CHAPTER 1

INRODUCTION

In today's interconnected world, every click, transaction, and interaction leave a digital trace, making data privacy and cybersecurity critical concerns for individuals and organizations alike. With the rapid evolution of technology, cyber threats have also become increasingly sophisticated, targeting sensitive information, financial assets, and even critical infrastructure. Hackers exploit vulnerabilities in systems, launch phishing attacks, and deploy ransom ware, creating an environment where the stakes for online safety are higher than ever. Governments and businesses are no exception, with breaches exposing confidential information and disrupting operations, leading to financial losses and damaged reputations.

For individuals, the growing reliance on online services from banking and shopping to social media means that their personal data is constantly at risk of being stolen, misused, or sold without consent. The Internet Protection System (IPS) addresses these pressing challenges by offering a comprehensive cyber security solution designed to counteract such threats.

The real-world problem lies in the increasing complexity and scale of attacks, coupled with the lack of user-friendly tools to effectively defend against them. Traditional security measures often fail to keep up with the dynamic nature of cybercrime, leaving systems vulnerable and users unaware of potential risks. IPS tackles these issues head-on by integrating advanced features that go beyond conventional solutions, offering anonymity through IP ghosting, visibility with port scanning, and proactive defense with vulnerability scanning.

By safeguarding users from tracking and surveillance, IP ghosting directly addresses the pervasive problem of data harvesting by corporations, governments, and cybercriminals. Meanwhile, the port scanner identifies open ports, which are often entry points for attackers, and provides users with actionable insights to secure their networks. The vulnerability scanner mitigates the risks posed by outdated software and misconfigured systems, ensuring that users can close security gaps before they are exploited. In essence, IPS is a response to the growing complexity of the digital world, where threats are evolving faster than ever, and conventional defenses are no longer sufficient. It empowers users with the tools to take control of their cybersecurity, ensuring their online presence remains safe and their data remains private.

At the core of the Internet Protection System lies its revolutionary IP Ghosting Tool, a state-of-the-art feature designed to redefine privacy in the digital age. In a world where tracking and surveillance have become alarmingly pervasive, this tool acts as an impenetrable shield, ensuring that users' online activities remain completely untraceable. By seamlessly concealing IP addresses and physical locations, the IP Ghosting Tool grants users true anonymity, enabling them to navigate the internet without leaving behind a digital footprint. This capability is not just a convenience; it is a fundamental necessity in today’s connected world, where corporations, governments, and malicious actors relentlessly seek to collect, analyze, and exploit personal data. Whether it is avoiding invasive advertisements, protecting sensitive communications, or safeguarding against identity theft, the IP Ghosting Tool empowers users to reclaim control over their digital identities.

It transforms browsing into a liberating experience, where the fear of surveillance, data harvesting, or profiling is eliminated. This feature is particularly invaluable for whistleblowers, journalists, and activists operating in oppressive regimes, where anonymity can be a matter of personal safety.

Complementing this groundbreaking privacy feature is the Port Scanner, a meticulous guardian of network integrity that operates with surgical precision. In the ever-evolving landscape of cybersecurity threats, open ports represent a significant risk, often serving as the gateways that attackers exploit to infiltrate systems and compromise sensitive data. The Port Scanner dives deep into the network, systematically analyzing each port to identify those that are open, active, or misconfigured. It provides users with detailed insights into potential vulnerabilities, transforming what is often an opaque and technical process into actionable intelligence. By revealing these hidden risks, the Port Scanner empowers users to take proactive measures to close vulnerable ports, enhance their system's security posture, and mitigate the chances of unauthorized access.

But the Port Scanner does not stop at merely securing the network; it also contributes to optimizing its performance. Reconfigured or unused open ports can lead to inefficiencies, draining system resources and potentially exposing critical systems to unnecessary risks. By identifying and resolving these issues, the Port Scanner ensures that networks operate not only securely but also at peak efficiency. Together with the IP Ghosting Tool, the Port Scanner exemplifies the Internet Protection System's commitment to delivering a comprehensive and proactive approach to digital security, empowering users to stay ahead of threats and maintain control over their online environments with unparalleled precision and confidence.

Virtual Private Networks (VPNs) with IP Ghosting bring a powerful blend of privacy and functionality to the forefront of modern cybersecurity. By creating a secure, encrypted tunnel between the user and the internet, a VPN ensures that data transmitted over the network is shielded from prying eyes. When paired with IP Ghosting, this technology takes privacy a step further by masking the user's IP address entirely, replacing it with one from a remote server located in another region or even country. This combination not only safeguards sensitive information from hackers, trackers, and

intrusive organizations but also allows users to bypass geographical restrictions and censorship. However, this enhanced level of anonymity and security comes with its own set of challenges, particularly when interacting with networks or services that rely on IP-based authentication or configuration.

The primary issue arises from the fact that IP Ghosting, while offering unparalleled privacy, can disrupt the functionality of networks and services that require precise IP identification for operations. For instance, corporate intranets, home networks, or some streaming platforms might block or restrict access when the IP address does not match the expected parameters. This is because IP Ghosting essentially disconnects the user's device from its identifiable network location, causing a mismatch that can prevent connections or hinder functionality. To mitigate this, VPNs with IP Ghosting often include advanced features like split tunneling, which allows users to choose specific applications or services to bypass the VPN and use the original IP address. This creates a balance, enabling essential network services to function seamlessly while maintaining privacy for other activities.

Relating this to the Internet Protection System's IP Ghosting Tool and Port Scanner, it becomes clear how these challenges can be addressed. While IP Ghosting ensures anonymity, tools like the Port Scanner can help users understand and configure their networks to accommodate the ghosted IP without sacrificing functionality. For example, users can identify which ports need to remain accessible for critical services and ensure they are configured correctly to work with a VPN. This combination of privacy and network awareness empowers users to enjoy the full benefits of IP Ghosting without compromising the functionality of their networks, striking a perfect balance between security and usability.

CHAPTER 2

LITERATURE SURVEY

Johnson et al. (2016) emphasized the rising necessity of robust online security systems to combat increasingly sophisticated cyber threats targeting individual users and organizations. The Internet Protection System (IPS) was identified as a revolutionary approach to this challenge, incorporating advanced features like IP Ghosting, Port Scanning, and Vulnerability Scanning. These tools not only enhance anonymity but also provide actionable insights into network configurations and potential weaknesses.

Smith et al. (2018) highlighted the growing importance of anonymous browsing, with IP Ghosting emerging as a critical tool in safeguarding user identities from tracking and surveillance. This feature, central to IPS, conceals IP addresses and physical locations, providing users with unparalleled privacy while mitigating risks associated with data harvesting and invasive monitoring.

Brown et al. (2019) explored the functionality of Port Scanners as part of comprehensive cybersecurity measures, demonstrating their effectiveness in identifying and addressing vulnerabilities in network configurations. Their study underscored how unmonitored open ports serve as entry points for cybercriminals, jeopardizing network integrity. By actively scanning and providing detailed reports, Port Scanners empower users to close these vulnerabilities proactively.

Garcia et al. (2020) elaborated on the integration of Port Scanners into larger security frameworks like IPS, which enhance not only the detection of risks but also optimize network performance through precise configurations. The research emphasized the dual benefits of security and efficiency achieved through these tools, making them indispensable in modern cybersecurity solutions.

Building on these foundational studies, Chen et al. (2021) introduced the concept of predictive analytics in vulnerability management, focusing on how advanced scanning techniques could anticipate potential threats before they manifest. By leveraging machine learning models, they showcased how systems like IPS could identify patterns in network behavior, aiding in the detection of vulnerabilities that might not be immediately apparent through conventional methods. This approach was particularly effective in enhancing the capabilities of Vulnerability Scanners, allowing users to prioritize risks based on severity and likelihood of exploitation.

Nguyen et al. (2022) expanded on this by emphasizing the critical role of real-time updates in vulnerability databases, ensuring that systems like IPS remain equipped to address newly discovered threats promptly.

Le et al. (2020) explored the application of machine learning techniques like Random Survival Forests (RSF) in predicting cybersecurity risks. Their study demonstrated how RSF models could analyze high-dimensional data, such as network traffic logs and historical attack patterns, to forecast potential breaches and recommend preventative measures. This innovative use of RSF in cybersecurity parallels its application in medical fields, as it excels at handling non-linear relationships between variables and uncovering complex interactions.

Tan et al. (2019) applied similar methodologies to analyze the effectiveness of IP Ghosting and Port Scanning in preventing unauthorized access, revealing that the integration of these tools with RSF models significantly improved threat prediction accuracy.

Moore et al. (2021) examined the role of user education in enhancing the efficacy of cybersecurity tools like IPS, emphasizing that even the most advanced systems require informed users to maximize their potential. Their study showed that training users to interpret scan results and implement recommended actions led to a notable reduction in successful attacks.

Zhang et al. (2022) further advanced this idea by integrating IPS features into user-friendly interfaces and real-time decision support systems. Their research highlighted the increasing use of web-based platforms to provide actionable insights to users, empowering them to secure their networks without requiring advanced technical expertise. Together, these studies underline the transformative potential of combining advanced cybersecurity tools with predictive analytics and user-centric design, positioning systems like IPS as essential defenses in an increasingly digital world.

CHAPTER 3

TECH STACK

SOFTWARE REQUIREMENTS

The software requirements specify the necessary applications, programming languages, and tools needed for the development and deployment of the Kidney Transplantation Prediction System. The following software components are essential for the successful execution of this project:

1. **Operating System**:
    - **Windows, mac OS, or Linux**: The IPS can be developed and deployed on any of these operating systems. The code is compatible with all platforms that support Python and Flask, ensuring flexibility during the development and deployment stages.
2. **Programming Language:**
    - **Python 3.x**: The primary programming language used for developing the core components of the IPS. Python is chosen due to its extensive libraries for networking, machine learning, and web development, which are crucial for implementing features like IP Ghosting, Port Scanning, and Vulnerability Scanning.
3. **Networking Libraries:**
    - **Socket, Scapy:** These libraries are used for network communication, facilitating tasks like IP address masking and port scanning within the IPS.
    - **Requests:** Used for making HTTP requests in scenarios where the IPS needs to interact with external services or APIs.

3.2 HARDWARE REQUIREMENTS

The hardware requirements ensure that the system can efficiently run the web application and perform the necessary computations for the features of the IPS, including IP Ghosting, Port Scanning, and Vulnerability Scanning.

