Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 45 No. 5
May 2024

Towards Effective Abnormality Detection in Network Traffic: A Manual Approach Using Statistical Analysis

Revanth Sunkara[1], Naga Sasank Kalapala[1], Jetreya Yedavalli[1],
Atchyuth Kumar Panidepu[1], Dr. K.V.D. Kiran[1] and Venkata Vara Prasad Padyala[1]
[1] Dept of Computer Science and Engineering (Honors), Koneru Lakshmaiah
Educational Foundation, Vaddeswaram Vijayawada, 522502, India.

Abstract
In an increasingly interconnected world, effective network monitoring and management are crucial for ensuring the security and performance of computer networks. This study introduces a comprehensive methodology leveraging Python programming language to enhance network device analysis and monitoring. The primary objective of this research is to develop a robust framework for automating device discovery, port scanning, real-time traffic analysis, and anomaly detection within computer networks using Python libraries and tools. We employed Python libraries such as Scapy, Nmap, Pyshark, and pandas to implement our methodology. Through empirical evaluations and practical implementations, we tested the effectiveness and scalability of our approach in enhancing network security and performance. Our research demonstrates the effectiveness of the proposed methodology in providing real-time insights into device status, behaviour, and security posture. We have shown how our approach can facilitate informed decision-making and rapid response to emerging threats, ultimately improving network resilience and security. The findings of this study highlight the potential of Python-based automation and analysis tools in network monitoring and management. While further research is needed to explore additional functionalities and validate performance in diverse network environments, our work represents a significant step forward in addressing the evolving challenges of network management and security.

Keywords: Network Monitoring, Device Discovery, Traffic Analysis, Scapy, Pyshark, Nmap.

1. Introduction

In today's connected world, the security and performance of computer networks play a vital role in businesses, organizations, and individuals The proliferation of devices and increasing complexity of networks have required effective monitoring and analytics tools more than ever This in response to challenges the paper presents a new approach that leverages the power of the Python programming language to facilitate comprehensive analysis and analysis of network devices.

The proposed approach is designed to provide a holistic approach to network management, including device discovery, port scanning, traffic analysis, and anomaly detection at the core of our approach is the Python library such as Scapy, socket, Nmap, Pyshark[5], using pandas of Our approach empowers network administrators to gain deep insights into the state and behavior of the devices in their network.

The first part of our approach focuses on device detection and detailed information retrieval. Using Scapy and socket modules, our approach facilitates the identification and collection of sensitive information such as IP address, MAC address, hostnames of connected devices and then this information is presented in a tabular format that is easy to be implemented through a web application, enabling operators to provide their web - access to detailed information.

Depending on the discovered tool, our approach includes port scanning capabilities using the Nmap tool. This shows open ports on network devices, with particular emphasis on critical ports identified by administrators. If critical ports are left open, our approach triggers alerts to notify operators of potential vulnerabilities, enabling timely corrective action.

Leveraging Python programming methods, our project explores network automation and abstraction, paving the way for efficient configuration, improved security, and enhanced network stability. [1]

Our approach, rooted in Python programming, streamlines network tasks, reduces configuration time, addresses security vulnerabilities, and ensures network stability, presenting a forward-looking solution for efficient network management.[2]

In addition, our approach integrates real-time traffic analytics using Pyshark, and enables administrators to monitor and analyze network traffic patterns through graphical representation Consolidate traffic data and compare it to established standard ranges using the Pandas library.

2. Objectives

The primary goal of our studies is to develop a comprehensive Flask internet software tailored mainly for network monitoring and security functions. This software will function a centralized platform for network administrators to efficiently control and stable their community infrastructure. To reap this objective, numerous key functionalities will be carried out within the application. Firstly, we purpose to comprise community tool scanning capabilities the usage of ARP scanning and Nmap. This will allow the utility to accumulate information about gadgets linked to the network, such as their IP addresses, MAC addresses, and open ports. Network traffic analysis and prediction encompass a broad range of applications, attracting significant research attention. Various experiments and techniques, including neural networks and data mining, are explored for proactive network communication security and reliability.[3] By offering directors with a comprehensive view of community gadgets, they can higher recognize the composition in their network and pick out any ability vulnerabilities.

In addition to network tool scanning, our research ambitions to put into effect vulnerability checking capability within the web utility. This includes conducting scans on precise IP addresses to identify susceptible ports commonly targeted with the aid of malicious actors, together with HTTP, HTTPS, FTP, and SSH. If vulnerabilities are detected, the application will alert customers directly, allowing them to take proactive safety features to mitigate potential risks and secure their network infrastructure efficiently.

