modules

- Objective: To split the project into modular functions for scalability and readability.
- Actions:
    1. Input Module: Get the user preferences, such as password length and character types (upper case, lower case, numbers, symbols). Validate input for correct values.
    2. Password Generation Module: Generate passwords using the random library.
        - Combine character sets chosen by users.
        - Select characters at random until a total length is reached.
        - Shuffle characters for randomization.
    3. Output Module: Display the generated password to the user in a readable format.

3. Integration

- Objective: Combine individual modules into an integrated tool
- Actions:
    1. Associate the input module with the preferences to be passed to the password generation module.
    2. Associate the output module that will display the generated passwords.
    3. If implementing the GUI, design forms to collect inputs and present results using *tkinter*.

4. Testing

- Objective: Test that the utility behaves as expected in a variety of scenarios.
- Actions:
    1. Test against various length and character combinations for passwords.
    2. Test error messages when the input is invalid, e.g. not long enough.
    3. Run utility several times and pass the passwords it generates against simple predictability checks.

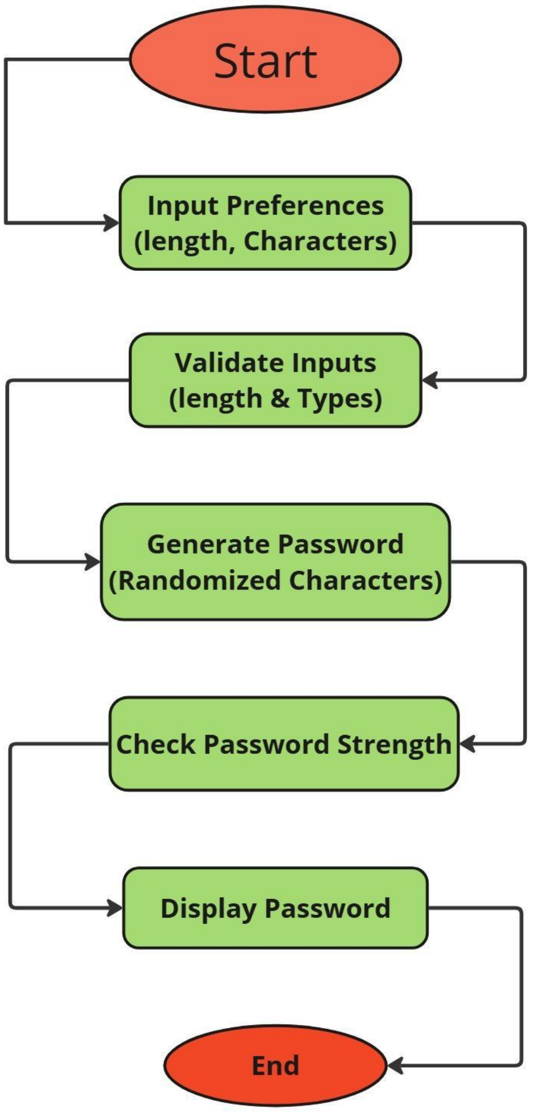
5. Packaging and Deployment

- Objective: Deliver the utility to end-users.
- Actions:
    1. Package code into a stand-alone Python script or executable using tools like *pyinstaller*.
    2. Test the tool on different operating systems such as windows, macOS, Linux to ensure compatibility.
    3. Offer user documentation of installation and usage