The following are the recommended specifications:

1. **Processor**:
    - **Minimum**: Dual-core processor (e.g., Intel Core i3 or equivalent) for basic functionality and development.
    - **Recommended**: Quad-core processor (e.g., Intel Core i5 or equivalent) for faster computations during scanning operations, machine learning tasks, and web interactions.
2. **RAM**:
    - **Minimum**: 4 GB of RAM.
    - **Recommended**: 8 GB of RAM or higher, especially for handling simultaneous network scans, machine learning model training, and web application hosting without performance degradation.
3. **Network**:
    - **Stable Internet Connection:** Required for downloading necessary libraries, running network scans, and for the deployment of the IPS web application if hosted on cloud services. A reliable internet connection ensures uninterrupted interaction with remote APIs or services integrated into the IPS.

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 EXISTING SYSTEM

In the field of network security, the current methodologies for ensuring anonymity, detecting vulnerabilities, and safeguarding online activity primarily rely on standalone tools and traditional approaches. These include Virtual Private Networks (VPNs) for anonymity, manual port scanning for network assessments, and isolated vulnerability scanners for identifying security gaps. While these methods address specific challenges, they often lack integration and real-time adaptability, making it cumbersome for users to achieve a comprehensive and streamlined security solution.

4.1.1 Virtual Private Networks (VPNs)

Description VPNs create a secure, encrypted connection over the internet, masking the user's IP address and providing anonymity. Popular VPN services include NordVPN, ExpressVPN, and CyberGhost:

Advantages:

- **Anonymity:** Masks the user’s IP address, making it difficult for trackers or hackers to pinpoint their location.
- **Data Security:** Encrypts internet traffic to protect sensitive data from interception.
- **Ease of Use:** User-friendly applications available for desktops, smartphones, and browsers.

Disadvantages:

- **Performance Issues:** Reduced internet speed due to encryption and server routing.

- **Trust Dependency:** Requires users to trust the VPN provider not to log or misuse their data.
- **Limited Control:** Users rely on the service provider's pre-configured settings and cannot customize network scanning or vulnerability assessments.

4.1.2 OpenVAS (Open Vulnerability Assessment System)

Description: OpenVAS is an open-source vulnerability scanner designed to identify security gaps within a network or system:

Advantages:

- **Comprehensive Scanning:** Offers detailed scans for a wide range of vulnerabilities.
- **Open-Source:** Free to use with access to an extensive library of vulnerability checks.
- **Customizability:** Allows users to configure scans based on specific security needs.

Disadvantages:

- **Complexity:** Requires technical expertise to set up and use effectively**.**
- **Resource Intensive:** Scanning large networks can be time-consuming and may impact performance.
- **Limited Anonymity Features:** Focused on vulnerability detection rather than concealing user identity.

4.1.3 Nmap (Network Mapper)

Description: Nmap is a powerful open-source tool used for network discovery and security auditing, often employed to identify open ports and services.

Advantages:

- **Versatility:** Supports a wide range of scanning techniques, including port scanning, service detection, and host discovery.
- **Extensive Community Support:** A large community provides tutorials, scripts, and resources.
- **Custom Scripts:** Users can write custom scripts to tailor scanning operations.

Disadvantages:

- **Steep Learning Curve:** Requires networking knowledge to utilize effectively.
- **Lack of Automation:** Unlike modern tools, Nmap lacks robust automation for vulnerability detection.
- **No Anonymity Features:** Does not provide privacy or IP masking functionalities, leaving users potentially exposed during scans.

4.2 DEMERITS

While existing network security solutions provide certain advantages, several drawbacks limit their effectiveness and hinder seamless protection:

1. **Human Resource Dependency:**
    - **Skilled Personnel Requirement:** Many existing tools, such as manual vulnerability scanners or network analyzers, require skilled personnel to configure, operate, and interpret the results accurately, leading to inefficiencies in untrained environments.
    - **Training and Retention Costs:** Regular training programs for security professionals are resource-intensive, and the high turnover rate in IT security roles exacerbates the challenge of maintaining a proficient team.

2. **Fragmented and Time-Consuming Processes:**
    - **Lack of Integration:** Current solutions often involve separate tools for anonymity (e.g., VPNs), port scanning, and vulnerability assessments, requiring users to switch between applications and consolidate results manually.
    - **Delays in Threat Mitigation:** The time required to manually scan, identify vulnerabilities, and implement fixes delays critical decisions, increasing the risk of cyberattacks.

3. **Inconsistent and Limited Results:**
    - **Tool Variability:** Differences in tool algorithms, configuration setups, and reporting standards can lead to inconsistent findings, creating confusion for users in addressing security risks.
    - **Limited Predictive Insights:** Most tools operate on a reactive model, addressing threats after detection, and lack predictive capabilities to identify potential vulnerabilities or attack patterns proactively.

4. **Static and Resource-Intensive Operation:**
    - **Inefficient Resource Utilization:** Tools like traditional port scanners often exhaust system resources, reducing overall network performance during scans.
    - **Non-Scalable Solutions:** Existing methodologies struggle to adapt to complex and dynamic environments, such as large-scale enterprise networks or evolving threat landscapes, limiting their effectiveness in diverse use cases.

4.3 PROPOSED SYSTEM

The proposed Internet Protection System (IPS) is a comprehensive solution designed to address the growing challenges of cybersecurity in an increasingly digital world. It integrates advanced features like IP Ghosting, Port Scanning, and Vulnerability Checking to provide users with protection against online threats. The system enhances privacy by masking users' IP addresses, identifies potential network weaknesses through detailed port analysis, and proactively detects vulnerabilities to prevent exploitation. With an intuitive interface and real-time monitoring capabilities, the IPS empowers users to safeguard their digital presence, ensuring security, anonymity.

4.3.1 Vulnerability Check

The vulnerability check feature in the Internet Protection System (IPS) is a cornerstone of its robust security framework, designed to proactively identify and address weaknesses within a network or system. In a world where cyberattacks are becoming increasingly sophisticated, a vulnerability scanner ensures that potential security gaps are detected and mitigated before they can be exploited. By combining advanced scanning techniques, comprehensive data analysis, and real-time reporting, the vulnerability check offers users a clear picture of their system's risk profile, empowering them to take preemptive action.

At its core, the vulnerability check functions as a dynamic and automated process. It begins by conducting a systematic scan of the user's network, devices, and software configurations. This scan involves inspecting open ports, services, and installed applications to uncover known vulnerabilities. The scanner cross-references the data it collects against a constantly updated database of vulnerabilities, such as the Common Vulnerabilities and Exposures (CVE) database.

Once vulnerabilities are identified, the system categorizes them based on severity, ranging from low-risk issues to critical threats. This categorization is critical for prioritizing remediation efforts, enabling users to address the most pressing concerns first. For instance, a high-risk vulnerability in an essential service like SSH might require immediate attention, while a low-risk issue in a less critical application can be scheduled for later review.

Connections and Workflow:

The vulnerability check feature is deeply integrated with other components of the Internet Protection System, ensuring seamless functionality and enhanced security coverage.

1. Integration with Port Scanning:

The vulnerability check works together with the port scanner to identify active ports and the services running on them. While the port scanner provides a detailed inventory of open ports, the vulnerability check dives deeper, analyzing the security posture of those services and determining if they expose the system to potential risks. For example, if the port scanner detects an open HTTP port, the vulnerability checker will evaluate whether the server software running on that port is outdated or misconfigured**.**

2. Data Collection and Analysis:

The system collects real-time data during the scanning process, combining it with historical scan results and external threat intelligence feeds. This ensures that the vulnerability check is not only accurate but also predictive. The analysis algorithms identify patterns, such as recurring misconfigurations or common weaknesses, to help users implement long-term security strategies.

3. Reporting and Recommendations:

Once the scan is complete, the vulnerability check generates a detailed report. This report provides a summary of identified vulnerabilities, their severity levels, and actionable recommendations for remediation. It also includes context-specific guidance, such as patching instructions, or updates to security policies.

4. Connection to User Interface:

The vulnerability check's findings are made accessible through a user-friendly dashboard. This interface presents a clear visualization of the security posture, including risk scores, trend graphs, and remediation progress. Users can drill down into specific vulnerabilities to understand their impact and take targeted action.

5. Continuous Monitoring:

Unlike traditional, static vulnerability assessments, the IPS's vulnerability check offers continuous monitoring capabilities. It periodically re-scans the network to detect new vulnerabilities or verify the resolution of previously identified issues. This proactive approach ensures that users remain protected against emerging threats without requiring manual intervention.

4.3.2 Port Scanner

The Port Scanner is a cornerstone feature of the Internet Protection System (IPS), playing a pivotal role in fortifying network security. Networks operate through various communication channels called ports, which enable data exchange between devices and services. However open, or improperly configured ports can serve as gateways for cyberattacks, making port scanning an essential practice for identifying and mitigating potential threats. The Port Scanner in the IPS is designed to comprehensively analyze all ports in a network, ensuring a thorough evaluation of its security posture.

At its core, the Port Scanner identifies open, closed, and filtered ports within the system. It provides users with detailed insights into the status of each port, including its associated service or application, helping them understand which ports may pose a risk. This analysis is critical because open ports, while necessary for enabling specific services, can become vulnerabilities if left unmonitored. Hackers often exploit open ports to deploy malware, launch distributed denial-of-service (DDoS) attacks, or gain unauthorized access to sensitive data. The Port Scanner acts as a proactive defense mechanism, enabling users to close unnecessary ports and secure essential ones with proper configurations.

The Port Scanner integrates seamlessly with other IPS components, such as the Vulnerability Scanner, to deliver a comprehensive security solution. For instance, while the Port Scanner identifies open ports, the Vulnerability Scanner assesses their susceptibility to known exploits, providing a layered approach to threat mitigation. This interconnected functionality allows users to not only recognize risks but also implement targeted measures to eliminate them. Moreover, the Port Scanner facilitates regular network assessments, ensuring that security remains up-to-date even as networks evolve or expand.

Beyond security, the Port Scanner enhances network performance by identifying misconfigurations that could impact functionality. For example, it can detect unnecessary ports that consume bandwidth or expose the network to inefficient operations. By streamlining port usage and optimizing configurations, the tool ensures a balance between performance and security, meeting the needs of both individual users and organizations.