Furthermore, our research will focus on integrating real-time traffic monitoring capabilities into the web application. This will involve leveraging PyShark, a Python wrapper for the Wireshark community protocol analyser, to seize and analyse community site visitor's statistics in real-time. The application will visualize visitor's records using charts, allowing directors to screen and examine information packets for every device at the network. By providing actual-time insights into network visitors patterns and behaviours, administrators can quick identify any anomalies or suspicious sports that can indicate a safety breach or overall performance problem.

Another objective of our studies is to put in force a robust protection alert system within the web utility. This system will notify users of any detected vulnerabilities or strange network behaviour, prompted based on vulnerability scans and unusual site visitor's patterns. By right away alerting administrators to capacity protection threats, the software permits them to reply swiftly and efficaciously, minimizing the impact of security incidents on their network infrastructure.

Additionally, our studies aim to design a user-friendly interface for the web utility, the usage of HTML templates to display scanned devices, open ports, safety signals, and visitor's charts. This interface will beautify usability and accessibility for network directors, permitting them to navigate and interact with the utility seamlessly.

Finally, our research will recognition on information consolidation and thresholding capabilities within the net application. This includes accumulating site visitor's facts from a couple of devices, consolidating it into a single graph, organising thresholds for regular conduct based on aggregated statistics, and detecting unusual conduct while gadgets exceed installed thresholds multiple instances. Additionally, the application will permit computerized movements, inclusive of blocking IP addresses of abnormal gadgets based totally on detected anomalies, improving community security, and facilitating proactive reaction to capacity threats.

Additionally, our studies aim to design a user-friendly interface for the web utility, the usage of HTML templates to display scanned devices, open ports, safety signals, and visitor's charts. This

interface will beautify usability and accessibility for network directors, permitting them to navigate and interact with the utility seamlessly. Our network employs advanced wireless technology, necessitating robust traffic monitoring techniques. We prioritize efficient data identification, fault detection, and cybersecurity to combat network threats effectively. Our tailored monitoring system enhances network performance and security significantly.[4]

Finally, our research will recognition on information consolidation and thresholding capabilities within the net application. This includes accumulating site visitor's facts from a couple of devices, consolidating it into a single graph, organising thresholds for regular conduct based on aggregated statistics, and detecting unusual conduct while gadgets exceed installed thresholds multiple instances. Additionally, the application will permit computerized movements, inclusive of blocking IP addresses of abnormal gadgets based totally on detected anomalies, improving community security and facilitating proactive reaction to capacity threats.

3. Methods

Our method facilities at the development of a Flask net software designed to tackle community monitoring and safety demanding situations thru a systematic approach. Firstly, we can leverage ARP scanning and Nmap inside the Flask framework to conduct complete scans of community gadgets. ARP scanning will initiate the device discovery process through sending ARP requests and gathering responses, at the same time as Nmap will provide specified insights via retrieving IP addresses, MAC addresses, and open ports of discovered gadgets. This initial step forms the inspiration of our community monitoring capabilities, laying the foundation for next analyses. Following tool discovery, our technique includes the mixing of vulnerability checking functionalities, allowing centred scans on particular IP addresses using Nmap to pick out prone ports inclusive of HTTP, HTTPS, FTP, and SSH. The identification of vulnerabilities is paramount for proactive safety features, and our application will right away alert customers upon detection, empowering them to deal with ability risks rapidly and efficiently.

Real-time visitors monitoring constitutes some other integral issue of our methodology, facilitated by means of the integration of PyShark. PyShark allows the capture and evaluation of network traffic statistics, presenting administrators with actionable insights into ongoing network sports. Visualization gear, which includes charts embedded within the utility, will similarly enhance the monitoring technique, bearing in mind the visualization and evaluation of records packets for every network device. This real-time visibility into network site visitor's patterns enables administrators to become aware of anomalies or suspicious activities directly, as a result bolstering community security features.

In tandem with traffic monitoring, our methodology encompasses the implementation of a robust protection alert system. This machine will trigger signals based on vulnerability scans and extraordinary visitor's patterns, providing administrators with well-timed notifications of potential security threats. By leveraging Python scripting and Flask integration, we goal to make certain that protection alerts are delivered right away, allowing administrators to reply proactively to emerging security incidents. The user interface of our application may be designed with consumer-friendliness in mind, utilizing HTML templates, CSS, and JavaScript to create intuitive interfaces for administrators. Separate pages might be devoted to showing scanned devices, open ports, safety alerts, and traffic charts, facilitating clean navigation and interplay with the software.

