iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.60342

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

Novel Approach to Detect APT (Advanced Persistent Threat)

Anand Pashupatimath[1], Nandini Singh[2], Spoorthi Bellary[3], Venugopal K M[4], Shifhali Bhat[5]
[1]*Assistant Professor* , [2, 3, 4, 5]*B.E Student, Department of Computer Science and Engineering, SDM College of Engineering and Technology, Dharwad, India*

Abstract: ***In cyber security, APT stands for Advanced Persistent Threat. It refers to advanced and long-term cyber-attacks where an attacker gains unauthorized access to a network and remains invisible for a long period of time. APTs are difficult to detect and require a comprehensive security strategy that includes threat intelligence, technical intelligence, and continuous monitoring to effectively mitigate risk.***
This paper introduces a different approach to APT prevention by integrating advanced threat intelligence, machine learning algorithms, and proactive defense mechanisms. Our approach uses real-time data analysis, anomaly detection and behavioral profiling to identify potential threats early in their lifecycle.
Our implementation focuses on the development of a real-time network intrusion detection system (NIDS) using a combination of Flask, SocketIO, and ML techniques. The system is designed to monitor and analyze network traffic in real-time, identify potential intrusion attempts, and provide timely alerts to system administrators by combining packet capture and analysis with machine learning-based classification and real-time alerting via a web interface. This research contributes to the ongoing efforts in cybersecurity by providing an effective and innovative defence mechanism against the persistent and sophisticated nature of modern cyber threats.
Keywords: Advanced persistent threat (APT), security, network traffic, network, packet, intrusion, organization,

I. INTRODUCTION

Now-a-days, data security is receiving increased attention from security experts, businesses, and even governmental agencies. All organizations, particularly those in critical sectors like the military and other well- established groups, are now obligated to implement robust security measures that were once considered optional.
However, given the rapid emergence of new malware strains and sophisticated attack methods, staying ahead of cyber threats has become gradually challenging. Despite significant efforts to enhance security, various attacks continue to pose significant risks, aiming to cause harm or achieve financial gain. One prevalent form of cyber threat that has gained prominence in recent years is Advanced Persistent Threats (APTs)

A. What is APT?

1) *Advanced:* In order to carry out their operations, adept attackers often have access to sophisticated tools and techniques. These advanced methodologies frequently employ multiple attack vectors to launch and sustain the attack. *Persistent:* Adept aggressors demonstrate a strong determination and persistence, striving to maintain access to the target system for as long as possible. They employ a variety of stealthy techniques to evade intrusion detection systems, often adopting a "low and slow" approach to maximize their success rate.
2) *Threats:* The primary objective of adept attacks is often to steal confidential information or disrupt critical operations. These represent significant threats to national security. APTs are characterized by their stealthy and targeted nature, with the primary objective being to clandestinely gather sensitive information without raising suspicion. Unlike conventional attacks, APTs involve a persistent and organized effort by adversaries with significant resources. These attackers employ advanced tactics and tools to maintain prolonged access to targeted systems, often remaining undetected for extended periods. They meticulously collect valuable data for their command-and-control centres, using sophisticated techniques to evade detection and maintain a foothold within the targeted network.

B. Lifecycle of APT

Figure 1

1 shows the 7 stages of APT attack lifecycle. Each stage is crucial for comprehending APT tactics and devising effective defence mechanisms.