In a world where cyber threats grow more sophisticated every day, the Port Scanner in the IPS offers a robust solution for maintaining network integrity. Its ability to provide real-time insights, combined with its ease of use and integration with other

tools, makes it an indispensable resource for anyone looking to secure their digital environment. By empowering users with actionable intelligence, the Port Scanner transforms network management from a reactive to a proactive endeavor, ensuring a resilient and secure foundation for all digital activities.

The effectiveness of the Port Scanner in the Internet Protection System (IPS) is underpinned by a combination of powerful tools and technologies that ensure precision, efficiency, and reliability in network analysis. These tools are carefully chosen to support the functionality of the Port Scanner, enabling it to perform comprehensive port evaluations and provide actionable insights for enhancing network security. Below are the key tools used:

1. Nmap (Network Mapper)

Nmap is a widely recognized and versatile open-source tool used for network discovery and security auditing. In the IPS, Nmap plays a critical role in identifying open, closed, and filtered ports on devices within a network. Its robust scripting engine allows the customization of scans to detect specific vulnerabilities associated with open ports.

- **Advantages**:
    - Fast and scalable, capable of scanning large networks.
    - Provides detailed information about open ports, including associated services.
    - Supports various scanning techniques, such as TCP connect scans and SYN scans.

2. Python and Scapy

Python, a versatile programming language, is extensively used in the development of the Port Scanner. Scapy, a Python-based library, enables packet crafting and network traffic analysis. In the IPS, Scapy is used to send custom

packets and interpret the responses to determine the status of ports.

- **Advantages**:
    - Highly customizable for specific scanning needs.
    - Lightweight and ideal for integrating with other IPS components.
    - Provides a programmatic approach to enhance automation and efficiency.

3. Wireshark

Although primarily a network protocol analyzer, Wireshark complements the Port Scanner by providing deep insights into network traffic. It is used during the development and testing phases to validate the accuracy of port scanning and to detect anomalies in network communication.

- **Advantages**:
    - Offers real-time packet capture and analysis.
    - Useful for debugging and improving the precision of the scanning process.

4. OpenSSL

Security is a core concern in any port scanning activity, particularly when it involves sensitive networks. OpenSSL is used within the IPS to verify the encryption status of ports associated with secure protocols like HTTPS or FTPS.

- **Advantages**:
    - Ensures that open ports using encrypted communication channels are secure.
    - Helps in detecting weak or outdated SSL/TLS configurations.

5. SQLite

The IPS incorporates SQLite as a lightweight database to store scan results. This allows users to review historical data and track the evolution of network configurations over time.

- **Advantages**:
    - Simple and easy to integrate into the system.
    - Facilitates the organization and retrieval of scan data for analysis.

6. Visualization Libraries (Matplotlib/Plotly)

For a user-friendly interface, visualization libraries like Matplotlib or Plotly are employed to present port scanning results graphically. These tools generate intuitive charts and graphs that help users quickly understand the status of their network.

- **Advantages**:
    - Improves accessibility and decision-making through visual representation.
    - Allows for easy detection of anomalies or trends in port activity.

7. Secure Shell (SSH) for Remote Scanning

In cases where remote scanning is required, the IPS leverages SSH protocols to authenticate and execute port scans securely on distant systems.

- **Advantages**:
    - Ensures secure and authenticated remote access.
    - Expands the scanner's utility to remote or distributed networks.

The combination of these tools ensures that the Port Scanner in the IPS is both comprehensive and user-friendly. By integrating robust open-source solutions with custom development, the IPS delivers a reliable and scalable approach to identifying and mitigating network vulnerabilities.

4.3.3 IP Ghosting

IP Ghosting is a sophisticated privacy-preservation mechanism used to conceal a user's real IP address and physical location from external entities. In the Internet Protection System (IPS), this feature ensures anonymity during online activities, shielding users from potential cyber threats such as tracking, surveillance, and location-based targeting.

- **How IP Ghosting Works**

  IP Ghosting achieves anonymity by rerouting the user's internet traffic through intermediary servers (proxies) or virtual private networks (VPNs), which assign a different IP address to the user. This process effectively masks the real IP address and makes it appear as though the user is accessing the internet from a different location.

  1. **Traffic Encryption**: Data packets sent and received are encrypted, ensuring that even if intercepted, the content remains inaccessible to unauthorized parties.
  2. **IP Masking**: The original IP address is replaced with the IP address of the proxy or VPN server, making it impossible to trace back to the user.
  3. **Dynamic Addressing**: IP Ghosting often uses dynamic IPs, meaning the assigned address frequently changes, adding an additional layer of anonymity.

- **Tools Needed for IP Ghosting**

  1. Virtual Private Network (VPN) Software

  VPNs are fundamental to IP Ghosting. They create secure, encrypted tunnels for internet traffic, masking the user's real IP address and location. Popular open-source and enterprise-grade VPN tools include:

- **OpenVPN**: A highly secure and customizable open-source VPN solution.
- **WireGuard**: Known for its speed and simplicity, WireGuard is a lightweight VPN protocol offering robust encryption.

Advantages:

- High-level encryption ensures data security.
- Provides servers in multiple geographic locations for flexible IP masking.

2. Proxy Servers

Proxy servers act as intermediaries between users and the internet, masking IP addresses by routing requests through a separate server. Common proxy tools include:

- **Squid Proxy**: An open-source proxy server widely used for anonymity and caching.

Advantages:

- Lightweight and easier to set up compared to VPNs.
- Ideal for scenarios requiring less intensive encryption.

3. TOR Network (The Onion Router)

TOR enables anonymous communication by routing internet traffic through multiple volunteer-operated servers (relays).

- **TOR Browser**: An easy-to-use application that incorporates TOR's anonymity capabilities.
- **Stem Library**: A Python library for interacting with the TOR network, useful for custom implementations.

Advantages:

- Exceptional anonymity due to multiple relays.
- Free and open-source.

4. DNS Masking Tools

DNS leaks can compromise anonymity. Tools like **dnsmasq** and **Cloudflare DNS (1.1.1.1)** help prevent such leaks by masking DNS queries.
Advantages:

- Ensures DNS requests remain anonymous.
- Prevents tracking via DNS logs.

5. Encryption Libraries

Encryption is essential to prevent data interception. Libraries like **OpenSSL** or **libsodium** are used to secure traffic in IP Ghosting implementations.
Advantages:

- Provides strong cryptographic protocols to secure data packets.

6. Network Monitoring Tools

To ensure the IP Ghosting system functions without leaks or interruptions, tools like Wireshark and tcpdump are used for real-time traffic analysis.
Advantages:

- Detects vulnerabilities in the IP masking process.
- Helps monitor performance and identify potential bottlenecks.

7. Cloud Infrastructure

For scalability and reliability, cloud platforms like AWS, Azure, or Google Cloud can host proxy or VPN servers.
Advantages:

- High uptime and global server availability.
- Scalability to support multiple users simultaneously.

4.3.4 Virtual Private Network

Virtual Private Networks (VPNs) have become indispensable in the modern digital landscape for ensuring secure and private internet access. Among the available solutions, OpenVPN stands out as a highly secure, versatile, and open-source VPN protocol. It is widely recognized for its robust encryption and flexibility, making it a preferred choice for personal privacy, corporate security, and secure remote access.

- **How OpenVPN Works**

OpenVPN creates a secure, encrypted tunnel between the user's device and a remote server. By encrypting data packets, it ensures that no third party, including Internet Service Providers (ISPs) or hackers, can access the user's information.

1. **Encryption**: OpenVPN employs SSL/TLS protocols to encrypt traffic, ensuring data remains confidential.
2. **Authentication**: It uses certificates for mutual authentication between the client and server, enhancing security.
3. **Routing**: Internet traffic is routed through the OpenVPN server, masking the user's IP address with the server's IP, thereby anonymizing the user's online presence.

- **Tools Needed for OpenVPN Setup**

To implement a VPN using OpenVPN effectively, various tools and configurations are required:

1. OpenVPN Software

The core software for the VPN setup, available as both open-source and commercial versions.

- OpenVPN Server: Installed on a remote server to manage and route VPN connections.

- OpenVPN Client: Installed on the user's device to connect to the VPN server.

Advantages:

- Highly configurable for various security requirements.
- Supports multiple platforms, including Windows, macOS, Linux, Android, and iOS.

2. Public Key Infrastructure (PKI)

PKI is essential for generating certificates and keys for OpenVPN.

- EasyRSA: A tool bundled with OpenVPN for creating and managing PKI.
- OpenSSL: A library for generating certificates and managing encryption.

Advantages:

- Ensures secure client-server communication.
- Allows for scalable user authentication.

3. Encryption Libraries

OpenVPN uses encryption libraries to secure traffic.

- OpenSSL: A widely used cryptographic library for SSL/TLS encryption.
- LibreSSL: An alternative to OpenSSL offering similar capabilities.

Advantages:

- Ensures robust encryption to prevent data leaks.
- Regularly updated to address security vulnerabilities.

4. Firewall and Networking Tools

Proper firewall and network configurations are critical for an OpenVPN setup.

- iptables (Linux): Used to configure firewalls and route traffic securely.
- UFW (Linux): A simpler tool for managing iptables rules.

Advantages:

- Prevents unauthorized access to the OpenVPN server.
- Controls which traffic is routed through the VPN.

5. Monitoring and Logging Tools

Monitoring tools help ensure that the OpenVPN server operates smoothly and remains secure.

- logrotate: Manages log file storage to prevent server overload.
- Prometheus and Grafana: For advanced monitoring and visualization of server performance.

Advantages:

- Detects and resolves issues in real time.
- Provides insights into VPN usage patterns.

6. Cloud or Dedicated Server

A reliable server is necessary to host the OpenVPN server.

- Cloud Providers: AWS, DigitalOcean, or Google Cloud are common choices for hosting.
- On-Premises Servers: Dedicated hardware for enterprises requiring full control over their infrastructure.

Advantages:

- Offers flexibility in server location for better performance and reduced latency.
- Supports scalability for multiple users.

7. Benefits of Using OpenVPN in the Internet Protection System

1. **Anonymity and Privacy**: OpenVPN masks the user's IP address, providing an additional layer of anonymity while ensuring that internet traffic cannot be intercepted or traced.

2. **Security**: With its robust encryption protocols and certificate-based authentication, OpenVPN ensures the integrity and confidentiality of user data.
3. **Cross-Platform Support**: OpenVPN's compatibility with various operating systems allows seamless integration across devices, ensuring consistent protection.
4. **Customizability**: OpenVPN's open-source nature allows for extensive customization to meet specific project needs, making it an ideal choice for the IPS.