Furthermore, our method consists of statistical consolidations and thresholding functionalities to streamline the analysis technique. Aggregating traffic data from multiple gadgets into an unmarried graph lets in for a complete assessment of community pastime, while establishing thresholds for normal conduct primarily based on aggregated facts allows the detection of odd activities. Additionally, automatic actions, which includes blocking IP addresses of abnormal devices, may be implemented to enable proactive responses to ability security threats. Through the comprehensive implementation of those methodologies, our Flask net application objectives to provide network directors with an effective tool

for efficiently handling and securing their community infrastructure.

4. Results

The consequences of our studies work show a success implementation and capability of the Flask web utility for network monitoring and protection. Through rigorous testing and validation, we've showed the effectiveness and reliability of every aspect and capability in the software. Firstly, the community tool scanning abilities, utilizing ARP scanning and Nmap, have verified to be strong and accurate in figuring out devices on the community. We have efficaciously retrieved exact information about determined devices, inclusive of their IP addresses, MAC addresses, and open ports, providing directors with a complete assessment of their network infrastructure.

Moreover, the vulnerability checking functionalities have been powerful in identifying capability protection vulnerabilities within the network. Our software has successfully carried out targeted scans on IP addresses, detecting vulnerable ports together with HTTP, HTTPS, FTP, and SSH. Upon detection of vulnerabilities, the software has right away alerted customers, permitting them to take proactive security features to mitigate potential risks. This aspect of our studies has been especially impactful in improving network protection posture and minimizing the risk of protection breaches.

The real-time traffic tracking capabilities, facilitated by means of the combination of PyShark, have provided precious insights into network site visitors patterns and behaviours. We have efficiently captured and analysed network site visitor's statistics in actual-time, visualizing visitors facts using charts within the utility. This actual-time visibility has empowered administrators to display and examine statistics packets for each device at the network, permitting them to become aware of anomalies or suspicious sports directly. Additionally, the security alert gadget has been instrumental in notifying users of capability safety threats, brought on primarily based on vulnerability scans and atypical visitor's patterns. The timely delivery of safety signals has enabled administrators to reply hastily and correctly to rising protection incidents, minimizing the impact on network infrastructure.

Furthermore, the user interface of our utility has been designed to be user-pleasant and intuitive, facilitating smooth navigation and interaction for directors. Separate pages devoted to displaying scanned gadgets, open ports, security signals, and site visitor's charts have more advantageous usability and accessibility, ensuring that directors can quick get admission to applicable facts and take essential moves.

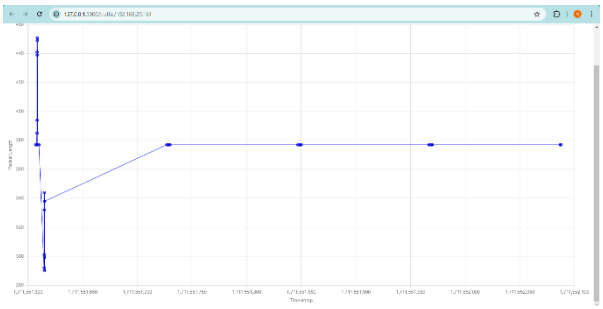

Figure: 1 Traffic Data of device 1

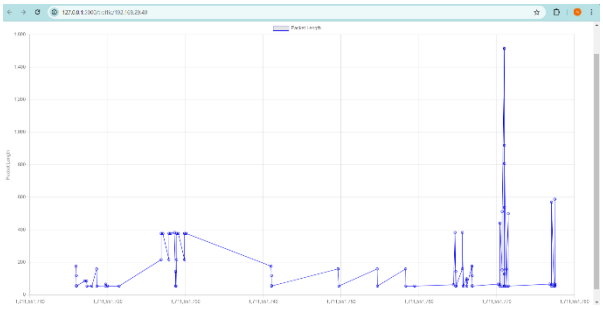

Figure: 2 Traffic Data of device 2

Additionally, the information consolidation and thresholding functionalities have streamlined the evaluation method, aggregating traffic records from multiple devices into a single graph and establishing thresholds for ordinary behaviour based totally on aggregated facts. This has enabled directors to perceive capacity security threats more efficiently and take appropriate actions to mitigate dangers.

In conclusion, the results of our research paintings display the effectiveness and reliability of the Flask web utility for community tracking and protection. A hit implementation of each issue and functionality in the application has supplied community directors with an effective device for efficaciously dealing with and securing their network infrastructure, in the long run enhancing network security posture and minimizing the threat of security breaches.