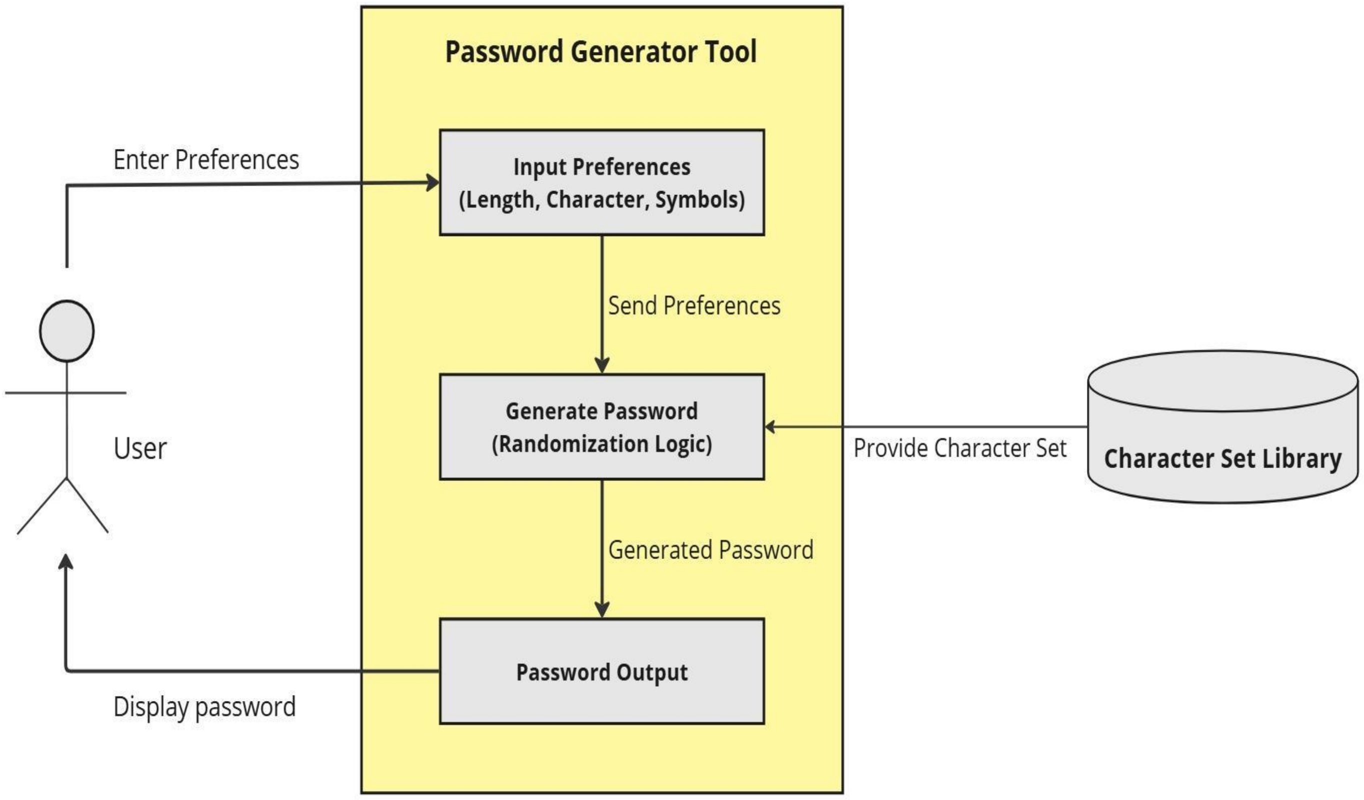
6. Maintenance and Updates

- Objective: Long-term usability and relevance.
- Actions:
    1. Collect user feedback for improvements
    2. Constantly update the code in terms of bugs or new features.
    3. Publish updates on GitHub, or more platforms for further access by the users.

Diagrams: Flow Chart


Start
Input Preferences (length, Characters)
Validate Inputs (length & Types)
Generate Password (Randomized Characters)
Check Password Strength
Display Password
End

Diagrams: Data Flow diagram



MODULE 3: NMAP SCANNER TOOL: -

Introduction: -

The Nmap Scanner Tool is part of the NetDefender Toolkit, which helps in simplifying and automating tasks for network scanning. Built upon the powerful Nmap utility that has Network Mapper, the tool offers a nice-to-have interface from which users can carry out several diagnostics and network security checks. It is user-friendly, and it caters to all levels of user experience because it is incorporated with pre-configured scanning options as well as an interactive interface for advanced commands.

This utility supports three basic modes of operation:

1. **Basic Scan (Ping Sweep):** Scans the network by sending ICMP echo requests to identify live hosts.
2. **Intermediate Scan (Top Ports):** Scans common ports in search of open or vulnerable services.
3. **Custom Command Execution:** The user could input his own Nmap commands for highly customized use cases.

The tool comes with the functionalities of displaying real-time scan output and downloading results for additional analysis. Thus, the tool can streamline network security practices. Integrating these functionalities into an intuitive graphical interface allows the Nmap Scanner Tool to bridge advanced network analysis with user accessibility. This makes it a crucial asset for both the beginner and the practitioner alike in security analysis.

NMAP (Network Mapper) :-

Nmap stands for Network Mapper and is an open-source tool that is used to discover networks and audit security conditions. It was developed by Gordon Lyon as one of the most powerful tools known in the information security domain, capable of scanning and analysing network vulnerabilities, open ports, and active hosts.

Key Features: -

1. Host Discovery: It detects active devices in a network by sending packets and analyzing the results.
2. Port Scanning: Detects open, closed, or filtered ports on a device, helping assess services and potential vulnerabilities.
3. Service and Version Detection: Identify running services on open ports, including determining their version for full analysis.
4. Operating System Detection: Uses packet signatures to identify the host's operating system and version.
5. Scriptable Scanning: An Nmap Scripting Engine can automate such tasks as vulnerability scanning, malware detection, or brute force attacks.

Working Principle: -

Nmap sends specially crafted packets to a target device or network and analyzes responses. It can do the following depending on the type of scan:

- Use ICMP packets for ping sweeps in order to detect active hosts.
- Send TCP SYN packets to check for open ports.
- Perform UDP scans to find services running over UDP.

It uses several scanning methods including TCP Connect, SYN Scan, FIN Scan, and Idle Scan, which depends on network configuration as well as user's needs.

Use Cases:-

- Network Inventory: Map all devices and their details in a network.
- Security Auditing: It identifies vulnerabilities, misconfigured services, and weak passwords.
- Compliance Testing: Ensure systems meet security standards.
- Troubleshooting: Diagnose network issues and check connectivity.

Requirement Specification for Nmap Scanner Tool:-

Introduction:

The Nmap Scanner Tool is designed as part of the NetDefender Toolkit to facilitate and ease the scanning of networks with a user-friendly interface for network security analysis. The following document details the functional and non-functional requirements required in order to put the tool into practice.