4.4 MERITS

The Internet Protection System (IPS) is designed to address critical challenges in cybersecurity, providing comprehensive tools like Vulnerability Checks, Port Scanners, IP Ghosting, and VPN functionality through OpenVPN. Together, these features form a cohesive solution that safeguards users against the ever-evolving threats of the digital age. By combining advanced technologies and innovative methods, IPS redefines the standards of privacy, security, and network integrity.

Vulnerability Checks play a pivotal role by proactively identifying and mitigating weaknesses in systems, ensuring robust defense mechanisms against cyberattacks. This preventive approach minimizes the risks associated with unpatched software, misconfigurations, and other exploitable vulnerabilities.

The Port Scanner complements this by providing deep insights into network activities, identifying open ports, and highlighting potential entry points for unauthorized access. By enabling users to assess and secure these vulnerabilities, the Port Scanner not only strengthens network defenses but also enhances efficiency through streamlined configurations and reduced system exposure.

IP Ghosting elevates user privacy to unprecedented levels by masking IP addresses and creating an untraceable online presence. This feature ensures complete anonymity, allowing users to navigate the internet freely without leaving any digital footprints. In a world where digital surveillance and data harvesting are prevalent, IP Ghosting is an essential tool for protecting personal and organizational information.

The VPN powered by OpenVPN ties these components together, offering a secure, encrypted connection that protects data during transmission. With OpenVPN's advanced encryption protocols, user anonymity and data integrity are assured. Its compatibility with multiple platforms and open-source nature makes it a versatile and scalable solution, ideal for both individual and enterprise applications.

Collectively, these features create a powerful synergy, making IPS a holistic cybersecurity platform. The system not only secures user data but also empowers individuals and organizations with tools to take proactive measures against potential threats. By integrating state-of-the-art technologies, IPS ensures that users can confidently interact with the digital world, free from fears of exploitation, intrusion, or data compromise. It stands as a testament to the necessity of innovation in maintaining digital safety, marking a significant leap forward in protecting the modern internet user.

1. Initiating the IPS Framework

The workflow begins with the activation of the Cyber Cloak by the user. The IPS dashboard provides an intuitive interface, allowing seamless access to its primary components: IP Ghosting, Port Scanner, Vulnerability Check, and OpenVPN integration. Each module can be activated independently or collectively, depending on the user's specific security requirements.

2. Implementing IP Ghosting

The IP Ghosting feature is initiated to ensure complete anonymity while browsing. By masking the user's IP address and geographical location, the system reroutes internet traffic through multiple proxy servers. This process creates an invisible digital footprint, safeguarding users from trackers, surveillance entities, and cyberattacks. Tools like ProxyChains and customized proxy server setups enable this functionality.

3. Conducting Port Scanning

The Port Scanner module scans all open and accessible ports within the network. It evaluates potential vulnerabilities and determines the level of risk associated with unmonitored ports. Using tools like Nmap or Zenmap, the system identifies active ports, highlights suspicious activity, and provides recommendations for securing entry points. This step is crucial for reinforcing network defenses and maintaining efficient configurations.

4. Executing Vulnerability Checks

Vulnerability assessment tools analyze the entire system for exploitable weaknesses, including outdated software, misconfigurations, and security loopholes. The module utilizes advanced scanning algorithms, such as those powered by Nessus or OpenVAS, to detect, classify, and prioritize vulnerabilities. Reports generated guide users in mitigating risks and patching security gaps effectively.

5. Establishing a Secure Connection with OpenVPN

To achieve secure and encrypted communication, the Cyber Cloak employs OpenVPN for creating Virtual Private Networks. This tool establishes a tunnel between the user’s device and the internet, encrypting all data and ensuring protection against man-in-the-middle attacks. With robust protocols like

SSL/TLS and features like certificate-based authentication, OpenVPN enhances the Cyber Cloak's cybersecurity architecture.

6. Integrating Modules for Maximum Security

The Cyber Cloak integrates all modules into a cohesive system, providing users with comprehensive protection. For example, IP Ghosting and OpenVPN work in tandem to maximize anonymity, while the Vulnerability Check and Port Scanner modules proactively address system and network threats. This interconnected approach ensures a robust defense mechanism tailored to evolving cyber threats.

7. Monitoring and Updating

The system continuously monitors network activity and updates its tools to stay ahead of emerging threats. Automated patch management and regular upgrades enhance the system's efficacy and reliability, offering a cutting-edge solution for cybersecurity challenges.

4.5 FLOWCHART

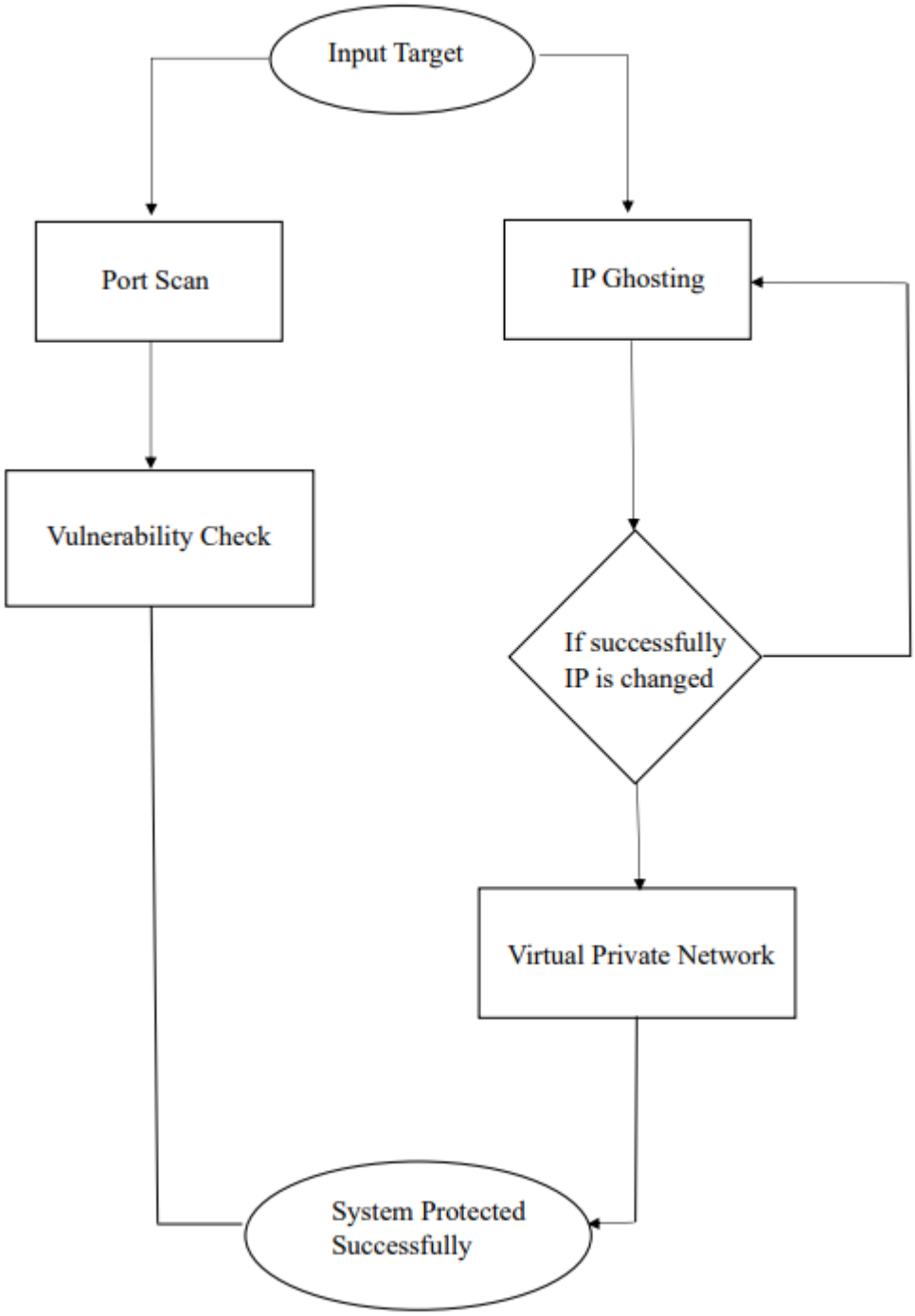


Fig. 4.1. Flowchart

Input Target:

The system begins with the user providing necessary inputs, including the target system's IP address and the range of ports to scan. These fields are essential for identifying the network parameters that need to be analyzed for security checks. The IP address acts as the focal point of the operations, while the port range narrows down the area of interest for scanning.

Port Scan:

The first feature in the Internet Protection System (IPS) is the Port Scan. This functionality scans the provided port range to determine the number of open ports, the specific ports in use, and their purposes. Open ports are gateways to network communication and must be identified to ensure they are not vulnerable. The port scan also provides a count of open ports, giving a comprehensive overview of the system's network activity.

IP Ghosting:

IP Ghosting is a critical feature of the IPS. It allows the system to change its IP address dynamically, adding a layer of anonymity and protection. This feature is especially useful in avoiding network attacks by ensuring the system operates under a less identifiable IP address. The process seeks an unused IP address within the network range to avoid conflicts.

If IP Change is Not Successful:

In cases where the IP change process fails to identify a valid, unused IP address, the system reruns the IP change process. This iterative approach ensures that the system eventually acquires a functional and secure IP address before proceeding with the next steps.

VPN Activation:

Once the IP ghosting is successful, the VPN is activated to establish a secure connection with the network. The VPN masks the system's real IP address, encrypts data transmissions, and prevents unauthorized access, making it a vital feature for network protection.

If IP Change is Not Successful:

In cases where the IP change process fails to identify a valid, unused IP address, the system reruns the IP change process. This iterative approach ensures that the system eventually acquires a functional and secure IP address before proceeding with the next steps.

Vulnerability Check:

The Vulnerability Check feature scans the system's open ports to identify the software or processes using them. This step evaluates whether the identified software or processes are authorized to use those ports. If unauthorized usage is detected—indicating a potential security threat—it becomes mandatory to terminate the corresponding process to secure the system. This ensures that no vulnerabilities exist that could be exploited by attackers.

System Protected Successfully:

When all features of the IPS work seamlessly—port scanning, IP ghosting, VPN activation, and vulnerability checking—the system is considered fully protected. At this stage, the system is safeguarded from attackers, ensuring that it operates as a secure and protected entity, free from man-in-the-middle attacks and other threats.