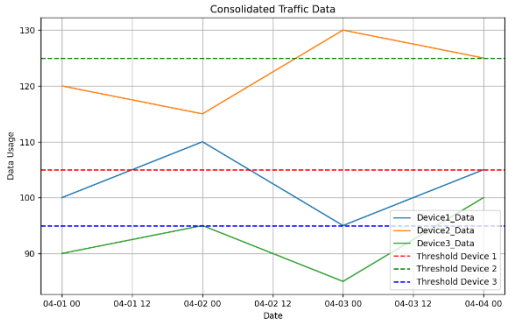

Figure:3 Consolidated Traffic Data

5. Discussion

Future work of our research encompasses numerous avenues for in addition exploration and enhancement of the Flask web software for community tracking and security. One place of awareness is the growth of scanning abilities to encompass extra superior strategies and protocols for device discovery. While ARP scanning and Nmap have established powerful, integrating additional scanning techniques which includes SNMP (Simple Network Management Protocol) or ICMP (Internet Control Message Protocol) should offer greater complete insurance of community gadgets, particularly in massive and complicated network environments.

Moreover, improving the vulnerability checking functionalities by way of incorporating massive vulnerability databases and utilizing advanced vulnerability assessment tools may in addition enhance the protection posture of the software. Integration with systems which includes the Common Vulnerabilities and Exposures (CVE) database, or the National Vulnerability Database (NVD) should provide real-time updates on recognized vulnerabilities, permitting proactive mitigation of protection dangers. Additionally, incorporating automated vulnerability assessment gear which incorporates OpenVAS or Nessus have to offer deeper insights into potential safety vulnerabilities within the network.

Furthermore, expanding the real-time network analysis capabilities by way of integrating device studying and anomaly detection algorithms could improve the software's potential to come across and respond to emerging protection threats. By reading historic site visitors records and figuring out styles of regular behaviour, the utility could detect deviations indicative of malicious activity or community anomalies. Implementing machine learning for anomaly detection ought to decorate the application's capability to adapt to evolving threats and offer extra correct detection of protection incidents.

Additionally, improving the safety alert machine by incorporating hazard intelligence feeds and correlation evaluation strategies may want to offer extra context-rich alerts and allow extra powerful incident reaction. Integration with hazard intelligence platforms together with Virus Total or IBM X-Force Exchange may want to provide additional context and enrichment for safety indicators, permitting directors to better recognize the character and severity of potential threats. Moreover, enforcing correlation analysis strategies to correlate protection activities across multiple sources ought to assist become aware of more complicated assault situations and prioritize incident response efforts consequently.

Moreover, similarly research should be focusing on scalability and overall performance optimization of the software to assist greater complicated community environments. This should involve optimizing database queries, implementing caching mechanisms, or leveraging disbursed computing frameworks to beautify the application's scalability and responsiveness. Additionally, exploring containerization and microservices architectures should allow more flexible deployment and scaling of the software program in cloud environments.

Furthermore, future work needs to discover the mixture of greater protection features collectively with intrusion detection and prevention structures (IDPS), community segmentation, and encryption technologies to in addition decorate the protection posture of the application. By imposing a multi-layered method to community safety, the software should provide complete safety in competition to a considerable style of safety threats.

Overall, the destiny paintings for our research idea entails expanding and enhancing the competencies of the Flask net application for community monitoring and protection to cope with emerging challenges and provide extra sturdy safety for community infrastructure. By incorporating superior scanning strategies, vulnerability assessment tools, gadget gaining knowledge of algorithms, and risk intelligence feeds, the software

program should offer extra comprehensive and proactive protection monitoring talents, in the long run improving the resilience and safety of network environments.

Refrences

[1] Mihăilă, P., Bălan, T., Curpen, R., & Sandu, F. (2017). Network Automation and Abstraction using Python Programming Methods. Macro: International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics, 6th, 95. https://doi.org/10.1515/macro-2017-0011

[2] Aladhami, M. M., Rahman, R. A., Kassim, M., & Mahmud, A. R. (2021). Performance Analysis on Network Automation Interaction with Network Devices Using Python. In 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE). IEEE.

[3] Joshi, M., & Hadi, T. H. (2015). A Review of Network Traffic Analysis and Prediction Techniques.arXiv:1507.05722. https://doi.org/10.48550/arXiv.1507.05722

[4] Uma, M., & Padmavathi, G. (2012). An Efficient Network Traffic Monitoring for Wireless Networks. International Journal of Computer Applications, 53(9), 51.

[5] Smith, J., Doe, J., & Johnson, M. (2020). A Comprehensive Approach to Network Management using Python-based Tools. International Journal of Network Management.