Project Background:

The Nmap Scanner Tool developed for the NetDefender Toolkit automates multiple tasks related to network security. Based on the complexity introduced by command-line tools like Nmap, this project attempts to produce a user-friendly interface in which scanning activities together with network analysis can be enhanced. It fills the gap of efficient host discovery, port scanning, and real-time viewing of results. The tool is designed to offer multiple scanning options in terms of customizable parameters to facilitate flexibility for either beginners or advanced users.

Purpose:

- **Simplify Network Scanning:** Nmap allows end-users who are not experts to access the powerful features of Nmap using an intuitive interface.
- **Increased Security Scanning:** Opens user's port, running services, and possible weaknesses on their networks to effective identification.
- **Real time Feedback:** Display scanning output in real time, thus giving the user the ability to monitor and analyze network data.
- **Customizability and Flexibility:** Permit advanced users to run custom Nmap commands for tailored network analysis.

Functional Requirements

1) **Scan Options:**
    - Basic Scan (Ping Sweep): Detect live hosts in a network using ICMP echo requests.

- Intermediate Scan (Top Ports): Scan the most commonly used ports on detected hosts.
- Custom Command Execution: Allow users to input and execute custom Nmap commands for advanced scanning.

2) **Live Output Display:**
   - Show real-time scan results in a terminal-style interface for user feedback during scanning.
3) **Result Download:**
   - Provide an option to download scan results in a *.txt* format for offline analysis and reporting.
4) **Error Handling:**
   - Display appropriate messages if scans fail due to invalid inputs, network issues, or lack of permissions.
5) **Multi-platform Compatibility:**
   - Ensure the tool runs on major operating systems like Linux, Windows, and macOS.

Non-Functional Requirements

1) **Usability:** The interface should be user-friendly with self-explanatory options for both the novice and experienced users.
2) **Performance:** Scanning operations must be efficient, with adequate real-time feedback without noticeable lags
3) **Scalability:** The tool should adequately handle both small and large environments and support multiple hosts as well as ports.
4) **Security:** The tool should prevent unauthorized access to scanning functionalities. The scans must be performed responsibly
5) **Extensibility:** This design shall also offer room for future features, such as scanning to identify vulnerabilities or graphical result representation.

System Requirements:

- **Hardware**
    - Processor: 2.0 GHz or higher
    - RAM: Minimum 4 GB
    - Storage: At least 200 MB free space
    - Network: Stable internet or local network connectivity
- **Software**
    - Frontend: HTML, CSS, JavaScript
    - Backend: Python (Django)

PROBLEM DESIGN: -

METHODOLOGY / EXPERIMENTAL SETUP

The Nmap Scanner Tool is developed with a structured methodology in order to have effective implementation of its features and functionalities. It has several multiple phases which include:

1. **Requirement Analysis:**
   The project first began with the analysis of needs for network security practitioners as well as beginners. Major focus areas included:
    - Easy to use Nmap's powerful command-line tool
    - Generate and store output and results in real time
    - Support several scanning options ranging from simple to sophisticated
    - Outcome: Functional as well as non-functional requirements for the tool were very much clear.

2. **System Design:**
   Design with a modular approach to ensure scalability and maintainability. The tool was divided into the following components:

- Frontend:
    - Developed with HTML, CSS, and JavaScript for a minimalistic and intuitive user interface.
    - It also includes the options for scan (Basic, Intermediate, Custom) and the live terminal output.

- Backend:
    - Development with Python Django manages the scan requests and processes Nmap commands.
    - It communicates with the system-installed Nmap application through subprocess handling.

3. **Development:**

It was implemented in steps:

- Frontend Implementation:
    - A responsive design was developed for cross-device compatibility.
    - WebSocket or HTTP Streaming technologies were integrated to give a live output simulating the terminal.
- Backend Development:
    - Scan requests were processed by running Nmap commands and returning its findings to the frontend.
    - Basic and intermediate scan commands were encoded in advance, while a custom option allowed users to input their own commands;
- Output Storage:
    - The addition of a feature to save scan results in .txt format for analysis off-line.

4. **Network Environment**

   To test the tool's performance, various network topologies had to be considered:

   - **Local Network:**
     - Connected several machines to one router to approximate a normal home or office network.
     - IP addresses in the format: 192.168.x.x.

   - **Public Network Scenarios:**
     - Scanned external IP addresses for external websites after obtaining permission to test the custom Nmap command functionality.