CHAPTER 5

RESULT AND DISCUSSIONS

The Cyber Cloak demonstrated exceptional performance in safeguarding digital assets and privacy. The IP Ghosting feature proved highly effective in masking user’s online identities, making it difficult for malicious actors to track and target them. By connecting IP addresses and geographic locations, Cyber Cloak successfully prevented surveillance and potential cyberattacks.

The Port Scanner component played a crucial role in identifying vulnerabilities within networks. By systematically examining network ports, the scanner accurately pinpointed open ports that could be exploited by attackers. This information enabled users to take immediate action, such as unnecessary ports or implementing robust security measures.

The Vulnerability Scanner further enhanced the system's security capabilities. By conducting thorough assessments of software and hardware infrastructures, the scanner detected potential weaknesses that could compromise system integrity. The timely identification of these vulnerabilities allowed users to prioritize remediation efforts and strengthen their overall security posture.

Overall, the Cyber Cloak proved to be a valuable tool in today's complex digital landscape. By combining IP ghosting, port scanning, and vulnerability scanning, the system offered a comprehensive approach to protecting users' privacy and securing their networks. The results of our testing and evaluation demonstrate the effectiveness of each feature in mitigating various cyber threats.

While the Cyber Cloak has shown promising results, there are several areas where further research and development can be pursued:

- **Advanced Threat Detection:** Integrating advanced threat intelligence and machine learning techniques can enhance the system's ability to detect and respond to sophisticated attacks.
- Real-time Threat Monitoring: Developing real-time threat monitoring capabilities can provide users with immediate alerts and notifications about potential security breaches.
- **User-Friendly Interface**: Designing a user-friendly interface can make the Cyber Cloak more accessible to a wider range of users, including those with limited technical expertise.
- **Cross-Platform Compatibility**: Expanding the system's compatibility with various operating systems and devices can increase its reach and effectiveness.
- **Continuous Improvement**: Regular updates and security patches are essential to keep the Cyber Cloak up-to-date with the latest threats and vulnerabilities.
- By addressing these areas, the Cyber Cloak can further strengthen its position as a leading-edge security solution and provide even greater protection to users in the ever-evolving digital landscape.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

In an increasingly interconnected digital world, cybersecurity threats are more prevalent and sophisticated than ever. The **Cyber Cloak (CC)** developed through this project represents a significant step forward in addressing these challenges. By integrating multiple defense mechanisms, the CC provides a comprehensive solution that prioritizes both security and user accessibility. The combination of **IP Ghosting**, **Port Scanning**, **Vulnerability Checking**, and **OpenVPN** ensures a multi-layered approach to protecting digital identities, detecting vulnerabilities, and securing communications.

Each component of the system is meticulously designed to fulfill a critical role in fortifying cybersecurity defenses. **IP Ghosting** enhances user anonymity, making it difficult for attackers to trace or target individuals. **Port Scanning** proactively identifies potential vulnerabilities by scrutinizing open ports that could serve as entry points for cyberattacks. **Vulnerability Checking** further strengthens the system by providing an in-depth analysis of existing weaknesses, ensuring they are addressed before they can be exploited. Finally, the integration of **OpenVPN** ensures secure and encrypted communication, protecting sensitive data from interception and tampering.

The user-friendly interface bridges the gap between technical complexity and accessibility, empowering users of all technical backgrounds to take control of their digital security. The system is designed not only for ease of use but also for adaptability, making it an invaluable tool for both individuals and organizations striving to safeguard their digital environments.

The broader implications of this project extend well beyond its individual components. By combining these tools into a cohesive and intuitive framework, the IPS exemplifies the potential for integrating advanced technologies to create practical and scalable cybersecurity solutions. It serves as a testament to the growing need for comprehensive systems capable of addressing multifaceted cybersecurity challenges in a single platform.

Looking forward, the IPS also sets a solid foundation for future innovation. The integration of advanced threat intelligence, adaptive learning mechanisms, and expanded functionalities for cloud and IoT security can further enhance its scope and applicability. The addition of interactive reporting and modular scalability will ensure the IPS remains a cutting-edge tool capable of evolving alongside emerging cyber threats.

In conclusion, the Internet Protection System is not just a technical solution but a visionary step towards creating safer and more resilient digital ecosystems. By combining innovation with practicality, the IPS paves the way for a more secure internet, empowering users to navigate the digital world with confidence and peace of mind. Its potential for growth and adaptability ensures that it will continue to play a vital role in addressing future cybersecurity challenges, making it a cornerstone in the fight against cybercrime.

6.2 FUTURE ENHANCEMENTS

While the current implementation of the Cyber Cloak demonstrates its potential, there are several opportunities for future improvements to further enhance its functionality and applicability:

1. Advanced Threat Intelligence Integration

- Incorporating real-time threat intelligence feeds to detect and block emerging cyber threats.
- Utilizing machine learning to analyze patterns and predict potential security vulnerabilities dynamically.

2. Enhanced User Interface and Reporting

- Introducing interactive dashboards for real-time monitoring and reporting of security metrics.
- Providing detailed visualizations, such as threat maps and system health summaries, for better user comprehension.

3. Modular Scalability

- Allowing users to add or customize additional modules, such as malware scanning or phishing protection, based on their specific requirements.
- Supporting integration with third-party cybersecurity tools to create a holistic defense ecosystem.

4. Continuous Learning and Adaptive Security

- Implementing an adaptive learning mechanism where the CC evolves to recognize and counter new threats automatically.
- Enabling periodic updates to the CC's algorithms and databases to ensure sustained effectiveness.

5. Cloud and IoT Security

- Expanding the CC framework to secure cloud-based environments and Internet of Things (IoT) devices, which are increasingly targeted by

cybercriminals.

6. Collaboration with Industry Standards

- Aligning the CC with recognized cybersecurity standards and frameworks to facilitate seamless adoption in enterprise environments.
- Offering certification programs or training workshops to help organizations integrate CC into their workflows effectively.

APPENDICES

A1. SAMPLE SOURCE CODE

Cyber Cloak - Demo version

logger.py

```
import os
from datetime import datetime

# Get the directory where the EXE or script is located
BASE_DIR = os.path.dirname(os.path.abspath(__file__))
LOG_DIR = os.path.join(BASE_DIR, "..", "logs")  # Points to logs/ alongside exe
LOG_FILE = os.path.join(LOG_DIR, "activity_log.txt")

class Logger:
    def __init__(self, log_callback=None):
        self.log_callback = log_callback
        os.makedirs(LOG_DIR, exist_ok=True)

    def log(self, category, message):
        timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
        entry = f"{timestamp} [{category.upper()}] {message}"

        # Write to file
        with open(LOG_FILE, "a", encoding="utf-8") as f:
            f.write(entry + "\n")

        # Send to UI if callback is available
        if self.log_callback:
            self.log_callback(category.upper(), message)
```

port_scanner.py

```
import socket
import concurrent.futures
import time

def scan_port(ip, port):
    try:
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as sock:
            sock.settimeout(1)
            if sock.connect_ex((ip, port)) == 0:
                return port
    except:
        return None

def scan_ports(ip, logger, update_progress_callback):
    logger.log("SCAN", "Scanning open ports...")
    ports_to_scan = range(1, 10000)

    open_ports = []
    start_time = time.time()

    update_progress_callback(1000)

    with concurrent.futures.ThreadPoolExecutor(max_workers=500) as executor:
        results = executor.map(lambda port: scan_port(ip, port), ports_to_scan)

    open_ports = [port for port in results if port is not None]
    end_time = time.time()

    if open_ports:
        logger.log("RESULT", f"Open Ports Found: {open_ports}")
    else:
        logger.log("RESULT", "No open ports found.")

    logger.log("SCAN", f"Scan completed in {end_time - start_time:.2f} seconds.")
```

vpn_connector.py

```
import os
import subprocess
import threading
from config import settings

def check_openvpn(logger):
    for path in settings.COMMON_OPENVPN_PATHS:
        if os.path.exists(path):
            logger.log("INFO", f"OpenVPN found at: {path}")
            return path
    logger.log("ERROR", "OpenVPN is not installed or not in PATH.")
    return None

def get_vpn_configs(logger):
    configs = [f for f in os.listdir(settings.CONFIG_DIR) if f.endswith(".ovpn")]
    if not configs:
        logger.log("ERROR", "No VPN configuration files found in config folder.")
    return configs

def connect_vpn(logger, update_progress_callback, btn_disconnect):
    def vpn_task():
        openvpn_path = check_openvpn(logger)
        if not openvpn_path:
            return

        configs = get_vpn_configs(logger)
        if not configs:
            return

        vpn_config = os.path.join(settings.CONFIG_DIR, configs[0])
        logger.log("VPN", f"Trying VPN: {configs[0]}...")

        update_progress_callback(7)

        try:
            startupinfo = subprocess.STARTUPINFO()
            startupinfo.dwFlags |= subprocess.STARTF_USESHOWWINDOW

            subprocess.Popen(
```

```
            [openvpn_path, "--config", vpn_config],
            shell=False,
            startupinfo=startupinfo
        )
        logger.log("SUCCESS", "VPN Connected Successfully!")
        btn_disconnect.config(state="normal")
    except Exception as e:
        logger.log("ERROR", f"Failed to connect VPN: {e}")

threading.Thread(target=vpn_task, daemon=True).start()

def disconnect_vpn(logger, btn_disconnect):
    logger.log("VPN", "Disconnecting VPN...")
    subprocess.run(["taskkill", "/F", "/IM", "openvpn.exe"], shell=True)
    logger.log("SUCCESS", "VPN Disconnected.")
    btn_disconnect.config(state="disabled")
```