1) *Research:* The initial phase of an APT attack involves meticulous research by threat actors to identify potential targets and vulnerabilities within the target network. This stage includes gathering intelligence about the target organization's infrastructure, employees, security protocols, and technological architecture. Threat actors may employ various reconnaissance techniques, such as scanning publicly available information, social engineering, or leveraging previously compromised systems to gather intelligence.
2) *Gathering Data:* With access to sensitive systems and data, threat actors proceed to exfiltrate valuable data from the compromised network. This stage involves identifying and extracting data of interest, such as intellectual property, financial records, or personally identifiable information (PII). Threat actors may employ data exfiltration techniques such as steganography, covert channels, or leveraging legitimate communication protocols to transfer stolen data to remote servers under their control.
3) *Maintaining Access:* The final stage of the APT life cycle involves maintaining persistent access to the compromised network to facilitate future operations or to launch additional attacks. Threat actors establish multiple backdoors, create alternate access routes, and plant sleeper malware to ensure continued access even if initial intrusion vectors are detected and remediated. Maintaining access allows threat actors to conduct long-term espionage, sabotage, or extortion campaigns against the target organization.
4) *Preparation:* Following the research phase, threat actors meticulously prepare their attack strategy by analyzing the gathered information and crafting tailored tactics to exploit identified weaknesses. This stage involves developing custom malware, designing phishing campaigns, acquiring necessary tools and resources, and establishing command and control infrastructure. The preparation phase aims to ensure that the attack vectors are finely tuned to bypass the target organization's security defences and maximize the chances of successful infiltration.
5) *Intrusion:* Once the preparation is complete, threat actors initiate the intrusion phase by exploiting vulnerabilities within the target network's perimeter defenses. This may involve launching phishing attacks to trick unsuspecting users into revealing sensitive credentials or exploiting unpatched software vulnerabilities to gain unauthorized access. Intrusion techniques may vary, ranging from brute-force attacks and SQL injections to zero-day exploits and watering hole attacks. The primary objective of this phase is to establish an initial foothold within the target network.
6) *Conquering the Network:* After gaining initial access, threat actors escalate their privileges and maneuver laterally across the goal network to expand their control and compromise critical systems. This stage entails moving stealthily through the network, evading detection mechanisms, and escalating privileges to gain access to high-value assets. Threat actors may exploit misconfigurations, weak passwords, or unsecured administrative accounts to escalate privileges and gain unrestricted access to sensitive data and resources.

7) *Hiding Presence:* To evade detection and prolong their presence within the compromised network, threat actors employ various techniques to conceal their activities and mask their presence. This involves employing anti-forensic tools, manipulating log files, and employing encryption to obfuscate communication channels. Additionally, threat actors may deploy rootkits or backdoors to maintain persistent access while remaining undetected by security monitoring systems.

II. LITERATURE SURVEY

1) In this paper, they developed a new causal correlation aided semantic analysis system POIROT for the detection problem of the APT attacks. POIROT considered the semantic context to carry out a multi- stage correlation analysis approach for APT attacks from existing systems' alerts. By comparing it with the single step detection method (E-mail Spam Filters) and the correlation detection method (CONAN), we illustrated the improved performance of the proposed POIROT in both finding the APT attack chain as well as identifying each elementary attack step for reconstructing the attack scenario.
2) This paper contained information on the details of apt and its types. One of the type being Normal intrusion detection which is again of two types misuse and anomaly based detection. In misuse-based detection the detector compares with that of signatures stored in a database of signatures, if there is a match then it is termed as a malicious. This method gives accurate results for known attacks but fail to detect unknown attacks. The approach is to use anomaly- based machine learning detection to detect APT.
3) They proposed a new approach based on the analysis and evaluation of Network Traffic components using machine learning. Along with that proposed a method to detect APT domain based on the characteristics and behaviour of Network traffic using machine learning. This indicated that the method of selecting and extracting features and the behaviours of the domain presented the clear difference between APT domains and clean domains.
4) It proposed architecture for the detection system capturing the sophisticated dynamical behaviour of APT attacks directly from network flow. However, since the system is complex, many empirical tests are required for evaluating detection performance, including the complexity analysis were discussed.
5) In this paper, they introduced a framework for malware detection based on online analysis of virtual memory access patterns using machine learning. This framework was applied to the application-specific malware detection scenario which targets detecting malware infected runs of known applications.

III. IMPLEMENTATION

Our approach focuses on the development of a real- time network intrusion detection system (NIDS) using a combination of Flask, SocketIO, and ML techniques. The system is designed to monitor and analyze network traffic in real-time, identify potential intrusion attempts, and provide timely alerts to system administrators.