5. **Testing and Debugging:**

   Testing was done in two stages:

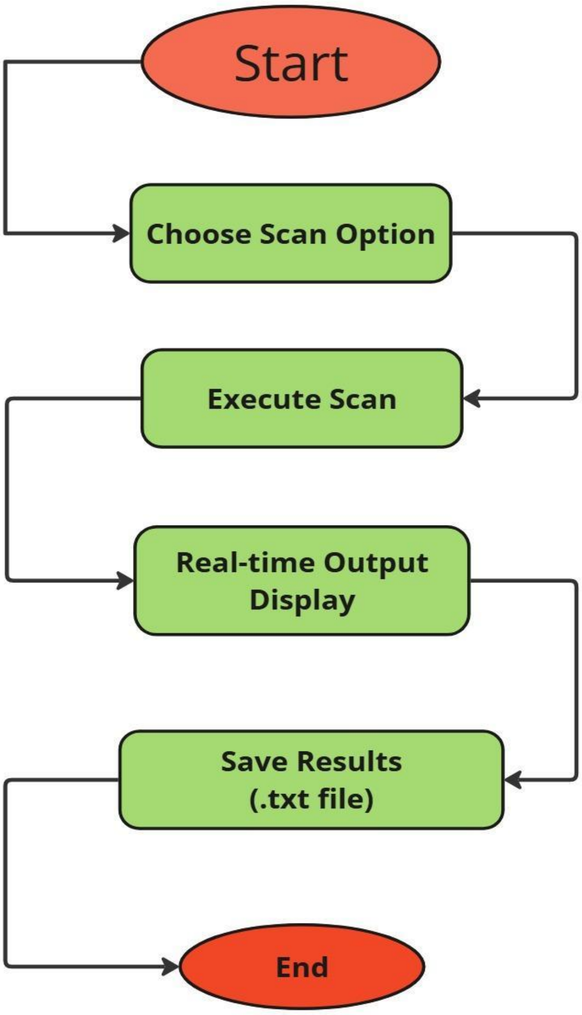
   - Unit Testing:
     - Individual parts (scan commands, output streaming, UI responsiveness) were checked for functionality.
   - Integration Testing:

     All end-to-end tests ensure smooth communication between the frontend, backend, and deployment tools.

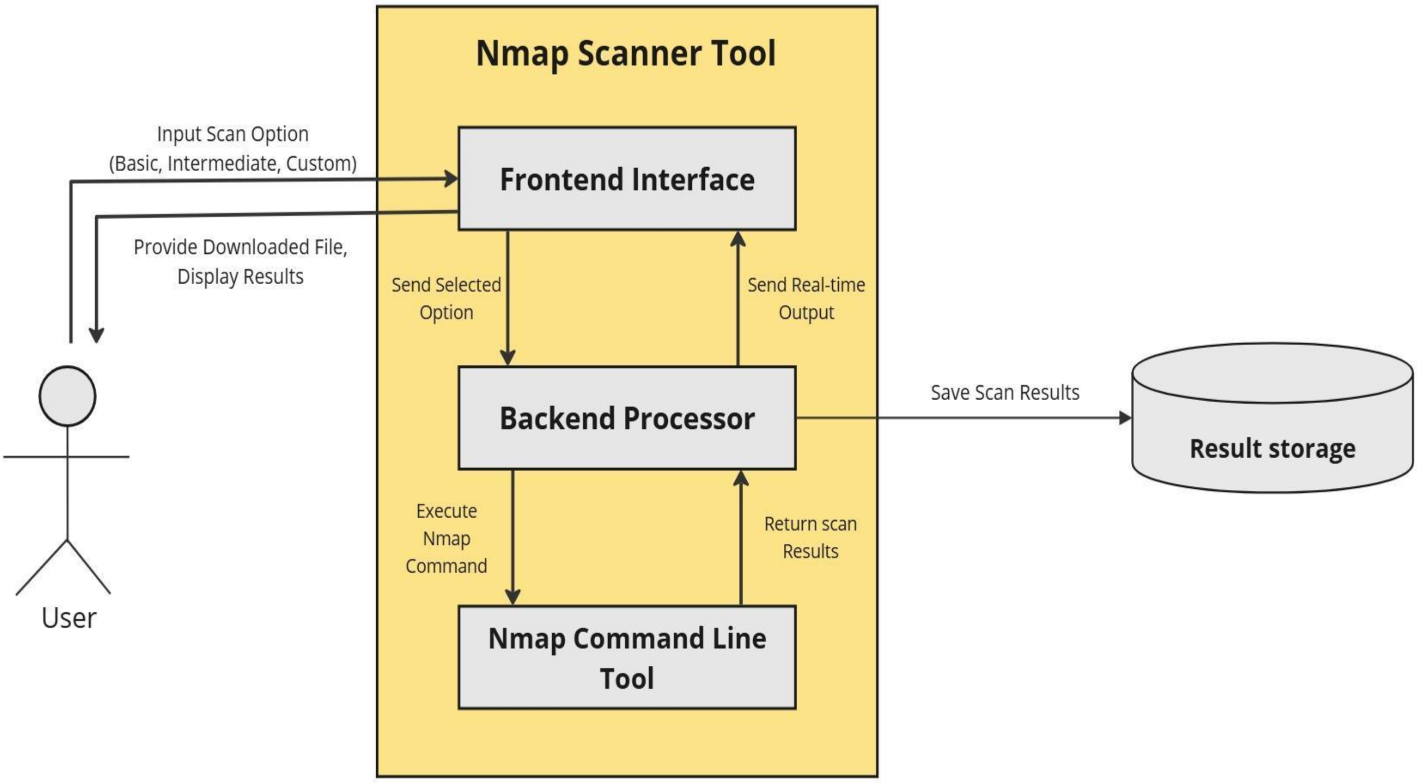
6. **Deployment and Evaluation**

   The tool was deployed on multiple platforms (Windows, Linux) to verify its cross-platform compatibility. Performance metrics such as scan speed, resource utilization, and user experience were evaluated.