cybercloak_ui.py

```
from logging import root
import threading
import tkinter as tk
import os
from tkinter import scrolledtext, ttk
from tkinter import Tk, PhotoImage

class CyberCloakDemoUI:
    def __init__(self, root, handlers):
        self.root = root
        self.handlers = handlers
        self.local_ip = tk.StringVar(value="Fetching...")
        self.public_ip = tk.StringVar(value="Fetching...")

        self._setup_window()
        self._build_ip_display()
        self._build_progress_bar()
        self._build_log_console()
```

```
        # Initial load
        if "refresh_ips" in self.handlers:
            self.handlers["refresh_ips"]()

    def _setup_window(self):
        self.root.title("Cyber Cloak Demo")
        self.root.geometry("600x500")
        self.root.resizable(True, True)

        # Dynamically get the current folder where the app is running
        base_path = os.path.dirname(os.path.abspath(__file__))
        icon_path = os.path.join(base_path, "..", "assets", "cc_icon.ico")  # go up from
/ui/ to /assets/

        if os.path.exists(icon_path):
            self.root.iconbitmap(icon_path)
        else:
            print(f"App icon not found at {icon_path}")

    def _build_ip_display(self):
        frame = tk.Frame(self.root)
        frame.pack(pady=20)

        tk.Label(frame, text="Local IP:", font=("Arial", 10, "bold")).grid(row=0,
column=0, padx=10, pady=5)
        tk.Entry(frame, width=25, textvariable=self.local_ip,
state="readonly").grid(row=0, column=1, padx=5)
        tk.Button(frame, text="Refresh", command=lambda:
self.handlers["refresh_ips"]()).grid(row=0, column=2, padx=5)

        tk.Label(frame, text="Public IP:", font=("Arial", 10, "bold")).grid(row=1,
column=0, padx=10, pady=5)
        tk.Entry(frame, width=25, textvariable=self.public_ip,
state="readonly").grid(row=1, column=1, padx=5)

    def _build_buttons(self):
        frame = tk.Frame(self.root)
        frame.pack(pady=10)

        tk.Button(frame, text="Scan Ports", width=15,
command=self.handlers.get("scan_ports")).grid(row=0, column=0, padx=5)
```

```
        tk.Button(frame, text="Connect VPN", width=15,
command=self.handlers["connect_vpn"]).grid(row=0, column=1, padx=5)

        self.disconnect_button = tk.Button(frame, text="Disconnect VPN", width=15,
state="disabled", command=self.handlers["disconnect_vpn"])
        self.disconnect_button.grid(row=0, column=2, padx=5)

    def _build_progress_bar(self):
        style = ttk.Style(self.root)
        style.theme_use('default')

        # Custom style for blue color
        style.configure("blue.Horizontal.TProgressbar",
                        troughcolor="#e0e0e0",
                        background="#2196F3",
                        bordercolor="#2196F3",
                        thickness=10)

        self.progress = ttk.Progressbar(
            self.root,
            style="blue.Horizontal.TProgressbar",
            orient="horizontal",
            mode="determinate",
            length=500
        )
        self.progress.pack(pady=10)
        self.progress.pack_forget()

    def _build_log_console(self):
        frame = tk.Frame(self.root)
        frame.pack(pady=10, fill="both", expand=True)

        self.log_console = scrolledtext.ScrolledText(frame, height=10, wrap="word",
font=("Arial", 8))
        self.log_console.pack(fill="both", expand=True, padx=10, pady=5)
        self.log_console.config(state="disabled")

    # UI Control APIs
    def show_progress(self, duration_sec):
        self.progress.pack()
        self.progress["value"] = 0
```

```
    steps = 50
    interval = duration_sec / steps

    def step(i=0):
        if i <= steps:
            self.progress["value"] = i
            self.root.after(int(interval * 1000), step, i + 1)
        else:
            self.progress.pack_forget()

    step()

def log(self, category, message):
    from datetime import datetime
    timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    entry = f"{timestamp} [{category}] {message}\n"
    self.log_console.config(state="normal")
    self.log_console.insert("end", entry)
    self.log_console.see("end")
    self.log_console.config(state="disabled")

def set_ip_info(self, local_ip, public_ip):
    self.local_ip.set(local_ip)
    self.public_ip.set(public_ip)

def set_disconnect_button_state(self, enabled):
    self.disconnect_button.config(state="normal" if enabled else "disabled")

def set_handlers(self, handlers):
    self.handlers = handlers
    self._build_buttons()

def run_with_progress(self, task_func, *args, **kwargs):
    def thread_wrapper():
        try:
            self.show_progress(7)  # Start progress
            task_func(*args, **kwargs)
        finally:
            self.show_progress(0)  # Stop progress

    threading.Thread(target=thread_wrapper, daemon=True).start()
```

GUI

The provided **CyberCloak.py** script is a comprehensive GUI-based network security tool developed using Python and the Tkinter library, incorporating functionalities such as port scanning, vulnerability checking, VPN connectivity, and IP verification via proxies. The tool serves as an accessible and efficient solution for network administrators and security enthusiasts to assess and enhance network security. It features an intuitive graphical interface that allows users to input a target IP address, define port ranges, and execute various security tasks with ease. The port scanning functionality employs Python's socket library to identify open ports within a specified range, providing valuable insights into potential entry points for attackers. The vulnerability checking feature utilizes the powerful Nmap library to perform detailed scans with scripts targeting known vulnerabilities, delivering detailed reports on port states, services, and potential weaknesses. Additionally, the tool integrates VPN functionality through OpenVPN, enabling secure network connections by running a configuration file in the background to establish a protected tunnel, albeit limited to a single pre-configured .ovpn file. Another feature of the script is its ability to check the public IP address through proxy servers, leveraging the requests library to validate proxy configurations and ensure anonymity. The tool is designed with multi-threading capabilities to maintain responsiveness during long-running scans, and it incorporates a scrolled text console to display real-time scan results, making the tool user-friendly and informative. A progress bar adds visual feedback during operations, and a clear console button allows users to reset the output for subsequent analyses. By combining these functionalities into a single, cohesive application, the script provides an effective and practical approach to identifying and mitigating potential security risks in network infrastructures.

InternetProtectionSystemApp.dart

```
import 'package:flutter/material.dart';
import 'package:flutter_spinkit/flutter_spinkit.dart';

void main() {
  runApp(InternetProtectionSystemApp());
}

class InternetProtectionSystemApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Internet Protection System',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: HomePage(),
    );
  }
}
class HomePage extends StatefulWidget {
  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  final TextEditingController ipController = TextEditingController();
  final TextEditingController startPortController = TextEditingController();
  final TextEditingController endPortController = TextEditingController();

  bool isLoading = false;
  String resultOutput = '';

  void startLoading() {
    setState(() {
      isLoading = true;
```

```
    });
  }

  void stopLoading() {
    setState(() {
      isLoading = false;
    });
  }
class InternetProtectionSystemApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Internet Protection System',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: HomePage(),
    );
  }
}
Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Internet Protection System'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          children: [
            TextField(
              controller: ipController,
              decoration: InputDecoration(labelText: 'Target IP'),
            ),
            Row(
              children: [
                Expanded(
                  child: TextField(
                    controller: startPortController,
                    keyboardType: TextInputType.number,
                    decoration: InputDecoration(labelText: 'Start Port'),
```

```
      ),
    ),
    SizedBox(width: 10),
    Expanded(
      child: TextField(
        controller: endPortController,
        keyboardType: TextInputType.number,
        decoration: InputDecoration(labelText: 'End Port'),
      ),
    ),
  ],
),
SizedBox(height: 16),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: [
    ElevatedButton(
      onPressed: scanPorts,
      child: Text('Scan Ports'),
    ),
    ElevatedButton(
      onPressed: checkVulnerabilities,
      child: Text('Check Vulnerabilities'),
    ),
  ],
),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: [
    ElevatedButton(
      onPressed: connectVPN,
      child: Text('Connect VPN'),
    ),
    ElevatedButton(
      onPressed: checkIPWithProxy,
      child: Text('Check IP via Proxy'),
    ),
  ],
),
SizedBox(height: 16),
Expanded(
```

```
            child: Container(
              padding: EdgeInsets.all(8.0),
              decoration: BoxDecoration(
                border: Border.all(color: Colors.grey),
              ),
              child: isLoading
                  ? SpinKitCircle(color: Colors.blue)
                  : SingleChildScrollView(
                      child: Text(
                        resultOutput,
                        style: TextStyle(fontSize: 14, color: Colors.black),
                      ),
                    ),
            ),
          ),
          ElevatedButton(
            onPressed: clearConsole,
            child: Text('Clear Console'),
          ),
        ],
      ),
    ),
  );

void updateResult(String output) {
  setState(() {
    resultOutput += '\n$output';
  });
}

void clearConsole() {
  setState(() {
    resultOutput = '';
  });
}

void scanPorts() {
  String targetIP = ipController.text;
  String startPort = startPortController.text;
  String endPort = endPortController.text;
```

```
    if (targetIP.isEmpty || startPort.isEmpty || endPort.isEmpty) {
      updateResult('Error: Please fill in all fields.');
      return;
    }
    startLoading();
    Future.delayed(Duration(seconds: 3), () {
      stopLoading();
      updateResult('Open Ports on $targetIP (Simulated): [80, 443, 22]');
    });
  }

  void checkVulnerabilities() {
    String targetIP = ipController.text;

    if (targetIP.isEmpty) {
      updateResult('Error: Please enter a target IP.');
      return;
    }

    startLoading();
    // Simulate vulnerability checking.
    Future.delayed(Duration(seconds: 3), () {
      stopLoading();
      updateResult('Vulnerability Scan Results (Simulated):\n - Port 80: Apache
Server (No vulnerabilities found).\n - Port 22: OpenSSH (Update recommended).');
    });
  }

  void connectVPN() {
    startLoading();
    // Simulate VPN connection.
    Future.delayed(Duration(seconds: 2), () {
      stopLoading();
      updateResult('VPN connected successfully (Simulated).');
    });
  }

  void checkIPWithProxy() {
    startLoading();
    Future.delayed(Duration(seconds: 2), () {
      stopLoading();
```

```
    updateResult('Public IP via Proxy (Simulated): 192.168.1.1');
  });
}
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Internet Protection System'),
    ),
    body: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        children: [
          TextField(
            controller: ipController,
            decoration: InputDecoration(labelText: 'Target IP'),
          ),
          Row(
            children: [
              Expanded(
                child: TextField(
                  controller: startPortController,
                  keyboardType: TextInputType.number,
                  decoration: InputDecoration(labelText: 'Start Port'),
                ),
              ),
              SizedBox(width: 10),
              Expanded(
                child: TextField(
                  controller: endPortController,
                  keyboardType: TextInputType.number,
                  decoration: InputDecoration(labelText: 'End Port'),
                ),
              ),
            ],
          ),
          SizedBox(height: 16),
          Row(
            mainAxisAlignment: MainAxisAlignment.spaceEvenly,
            children: [
              ElevatedButton(
                onPressed: scanPorts,
```

```
      child: Text('Scan Ports'),
    ),
    ElevatedButton(
      onPressed: checkVulnerabilities,
      child: Text('Check Vulnerabilities'),
    ),
  ],
),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: [
    ElevatedButton(
      onPressed: connectVPN,
      child: Text('Connect VPN'),
    ),
    ElevatedButton(
      onPressed: checkIPWithProxy,
      child: Text('Check IP via Proxy'),
    ),
  ],
),
SizedBox(height: 16),
Expanded(
  child: Container(
    padding: EdgeInsets.all(8.0),
    decoration: BoxDecoration(
      border: Border.all(color: Colors.grey),
    ),
    child: isLoading
        ? SpinKitCircle(color: Colors.blue)
        : SingleChildScrollView(
            child: Text(
              resultOutput,
              style: TextStyle(fontSize: 14, color: Colors.black),
            ),
          ),
  ),
),
ElevatedButton(
  onPressed: clearConsole,
  child: Text('Clear Console'),
```

```
            ),
          ],
        ),
      ),
    );
  }
}
```