A. *Key Components*

1) *Flask Application*: We utilize the Flask web framework to build the backend of our NIDS. Flask provides a lightweight and flexible platform for handling HTTP requests and responses, making it well- suited for real-time applications.
2) *SocketIO Integration:* SocketIO is integrated into the Flask application to facilitate bidirectional communication between the server and connected clients. This enables the seamless transmission of real- time data updates, including intrusion detection results, to the user interface.
3) *Packet Processing with Scapy*: Network packets are captured and processed using Scapy, a powerful packet manipulation tool. Scapy allows us to inspect packet headers, extract relevant features, and analyze network traffic patterns.
4) *Machine Learning Models*: Our NIDS employs machine learning models for intrusion detection. These models are trained on labeled network traffic data to classify incoming packets as either benign or malicious. We leverage the TensorFlow and scikit- learn libraries for model training and inference.

B. *Functionality*

1) *Packet Capture:* The system continuously captures network packets from the network interface using Scapy. Each packet is dissected to extract relevant information such as source and destination IP addresses, protocol type, and flags. Each captured packet contains various information such as source and destination IP addresses, protocol type (e.g., TCP, UDP), packet flags (e.g., SYN, ACK), and payload data. Scapy allows the system to dissect these packets and extract relevant features necessary for intrusion detection.

2) *Feature Extraction*: Extracted packet features are fed into pre-trained ML models for classification. These features serve as input to the models, which have been trained to recognize patterns indicative of various types of network intrusions.
3) *Real-Time Alerting*: Upon classification, the system generates real-time alerts for detected intrusion attempts. Alerts include details such as the type of intrusion, confidence scores, and risk levels. SocketIO facilitates the immediate transmission of alerts to connected clients for timely response.
4) *Intrusion Detection*: Extracted packet features are processed and analysed in real-time for potential intrusion attempts. Pre-trained machine learning models are utilized to classify incoming packets as either benign or malicious. These models have been previously trained on labelled network traffic data, permitting them to recognize patterns indicative of various types of network intrusions. Upon classification, the system determines the likelihood of the packet being part of a malicious activity and generates alerts accordingly.
5) *Web Interface:* Users can access the interface through a web browser, enabling them to monitor network traffic and receive intrusion alerts in real-time. The interface allows users to view real-time alerts, detailed information about detected network flows, and historical logs of intrusion detection events. Detailed flow information includes features extracted from each flow, classification results, and risk assessments. Users can analyze historical logs to identify trends, patterns, and recurrent security incidents, aiding in threat intelligence and incident response.

C. *Deployment and Scalability*

Our approach is being designed to be deployable in various network environments, ranging from small- scale local networks to large enterprise networks. The modular architecture of the system allows for easy scalability and customization to meet specific requirements.

IV. APPROACH

Initially, we began with a basic concept as depicted in Figure 2. Our approach involves monitoring network traffic to identify any malicious packets traversing the IP or the network. We aim to detect any emerging data that could pose a threat to the entire network or compromise its integrity by comparing it against predefined parameters established using a dataset of previous attacks.