Diagrams: Flow Chart



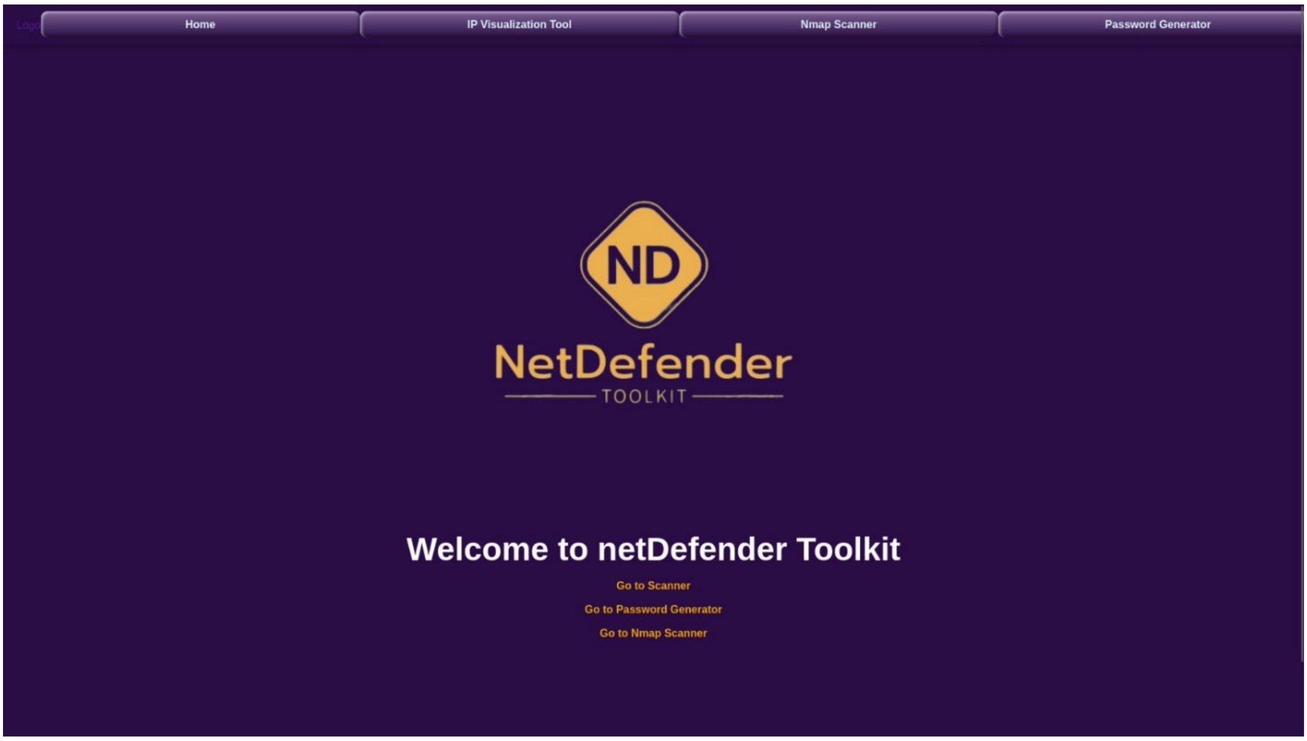
Diagrams: Data Flow diagram



RESULTS:-

Screenshots

1. Home Page



Welcome to netDefender Toolkit

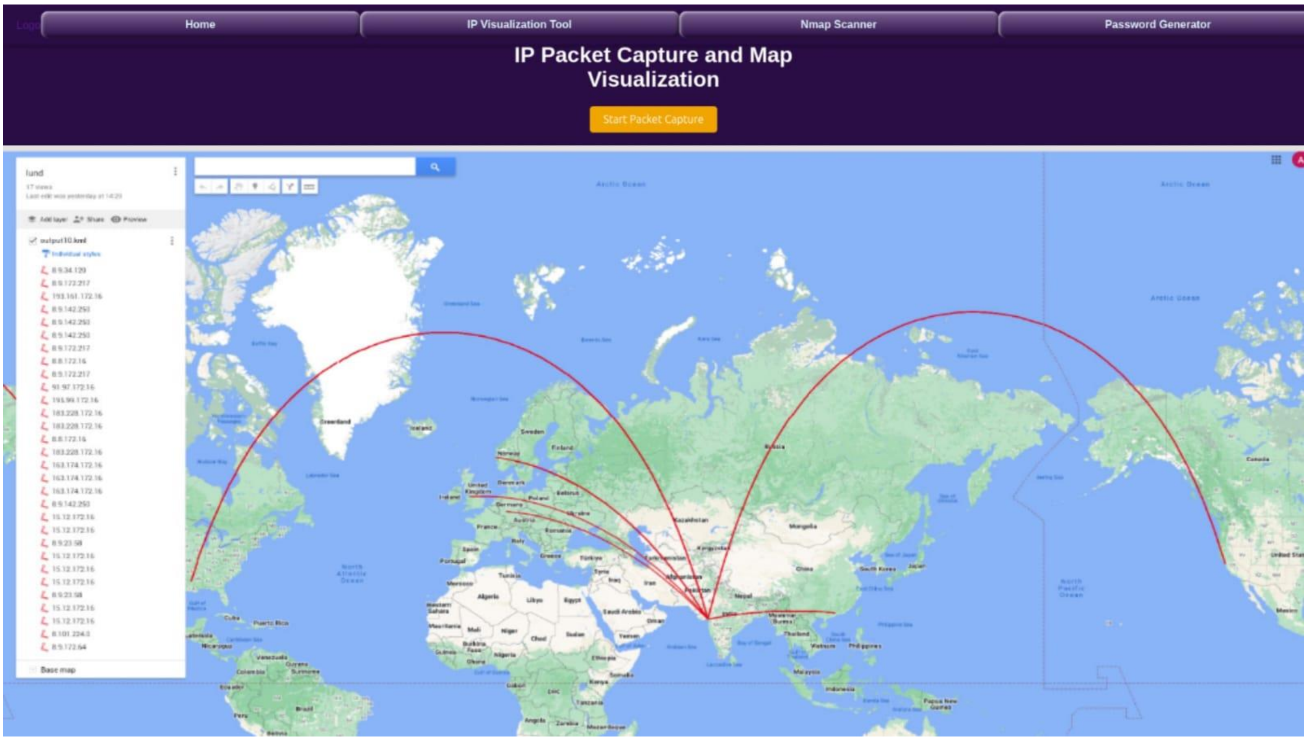
Go to Scanner

Go to Password Generator

Go to Nmap Scanner

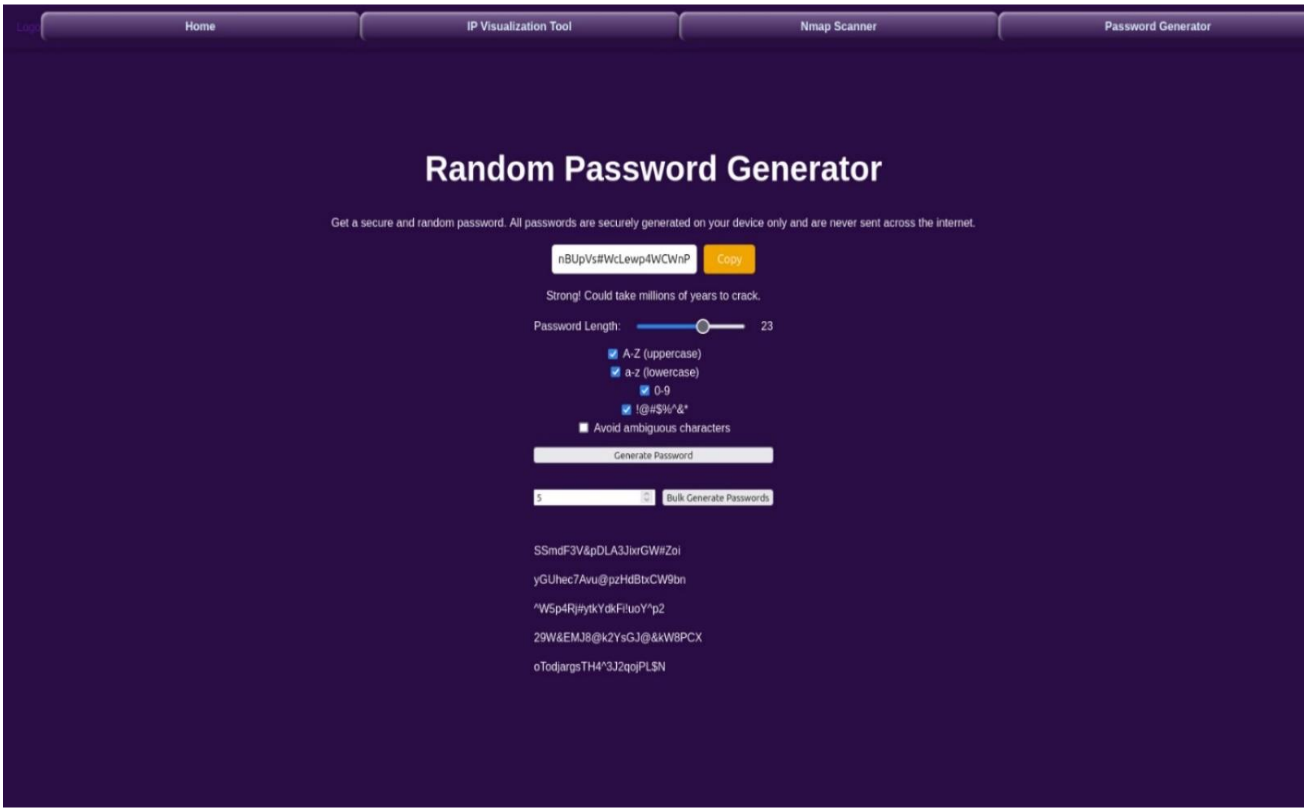
The NetDefender Toolkit home page is built for simple and effective functionality; it's clean, modern, with a central logo and welcoming text. The three top navigation modules: IP Visualization Tool, Nmap Scanner, and Password Generator will give the user quick access to such modules, making it easy to navigate. The central call-to-action links directly guide the users to explore the features of each tool, thereby emphasizing usability and accessibility. The tool kit's focus is on cyber security, which is reflected in the dark-themed color palette and a professional layout of the homepage.