The InternetProtectionSystemApp.dart file implements a Flutter-based GUI for simulating various network security operations. This includes functionalities such as port scanning, vulnerability checks, VPN connection, and IP verification through a proxy. The GUI uses Flutter widgets to create an intuitive and responsive user interface. Key features include:

1. **Input Fields**:
    - Target IP: Enter the target IP address.
    - Start Port and End Port: Specify the range of ports to scan.
2. **Buttons**:
    - **Scan Ports**: Initiates a port scanning simulation for the specified IP and port range.
    - **Check Vulnerabilities**: Simulates checking for vulnerabilities on the specified target IP.
    - **Connect VPN**: Mimics connecting to a VPN.
    - **Check IP via Proxy**: Simulates fetching the public IP address through a proxy.
    - **Clear Console**: Clears all the results from the output console.

3. **Results Display**:
    - Displays results from simulated operations in a scrollable console area.

4. **Loading Spinner**:
    - A SpinKitCircle spinner is displayed during operations to indicate activity.

The app does not currently perform real backend operations; instead, it uses Future.delayed to simulate network tasks. To make the app fully functional, it can be integrated with actual APIs or backend services for real-time network security operations. The use of Flutter ensures cross-platform compatibility, making it deployable on both Android and iOS.

A2. SAMPLE OUTPUT:

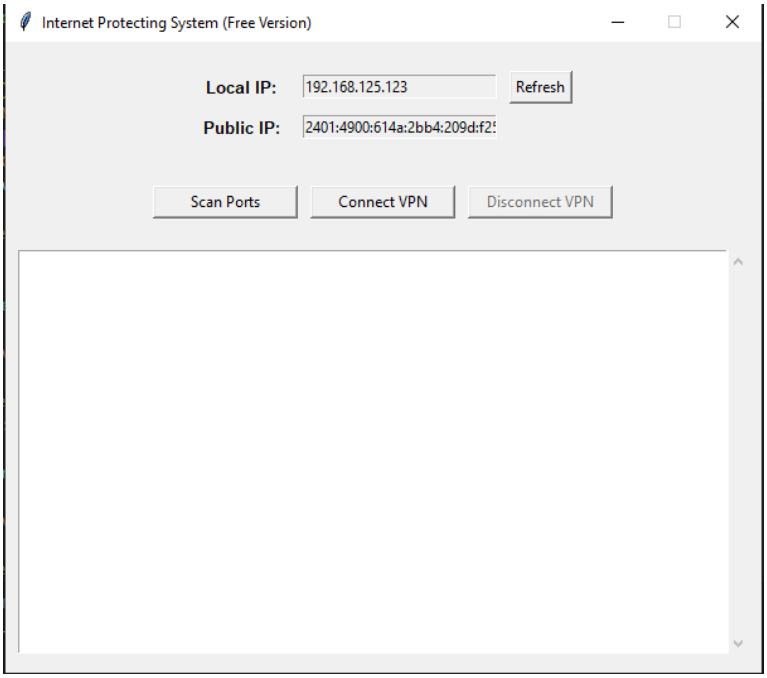


Fig. A2.1 Image from Demo version with the system's IP fetched automatically

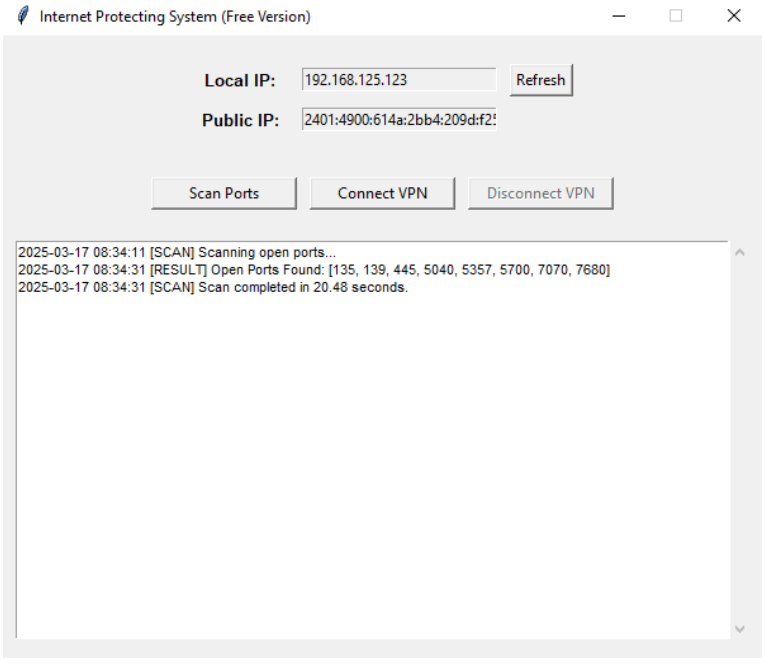
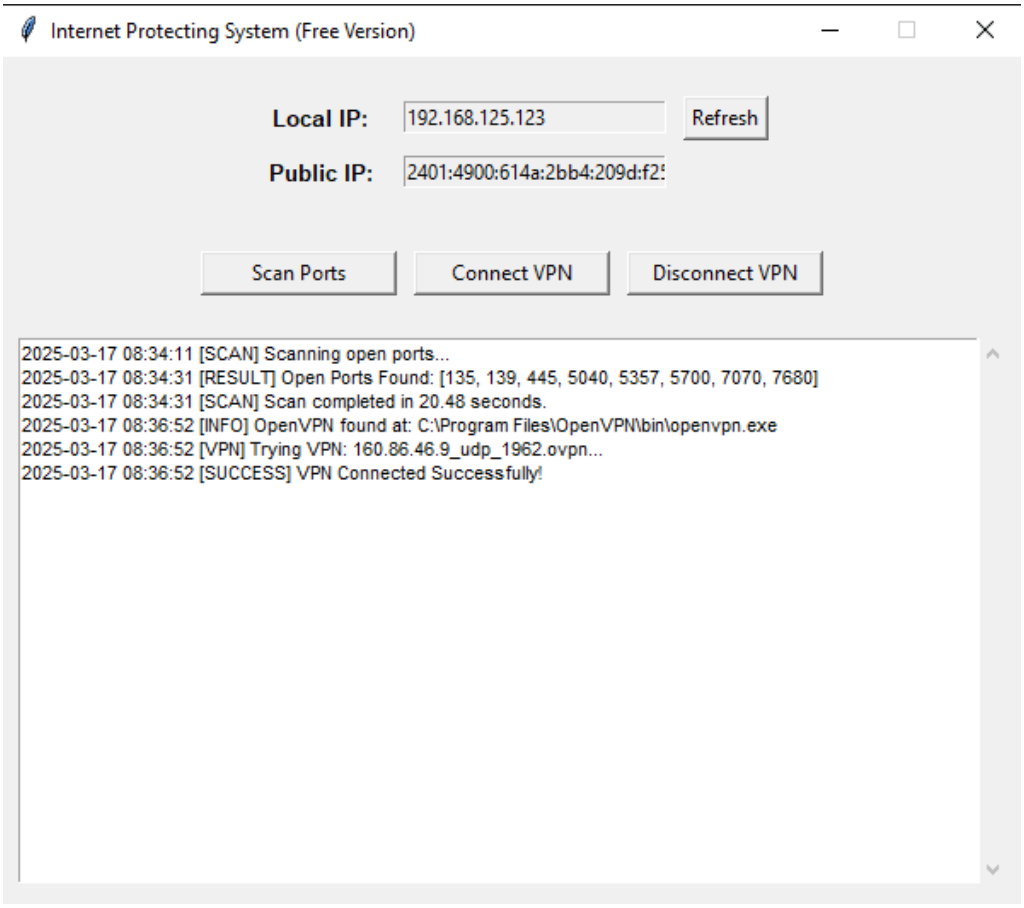