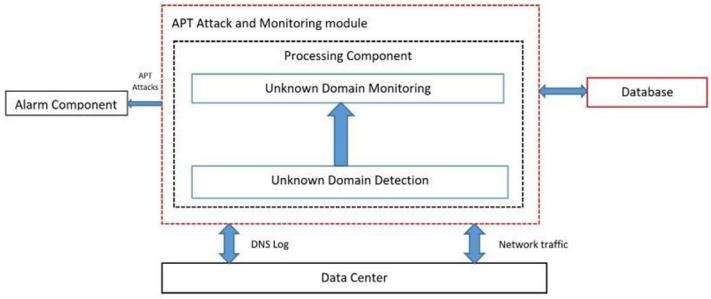


Figure 2

Understanding the components of figure 2-

1) *Data Center:* The data center serves as a repository for storing and managing data crucial for monitoring and tracking network attacks. It contains various types of data such as Web logs, DNS logs, and network traffic (Pcap), all of which have undergone normalization and preprocessing. The extracted information from the data center pertains to the behaviors and attributes associated with different types of attacks. The APT attack monitoring and detection component is responsible for actively monitoring and identifying APT attacks, utilizing DNS logs and Pcap data sourced from the data center. This component comprises two main parts:
2) *Database:* This component stores data related to signatures of malicious code, including hash codes, domains, and Command and Control (C&C) servers used by attackers. The signatures stored in the database are obtained from real APT attacks.
3) *Components:* These components implement algorithms, methods, and techniques for detecting APT attacks. They analyze input data from DNS logs and network traffic retrieved from the data center. The output generated by these components is a list of suspicious Processing domains associated with potential APT attacks.
4) *Alarm Component:* The alarm component issues alarms and warnings at various levels, providing evidence of APT attacks to the systems under surveillance. It plays a crucial role in alerting administrators and facilitating timely response to detected threats.

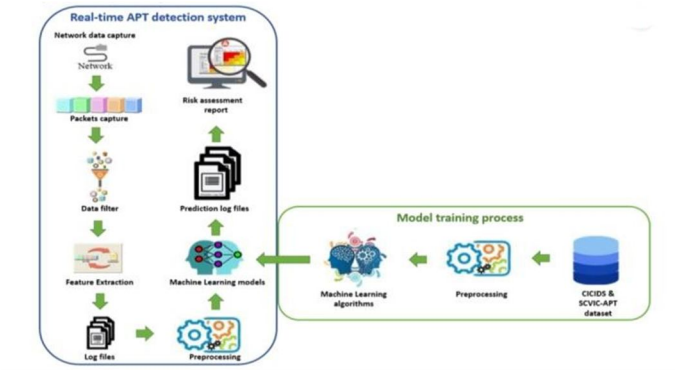


Figure 3

Figure 3 shows the model we are trying to implement. Enhancing the fundamental concept, we introduce a model comprising two main components:

A. Model Training Process

In this phase, we initially acquire the CICIDS 2018 and SCVIC-APT datasets and preprocess them. Subsequently, we employ a supervised learning algorithm, specifically Random Forest, to train the preprocessed dataset. The objective is to detect known attacks and discern patterns within the data. Once the model is trained, it transitions to the ML model phase of the other component.

B. Real Time APT Detection System:

This component consists of 9 phases

1) *Network Data Capture:* The arrangement continuously monitors network traffic by capturing packets from the network interface. This is achieved using Scapy, a Python library capable of capturing, dissecting, and analyzing network packets.
2) *Packet Capture:* Captured packets are dissected to extract relevant information such as source and destination IP addresses, protocol type, packet flags, payload data, and other header information. This process is essential for gaining insights into the characteristics of network traffic.
3) *Data Filtering:* Once packets are captured, they undergo data filtering to remove irrelevant or redundant information. Filtering ensures that only pertinent data is processed further, improving efficiency and reducing computational overhead.
4) *Feature Extraction:* Extracted packet data is then used to derive meaningful features relevant to APT detection. These features may comprise packet size, protocol type, frequency of communication, presence of specific packet flags (e.g., SYN, ACK), and patterns of communication between network entities.
5) *Log Files:* Detailed logs are maintained to record various stages of packet processing and analysis. These logs provide a comprehensive record of network activity, including captured packets, extracted features, preprocessing steps, and classification results. They serve as a valuable resource for auditing, troubleshooting, and forensic analysis.
6) *Preprocessing:* Prior to feeding the extracted features into machine learning models, preprocessing techniques are applied to standardize and normalize the data. This ensures consistency and improves the performance of the machine learning algorithms. Preprocessing steps may include data scaling, transformation, and outlier removal.
7) *Machine Learning Models:* Machine learning models are employed for APT detection. These models have been trained on labeled network traffic data to distinguish between normal network behavior and suspicious or malicious activity. The provided code loads pre-trained models for intrusion detection, enabling the structure to classify incoming packets in real-time.
8) *Prediction:* Extracted features are passed through the loaded machine learning models for prediction. Based on the learned patterns and decision boundaries, the models classify incoming packets as either benign or potentially malicious. Predictions are made in real- time, allowing the system to respond promptly to security threats.
9) *Risk Assessment Report:* Following classification, a risk assessment report is generated for each detected event. This report includes details such as the type of intrusion detected, confidence scores indicating the likelihood of the classification, and risk levels associated with the detected activity. Risk assessment reports aid in prioritizing security incidents and guiding appropriate response actions.