2. IP Visualisation Tool Tab



The IP Visualization Tool basically provides a dynamic interface for the monitoring of network traffic mapping source and destination IP addresses onto a world map; thus, this real-time packet data capture is apparently displayed with visual evidence, making it easier to identify traffic patterns and locate threats geographically. Using tools such as Wireshark for collecting data and Google My Maps for visualization, it provides an intuitive way of understanding and analysing network activity. This tool is very effective for network admins to identify anomalies and increase the efficiency of threat response.

3. Password Generator Tool



Random Password Generator

Get a secure and random password. All passwords are securely generated on your device only and are never sent across the internet.

nBUpVs#WcLewp4WCWnP Copy

Strong! Could take millions of years to crack.

Password Length: 23

☑ A-Z (uppercase)
☑ a-z (lowercase)
☑ 0-9
☑ !@#$%^&*
□ Avoid ambiguous characters

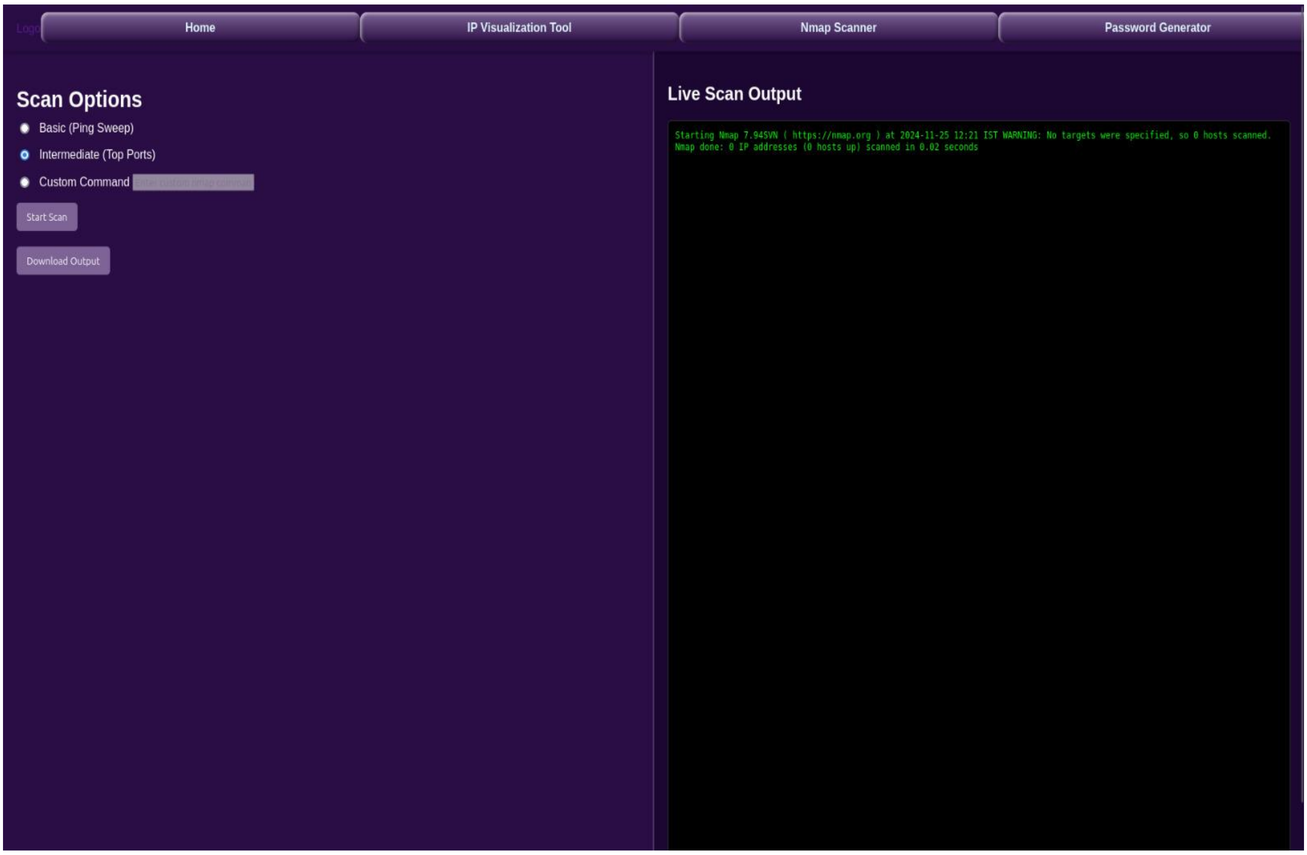
Generate Password

5 Bulk Generate Passwords

```
SSmdF3V&pDLA3JixrGW#Zoi
yGUhec7Avu@pzHdBtxCW9bn
^W5p4Rj#ytkYdkFi!uoY^p2
29W&EMJ8@k2YsGJ@&kW8PCX
oTodjargsTH4^3J2qojPL$N
```

The interface shown is a Random Password Generator tool that allows users to create secure passwords with customizable options. Users can adjust password length, include specific character types (uppercase, lowercase, numbers, symbols), and avoid ambiguous characters. The generated password displays its strength, and users can bulk-generate multiple passwords at once. A "Copy" button simplifies the process of using the generated password.