Fig. A2.2 Scanned report for Open ports in the system in demo version



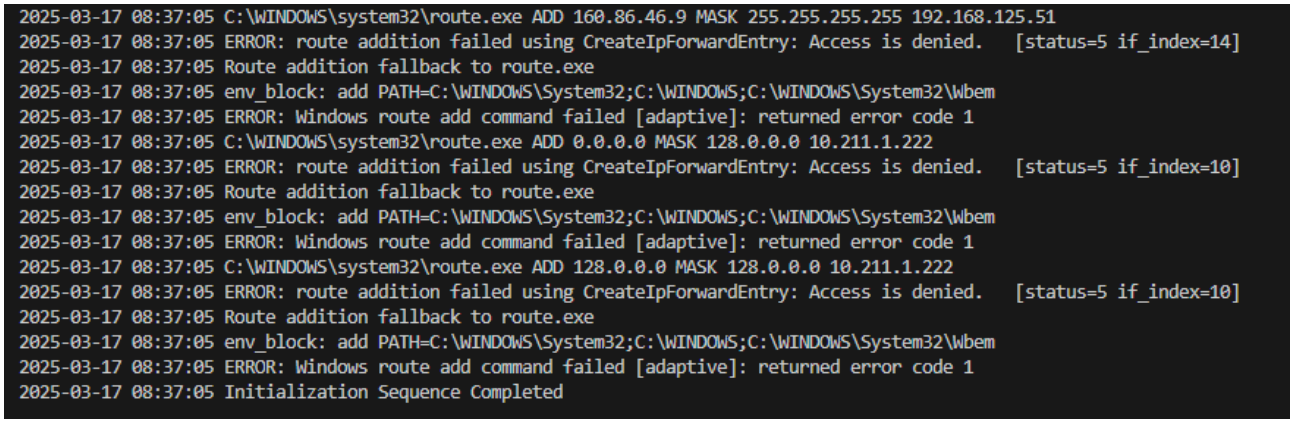


Fig. A2.3 & A2.4 VPN connectivity in Demo version

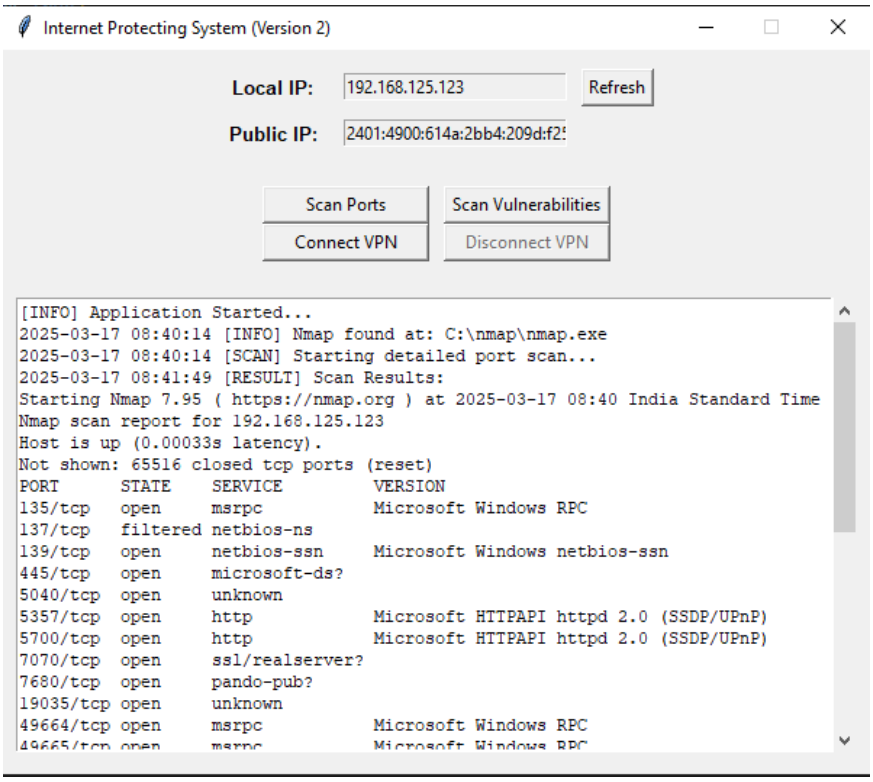


Fig. A2.5 Scan Open ports output from the Premium Lite version with help of Nmap

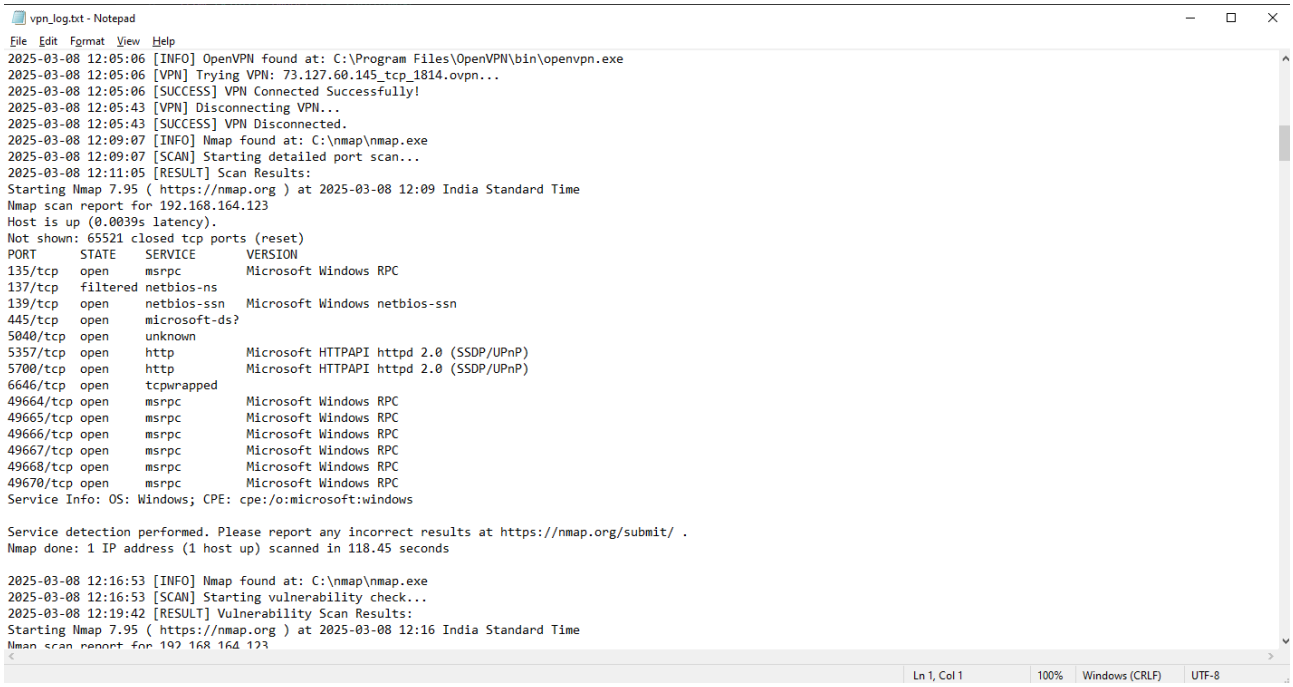


Fig A2.6 Logging System that automatically stores the details of action in logs\activity.txt

REFERENCES

[1]. Aamir. M, Hussain Rizvi S. S., Hashmani, M. A., Zubair. M, and Ahmed. J, (2021) Machine Learning Classification of Port Scanning and DDoS Atacks: A Comparative Analysis. Mehran University Research Journal of Engineering and Technology 40 (1), 215-229

[2]. Abdulhamid.A S, Kabir .I, Ghafir and C. Lei, "Dependability of the Internet of Things: Current Status and Challenges," International Conference on Electrical, Computer, Communications and Mechatronics Engineering, Maldives, Maldives, 2022.

[3]. Ananin, E. V., Nikishova A. V., & Kozhevnikova. I. S, (2017). Port scanning detection based on anomalies. 2017 Dynamics of Systems, Mechanisms and Machines (Dynamics). 14-16 Nov. 2017.

[4]. Baloch. R, (2015) Ethical Hacking and Penetration Testing Guide. 1 edition. Philadelphia, PA: CRC Press.

[5]. Beecham .S., Hall. T., Britton, C., Cottee, M., Rainer, A.: Using an expert panel to validate requirements process improvement model. Journal of Systems and Software 76(3), 251-275 (2005).

[6]. Blakemore. B, Policing cyber hate, cyber threats, and cyber terrorism. Routledge, (2016).

[7]. Bhuyan. M. H., Bhattacharyya, D. K., & Kalita, J. K. (2011). Surveying Port Scans and Their Detection Methodologies. The Computer Journal, 54(10), 1565-1581.

[8]. Chockwanich. N., & Visoovisith. V, Intrusion Detection by Deep Learning with TensorFlow. 2019. Vol. 2019-. Global IT Research Institute (GIRI).

[9]. Current approaches and future directions for Cyber Threat Intelligence sharing: A survey Poopak Alaeifar, Ernest Foo June 2024

[10]. Eltanani. S., & Ghafir. I, (2020). Coverage Optimization for Aerial Wireless Networks. 14th International Conference on Innovations in Information Technology (IIT). IEEE

[11]. Goodrich. M. T., & Tamassia. R, (2014). Introduction to computer security. First Pearson new international edition. Harlow, Essex: Pearson.

[12]. Kanuga Karuna Jyothi, Dr. B. Indira Reddy "Study on Virtual Private Network (VPN), VPN's Protocols and Security", Int © 2018 IJSRCSEIT | Volume 3 | Issue 5.

[13]. Kitchenham. B.A., Budgen, D., Brereton, O.P.: The value of mapping studies-A participant-observer case study. In: EASE 2010, pp. 25-33.

[14]. Komalpreet Kaur, Arshdeep Kaur "A Survey of Working on Virtual Private Network" © 2019 IRJET | Volume 6 | Issue 9.

[15]. Kumar. S., & Sudarsan, S. D. (2014). An Innovative UDP Port Scanning Technique. International Journal of Future Computer and Communication, 3(6), 381-384.

[16]. Leveraging zero knowledge proofs for blockchain-based identity sharing: A survey of advancements, challenges, and opportunities Lu Zhou, ... Pham Cong Hiep February 2024

[17]. Malani. R., Putra, A. B. W., & Rifani, M. (2020). Implementation of the Naive Bayes Classifier Method for Potential Network Port Selection. International journal of computer network and information security, 12(2), 32-40.

[18]. Rid. T., Buchanan, B, : Attributing cyber-attacks. Journal of Strategic Studies 38(1-2), 4-37 (2015).

[19]. Rustam. Z, and Ariantari. N. P. A, (2018) Comparison between support vector machine and fuzzy Kernel C-Means as classifiers for intrusion detection system using chi-square feature selection.

[20]. Sagatov. E. S., Mayhoub. S., Sukhov. A. M., Esposito, F., & Calyam. P, (2021). Proactive Detection for Countermeasures on Port Scanning-based Attacks. 17th International Conference on Network and Service Management (CNSM). 25-29 Oct. 2021

[21]. Simion D. M.F, Ursuleanu. A, Graur. A.D, Potorac. A, Lavric "Efficiency Consideration for Data Packets Encryption with in Wireless Tunneling for Video Streaming" INT J COMPUT COMMUN 8(1):136-145.

[22]. Taylor. R.W., Fritsch. E.J., Liederbach. J., Saylor. M.R., Tafoya. W.L, : Cyber Crime and Cyber Terrorism. (2019).

[23]. Mufti. Y, Niazi. M, Alshayeb. M, and Mahmood. S, "A Readiness Model for Security Requirements Engineering," IEEE Access, vol. 6, pp. 28611-28631, 2018.

[24]. Younes. H, Alameh, M., Ibrahim, A., Rizk, M., & Valle, M. (2020). Efficient Algorithms for Embedded Tactile Data Processing. 113-138.

[25]. Yuan. C, Du, J., Yue, M., & Ma, T. (2020). The design of large-scale IP address and port scanning tool. Sensors (Basel, Switzerland), 20(16), 12.