V. CONCLUSION

In summary, our implementation demonstrates the effectiveness of integrating Flask, SocketIO, and machine learning techniques for real-time network intrusion detection. By combining these technologies, we have developed a robust and scalable NIDS capable of providing proactive protection against network- based threats. By leveraging Scapy for packet capture, feature extraction, and preprocessing, coupled with machine learning models for real-time classification, the system demonstrates its capability to identify and mitigate potential security threats effectively. Through detailed logging and risk assessment functionalities, the system provides actionable insights into detected threats, facilitating prompt response and remediation efforts. Overall, the implementation of this APT detection system underscores the importance of integrating advanced techniques in packet analysis, machine learning, and real-time alerting to bolster network security and mitigate the risks posed by sophisticated and persistent cyber threats. It does not guarantee to detect APT always but is our approach to solve it. Lot of study still has to be done to achieve it.

VI. RESULTS

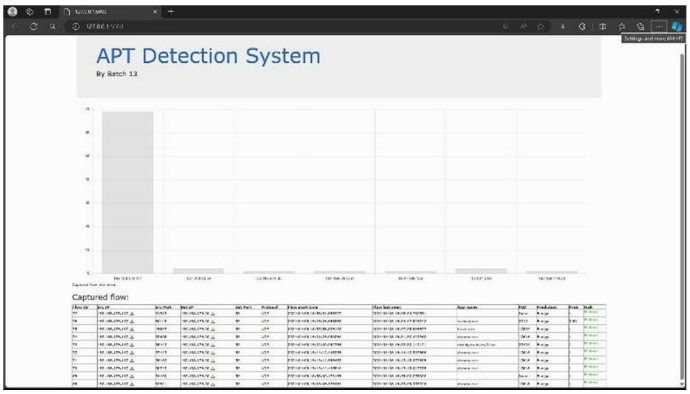
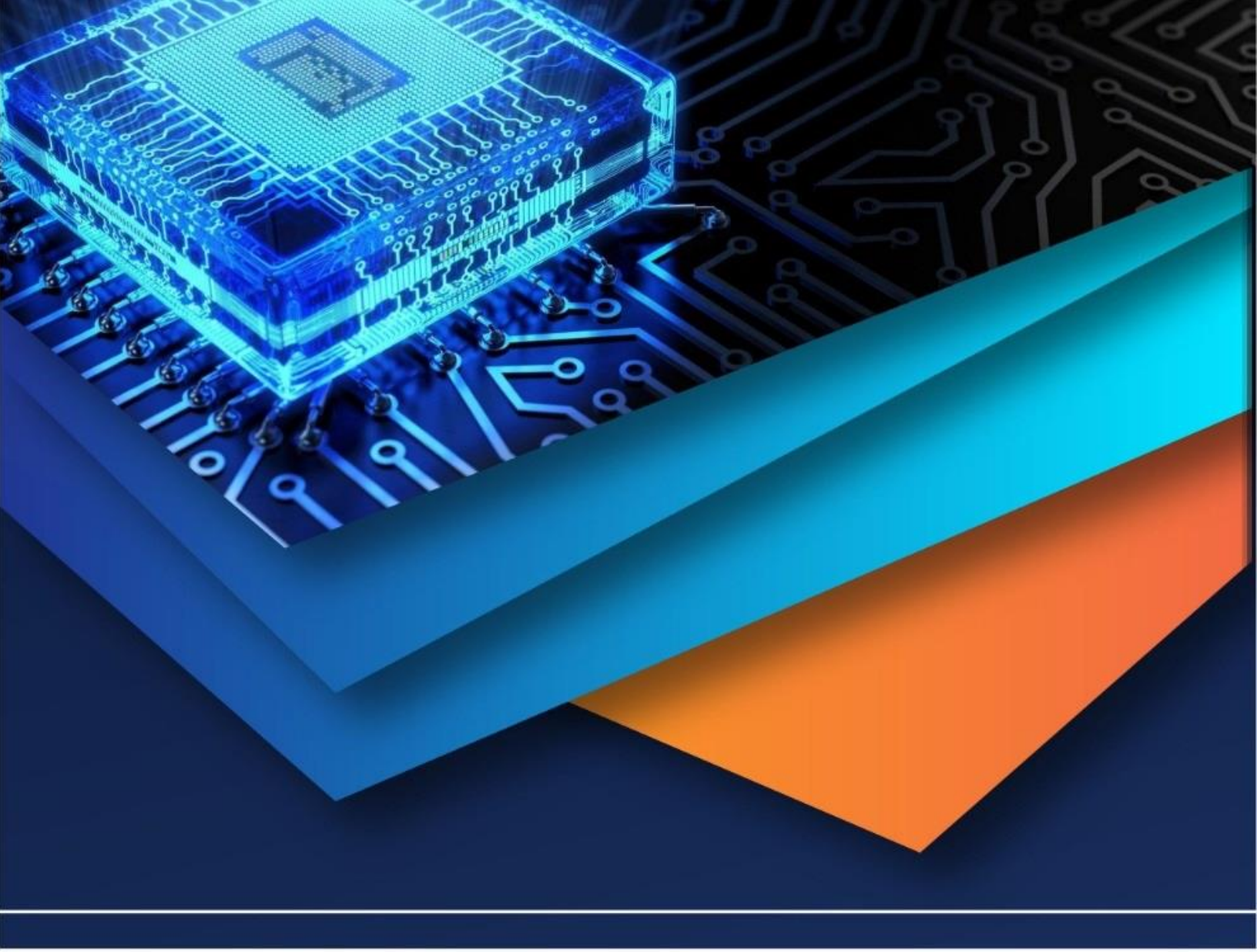