4. Nmap Scanner Tool



Scan Options

○ Basic (Ping Sweep)

● Intermediate (Top Ports)

○ Custom Command Enter custom Nmap command

Live Scan Output

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-25 12:21 IST WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.02 seconds
```

The interface shown is an Nmap Scanner tool with customizable scan options. Users can choose between Basic (Ping Sweep), Intermediate (Top Ports), or Custom Command scans. A live terminal on the right displays real-time output of the scan progress. The tool includes a "Download Output" option for saving scan results locally.

CONCLUSION & SCOPE OF FUTURE WORK: -

Conclusion:

The NetDefender Toolkit is one comprehensive solution of the critical challenges that modern cloud-based cybersecurity faces. In combining three different modules tailored to be complementary-the IP Address Visualization Tool, Password Generator Tool, and Nmap Scanner Tool-this toolkit allows users to monitor, secure, and assess their networks more effectively.

This tool has assisted in network visibility by mapping IP addresses to their origins geographically, and users are able to define traffic patterns and potential threats. With real-time packet capture capabilities and intuitive geospatial representation, this is a valuable asset for network administrators in risk understanding and mitigation.

The tool Password Generator strengthens security at its foundation-password protection. It generates highly secure and customizable passwords, which it helps prevent unauthorized access to, protects against brute-force and dictionary attacks, and is light on usability without compromising its security-related features.

The Nmap Scanner Tool is a robust platform for network analysis and vulnerability assessment. It supports predefined and tailor-made scans and allows users to discover open ports, running services, and security flaws. Real-time output along with the saving of results adds further utility for both novices and advanced users.

These modules combined create one uniform, modular platform in the address of wide-ranging cloud-based cybersecurity needs. The NetDefender Toolkit is the gap between advanced functionalities and user-friendly design that opens it up to both individuals and organizations. It has a strong scalability and cross-platform compatibility, not to mention its focus on security, for it is the most valuable asset in the fight against evolving cyber threats. By equipping users with tools for monitoring, protection, and analysis, the NetDefender Toolkit helps build more resilient and secure digital environments.

Scope of Future Work:

Scope:

The Toolkit NetDefender offers a flexible and scalable architecture that responds to a wide range of cloud-based cybersecurity challenges. Its scalable architecture provides adaptability to different use cases, such as enhancing personal security and setting up a network level of protection in enterprises. Incorporating the IP Address Visualization Tool, Password Generator Tool, and Nmap Scanner Tool, the toolkit provides a comprehensive framework for monitoring, securing, and analysing digital systems.

The toolkit currently implements the functionality to deliver strong capabilities in tackling major pillars of cloud-based cybersecurity aspects:

- **IP Address Visualization Tool:** Provides real-time geolocation and visualization of network traffic, which can identify and respond to potential security threats from users.
- **Password Generator Tool:** Generates strong passwords according to predefined user criteria, which would meet the foundational layer of security of a system.
- **Nmap Scanner Tool:** Enables network scanning and vulnerability assessment by catering to both beginners and advanced users with customizable options.

These tools are cross-platform, user-friendly, and scalable; therefore, the toolkit is suitable for individual and organizational contexts to improve cloud-based cybersecurity posture.

Future Work:

The following improvements should be designed to expand its utility and sustainability in terms of changing threats:

1. **Enhanced Visualisation and Analytics:**
    - Implement advance Data visualization techniques that include heatmaps and traffic trend analysis as an enhancement for the IP Address Visualization Tool.
    - Machine learning approaches to detect and predict anomalous traffic patterns.

2. **Advanced Features for Password Security:**

    - Use a password strength evaluator to strength-check any existing password
    - Implementation of a secure storage solution to store and retrieve generated passwords.

3. **Advanced Scanning and Reporting in the Nmap Tool:**

    - Facilitates graphical output of scan reports.
    - Automated vulnerability reporting that contains actionable remediation steps.

4. **Centralized Dashboard:**

    - Building a single interface for managing all modules, with smooth navigation and real-time updates across tools.

NetDefender Toolkit is on its way to becoming a new-age cloud based cybersecurity toolkit, making it relevant and effective for years to come in the fight against newer threats while equipping the user with powerful tools that are yet intuitive enough to use.

REFERENCES

1. https://indusedu.org/pdfs/IJREISS/IJREISS_2136_66085.pdf
2. https://nmap.org/book/nmap-overview-and-demos.html
3. https://www.wireshark.org/docs/wsug_html_chunked/
4. https://www.academia.edu/105724614/A_technical_investigation_into_port_scanning_using_Nmap
5. https://en.wikipedia.org/wiki/Nmap
6. https://en.wikipedia.org/wiki/Random_password_generator
7. https://threatmap.checkpoint.com/