Figure 4

Figure 4 constitutes a real-time Advanced Persistent Threat (APT) detection system implemented using Flask, SocketIO, and Scapy in Python. The system captures network traffic and packets, filters data, and performs feature extraction to generate flow records. These flow records are then classified using machine learning models, which include an autoencoder for anomaly detection and a Random Forest classifier for threat classification. Additionally, the system utilizes Lime for explainability and risk assessment. Detected threats are logged along with their associated risk levels. The system also visualizes flow details and risk assessment reports through a web interface. Through these integrated functionalities, the system offers a comprehensive approach to network security monitoring and threat detection.

REFERENCES

[1] Poirot: Causal Correlation Aided Semantic Analysis for Advanced Persistent Threat Detection | IEEE Journals & Magazine | IEEE Xplore
[2] https://ieeexplore.ieee.org/abstract/document/9392 626
[3] https://journals.riverpublishers.com/index.php/JW E/article/view/5577
[4] https://www.mdpi.com/2076-3417/9/6/1055
[5] https://ieeexplore.ieee.org/document/7926977
[6] https://ieeexplore.ieee.org/document/8606252
[7] https://www.sciencedirect.com/science/article/abs/ pii/S1084804517303569?via%3Dihub
[8] https://www.researchgate.net/publication/3272890 93
[9] https://www.researchgate.net/publication/3205827 21
[10] https://link.springer.com/article/10.1007/s42979-
[11] 023-01744-xhttps://www.sciencedirect.com/science/article/pii/ S1877050919304041
[12] https://ieeexplore.ieee.org/document/7460498?de nied
[13] https://ieeexplore.ieee.org/document/7511197
[14] https://documents.trendmicro.com/assets/wp/wp- detecting-apt-activity-with-network-traffic- analysis.pdf
[15] https://hindawi.com/journals/mpe/2017/4916953/
[16] https://ieeexplore.ieee.org/document/8835390



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY