Security Posture Evaluation and Threat Intelligence Analysis using Python

Thilagavathy S
Computer Science and Engineering
Saveetha Engineering College
(Anna University)
Chennai, India
thilagavathy@saveetha.ac.in

Bharathi Priyan T
Computer Science and Engineering
Saveetha Engineering College
(Anna University)
Chennai, India
tbharathipriyan@gmail.com

Akash A
Computer Science and Engineering
Saveetha Engineering College
(Anna University)
Chennai, India
akasharul2407@gmail.com

Dhinesh Kumar T
Computer Science and Engineering
Saveetha Engineering College
(Anna University)
Chennai, India
dhineshkumardhineshkumar@gmail.com

***Abstract*— In today’s digital landscape, safeguarding sensitive information from unauthorized access and data breaches is critical. As cyber threats become increasingly sophisticated, organizations face heightened risks, making effective monitoring and analysis of system behavior essential. This project aims to develop a comprehensive solution for detecting and analyzing suspicious files and processes within computer systems, specifically targeting potential data transmissions to unauthorized entities.**

Furthermore, the project will integrate robust analysis tools to evaluate network traffic for unauthorized data transmissions, ensuring comprehensive oversight. By incorporating external threat intelligence, the solution will remain vigilant against emerging vulnerabilities and evolving attack vectors. Beyond detection and analysis, this initiative aims to empower users with intuitive tools and insights that enhance their understanding of data security. By fostering a proactive security culture, individuals will be better equipped to maintain the integrity and confidentiality of their sensitive information. Ultimately, this project seeks to significantly mitigate the risks associated with data breaches and bolster organizational resilience in an increasingly complex cyber threat landscape. Through rigorous analysis and proactive detection capabilities, the proposed solution will play a vital role in protecting critical data against evolving cyber threats*Keywords*— *Crash, Crash Response Time, Accident, and Fuzzy Logic.*

I. Introduction

In today’s digital landscape, the safeguarding of sensitive information is of utmost importance as individuals and organizations face an ever-increasing array of cyber threats. With the rise of sophisticated hacking techniques, traditional security measures often prove inadequate in preventing unauthorized access and data breaches.

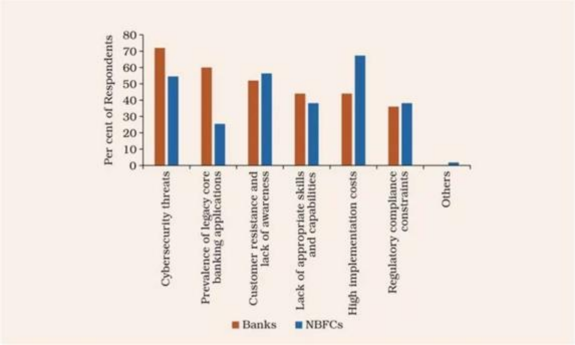
These incidents can have severe repercussions, including financial loss and damage to reputation, highlighting the urgent need for enhanced security protocols.

This project aims to address these challenges by developing a comprehensive solution for detecting and analyzing suspicious files and processes on computer systems. By employing advanced behavioral analysis and anomaly detection techniques, the solution will identify potential data transmissions to unauthorized entities. Ultimately, this initiative seeks to empower users and organizations to better protect their sensitive information and strengthen their defenses against the evolving landscape of cyber threats.

Reducing the time between an incident and the moment first responders, such as medical personnel, arrive and take appropriate action can greatly lessen the severity of the accident. To detect the existence of an accident, the Accident Emergency Alert System makes use of object tracking in a camera.

Figure 1: India's Driving too fast [1]

Overall, 83% of organizations in India reported at least one cybersecurity incident in the last year, placing the country at No. 4, behind Vietnam (94%), New Zealand (90%), and Hong Kong (86%) in rankings for the Asia-Pacific region, according to a Cloudflare report.



A variety of rapidly digitized critical infrastructure sectors in India — from finance to government systems and from manufacturing to healthcare — now are facing increased cyberattacks and cyber threats.

Consider this: A hacking group in April of this year leaked 7.5 million records containing personal information stolen from India's leading manufacturer of wireless audio and wearable devices boat. Most recently, the Reserve Bank of India — the nation's central bank — called out increased digitization as a potential risk for the country's financial infrastructure. Cyber incidents against finance and handled by the national CERT team jumped to some 16 million incidents in 2023, up from 53,000 in 2017, according to a recent report by RBI

II. LITRETURE SURVEY

2.1 This publication by NIST is a comprehensive guide to securing Industrial Control Systems (ICS), which are used in critical infrastructure such as power grids, water supplies, and transportation systems. The authors emphasize the unique security challenges posed by ICS, which often have a different set of priorities (such as reliability and availability) compared to traditional IT systems. The guide offers detailed best practices, including the use of layered defenses (defense-in-depth), network segmentation, monitoring for abnormal behavior, and secure configurations of both hardware and software. It provides a solid foundation for organizations to understand and mitigate the cyber risks associated with ICS environments, which are increasingly targeted by sophisticated cyber-attacks. [1].

2.2 This paper offers an extensive review of the role of Cyber Threat Intelligence (CTI) in cybersecurity, with a particular focus on the integration of machine learning for enhanced threat detection and prevention. The authors discuss various sources of CTI, such as open-source intelligence (OSINT), dark web data, and proprietary intelligence feeds, and how they contribute to a comprehensive understanding of emerging threats. The study highlights the challenges in processing and analyzing large volumes of threat data, and how machine learning techniques like clustering, classification, and anomaly detection can automate this process, allowing security teams to focus on more strategic tasks. By leveraging machine learning, organizations can predict and proactively respond to cyber threats before they cause harm. [2].

2.3 Sommer and Paxson examine the potential and limitations of using machine learning for network intrusion detection, a critical component of cybersecurity defense. While machine learning offers promising capabilities for detecting novel threats, the authors point out several obstacles to its real-world application These include the high false-positive rates that can overwhelm security teams, the difficulty of obtaining labeled training data, and the evolving nature of threats that make it hard to maintain an up-to-date model. The paper calls for a careful consideration of the trade-offs between automation and human oversight in intrusion detection systems and suggests a hybrid approach combining machine learning with rule-based detection to improve accuracy. [3].

2.4 This research presents the development of an intelligent CTI system that utilizes machine learning algorithms to automate the identification and analysis of cyber threats. By applying techniques such as natural language processing (NLP) for analyzing threat reports and clustering algorithms for grouping similar threats, the system enhances the ability of security teams to quickly respond to emerging threats. The intelligent system is designed to continuously learn from new data, improving its accuracy over time and reducing the burden of manual threat analysis. The study demonstrates how machine learning can streamline the CTI process, making it more efficient and scalable for large organizations with complex security needs. [4].

2.5 This study focuses on the use of Python, a versatile programming language, to automate various aspects of CTI. Python's extensive libraries, such as Scapy for network packet analysis and PyTorch for implementing machine learning models, make it an ideal tool for automating threat detection and analysis. The authors showcase how Python scripts can be used to pull data from multiple CTI sources, such as threat feeds and web scraping, and process it in real time. Automation reduces the need for manual data collection and allows for faster, more accurate identification of threats. By using Python, organizations can create flexible, customized CTI solutions tailored to their specific cybersecurity needs. [5].

2.6 This paper describes the creation of a sophisticated CTI framework that combines the capabilities of Python for automation and machine learning for advanced threat analysis. The framework is designed to analyze large datasets of threat intelligence, identifying patterns and anomalies that may indicate potential cyber attacks. Machine learning models are used to classify threats based on severity, enabling security teams to prioritize their response efforts. The framework also incorporates real-time monitoring and alerting features, ensuring that emerging threats are detected and addressed promptly. The study highlights the effectiveness of combining Python's flexibility with machine learning's predictive power to enhance cybersecurity defenses. [6].

III. METHODOLOGY

A. Problem Statement

Traditional methods of detecting suspicious files and processes on computer systems rely on signature-based detection techniques. These methods are prone to several limitations False Positives, Inability to Detect New Threats, No Real-Time Threat Intelligence Integration

Drawbacks of System

- High false-positive rates due to a lack of behavioral analysis.
- Inability to detect unauthorized network transmissions in real time.
- Limited adaptability to unknown threat patterns without external intelligence sources.

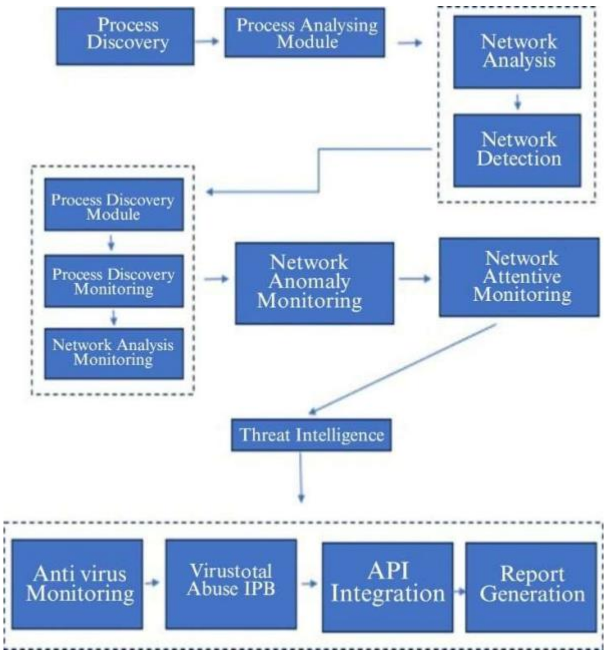
B. Proposed System

We propose integrating deep learning with behavioral and anomaly detection to identify suspicious activities in real-time. A CNN model will be trained on labeled datasets of malicious and benign activities, then integrated into a Django framework to monitor file processes and network traffic. Based on predictions, the system will trigger alerts and send automated notifications to administrators for investigation. Additionally, it will connect to external threat intelligence platforms to stay updated with the latest threat patterns, ensuring effective threat detection.

Benefits of the system:

- Real-Time Detection: Constant monitoring of file and process behavior to identify anomalies in real time.
- Proactive Alerts: Automatic notifications and emails for administrative action, ensuring quick responses to potential threats.
- Adaptable and Scalable: The system can be updated with new datasets and threat intelligence to handle emerging threats.
- Network Traffic Monitoring: Detect unauthorized data transmissions and flag potential data breaches.
- Efficient and Easy-to-Use: Simplified interface for administrators to handle alerts and take necessary actions.

IV. ARCHITECTURE



System Modules:

Module 1: Setting Up the Environment
Module 2: Obtaining API Keys
Module 3: Running the Scripts
Module 4: Interpreting the Output
Module 5: Analysis and Results
Module 6: Interpretation of Network Activity

Module 1: Setting Up the Environment:

This foundational module sets the stage for the entire project by guiding users through the essential steps needed to create a functional environment. Initially, it ensures that Python is installed on the system, which is crucial for executing the scripts. Users are instructed to install necessary libraries that are vital for the project’s functionality. Key libraries include **psutil**, which provides an interface for retrieving comprehensive information on system utilization, such as CPU usage, memory consumption, disk activity, and network statistics. The **requests** library is introduced next, facilitating seamless HTTP requests that allow interaction with external APIs like VirusTotal and AbuseIPDB for threat intelligence. The **wmi** library enables users to query and interact with Windows Management Instrumentation, making it possible to retrieve detailed system and process information specifically on Windows platforms. Lastly, the **pandas** library is highlighted for its powerful data manipulation and analysis capabilities, which will be invaluable as the project expands to handle larger datasets or generate reports on the findings.

Module 2: Pre-processing of Data :

users are guided through the critical process of registering on the VirusTotal and AbuseIPDB websites to obtain API keys. These keys are essential for making authorized queries to the respective APIs, allowing users to check if specific IP addresses or files are flagged as malicious. The module emphasizes the importance of these keys in enhancing the project’s capabilities by enabling real-time threat assessments. By securing these API keys, users position themselves to effectively leverage external threat intelligence services

Module 3: Running the Script:

This module covers the practical implementation of the project by instructing users on how to configure the API keys within the `threat_intelligence.py` file. Users replace placeholder text with the actual API keys obtained in the previous module. Once the configuration is complete, users execute the `main.py` script to initiate the monitoring process. This script is designed to perform a comprehensive collection of information about currently running processes on the system, gather associated network details, and assess these details against the external APIs for an in-depth threat evaluation. This step is crucial for establishing a baseline of system activity and for identifying any potential security threats that may arise.

Module 4: Interpreting the Output

Upon successful execution of the scripts, the output will present users with detailed **Network Details** concerning each suspected process. This includes vital information such as source IP addresses (from which data is being sent) and destination IP addresses (to which data is being sent). Furthermore, the **Threat Intelligence Results** section will display information retrieved from VirusTotal and AbuseIPDB, indicating whether any of the destination IPs are known to be associated with malicious activities. This module highlights the importance of analyzing the output data, as it forms the basis for informed decision-making regarding system security and risk management.

Module 5 Analysis and Results

Each destination IP is subjected to verification against VirusTotal and AbuseIPDB to assess its status regarding malicious or abusive behavior. This thorough analysis is key for validating the integrity of the system and ensuring that any potential threats are identified and addressed promptly.

Module 6: Concluding Forecast

This module delves into the intricacies of analyzing network connections for unusual patterns and anomalies. The analysis involves looking for red flags, such as connections to unfamiliar external IP addresses or unexpectedly high volumes of data transfers, which could signify suspicious or unauthorized activity. By cross-referencing the identified destination IPs with threat intelligence services, the script effectively identifies whether observed network activities are potentially harmful. This interpretative analysis plays a critical role in enhancing the security posture of the system, allowing for proactive measures to mitigate risks.

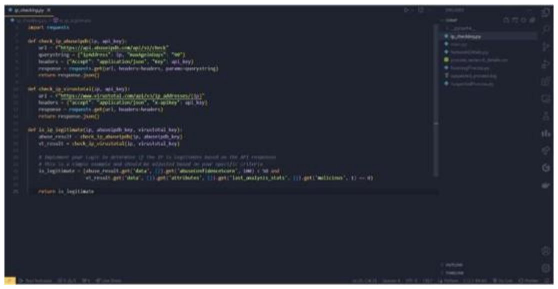
Module 6: Identified Suspicious Processes and Files

The final module culminates in the generation of a comprehensive report that details processes exhibiting suspicious network behavior. This report includes a summary of findings, listing process IDs (PIDs), names, network connection details, and results from VirusTotal and AbuseIPDB. Additionally, it provides insights into the threat level associated with each identified IP address, delivering valuable context that aids users in determining whether further action is warranted.

RESULTS

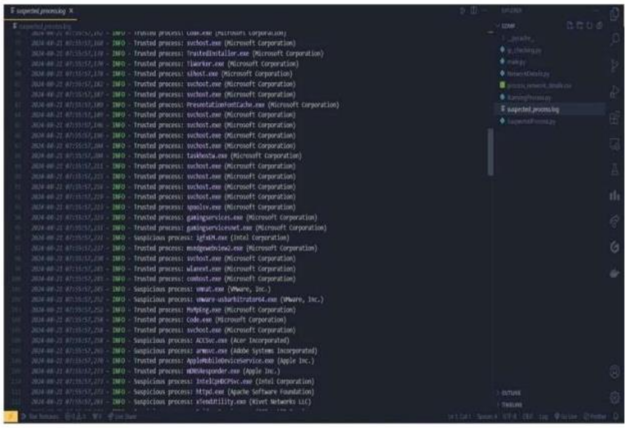
the evaluation results indicate that the trained CNN model exhibits strong performance metrics, making it a valuable tool for the detection and analysis of potential security threats. The combination of an 80% validation accuracy and a 92% test precision not only underscores the model's capability but also provides a solid foundation for future enhancements, such as real-time threat detection and automated response mechanisms.

Figure 3: Checking the threads running on the IP address[4]



```
import requests

def check_ip_abuseipdb(ip, api_key):
    url = f"https://api.abuseipdb.com/api/v2/check"
    querystring = {"ipAddress": ip, "maxAgeInDays": "90"}
    headers = {"Accept": "application/json", "Key": api_key}
    response = requests.get(url, headers=headers, params=querystring)
    return response.json()

def check_ip_virustotal(ip, api_key):
    url = f"https://www.virustotal.com/api/v3/ip_addresses/{ip}"
    headers = {"accept": "application/json", "x-apikey": api_key}
    response = requests.get(url, headers=headers)
    return response.json()

def is_ip_legitimate(ip, abuseipdb_key, virustotal_key):
    abuse_result = check_ip_abuseipdb(ip, abuseipdb_key)
    vt_result = check_ip_virustotal(ip, virustotal_key)

    # Implement your logic to determine if the IP is legitimate based on the API responses
    # This is a simple example and should be adjusted based on your specific criteria
    is_legitimate = (abuse_result.get('data', {}).get('abuseConfidenceScore', 100) < 50 and
                     vt_result.get('data', {}).get('attributes', {}).get('last_analysis_stats', {}).get('malicious', 1) == 0)

    return is_legitimate
```

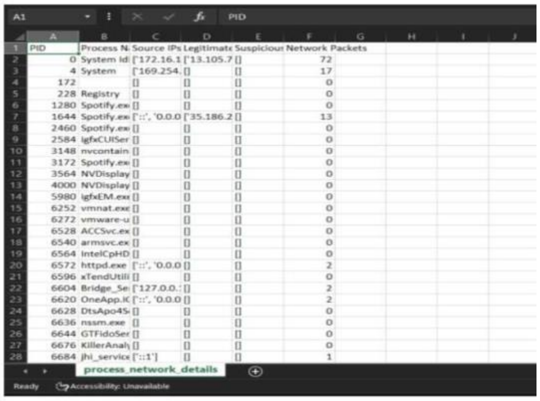
Figure 4 : Deep learning system that uses accident detection to trigger requests to hospital[5].



```
2024-08-21 07:55:57,162 - INFO - Trusted process: Code.exe (Microsoft Corporation)
2024-08-21 07:55:57,168 - INFO - Trusted process: svchost.exe (Microsoft Corporation)
2024-08-21 07:55:57,170 - INFO - Trusted process: TrustedInstaller.exe (Microsoft Corporation)
2024-08-21 07:55:57,176 - INFO - Trusted process: TiWorker.exe (Microsoft Corporation)
2024-08-21 07:55:57,178 - INFO - Trusted process: sihost.exe (Microsoft Corporation)
2024-08-21 07:55:57,182 - INFO - Trusted process: svchost.exe (Microsoft Corporation)
2024-08-21 07:55:57,187 - INFO - Trusted process: svchost.exe (Microsoft Corporation)
2024-08-21 07:55:57,189 - INFO - Trusted process: PresentationFontCache.exe (Microsoft Corporation)
2024-08-21 07:55:57,189 - INFO - Trusted process: svchost.exe (Microsoft Corporation)
2024-08-21 07:55:57,196 - INFO - Trusted process: svchost.exe (Microsoft Corporation)
2024-08-21 07:55:57,198 - INFO - Trusted process: svchost.exe (Microsoft Corporation)
2024-08-21 07:55:57,204 - INFO - Trusted process: svchost.exe (Microsoft Corporation)
2024-08-21 07:55:57,208 - INFO - Trusted process: taskhostw.exe (Microsoft Corporation)
2024-08-21 07:55:57,211 - INFO - Trusted process: svchost.exe (Microsoft Corporation)
2024-08-21 07:55:57,215 - INFO - Trusted process: svchost.exe (Microsoft Corporation)
2024-08-21 07:55:57,216 - INFO - Trusted process: svchost.exe (Microsoft Corporation)
2024-08-21 07:55:57,219 - INFO - Trusted process: svchost.exe (Microsoft Corporation)
2024-08-21 07:55:57,223 - INFO - Trusted process: spoolsv.exe (Microsoft Corporation)
2024-08-21 07:55:57,229 - INFO - Trusted process: gamingservices.exe (Microsoft Corporation)
2024-08-21 07:55:57,231 - INFO - Trusted process: gamingservicesnet.exe (Microsoft Corporation)
2024-08-21 07:55:57,232 - INFO - Suspicious process: igfxEM.exe (Intel Corporation)
2024-08-21 07:55:57,237 - INFO - Trusted process: msedgewebview2.exe (Microsoft Corporation)
2024-08-21 07:55:57,238 - INFO - Trusted process: svchost.exe (Microsoft Corporation)
2024-08-21 07:55:57,243 - INFO - Trusted process: wlanext.exe (Microsoft Corporation)
2024-08-21 07:55:57,245 - INFO - Trusted process: conhost.exe (Microsoft Corporation)
2024-08-21 07:55:57,245 - INFO - Suspicious process: vmnat.exe (VMware, Inc.)
2024-08-21 07:55:57,252 - INFO - Suspicious process: vmware-usbarbitrator64.exe (VMware, Inc.)
2024-08-21 07:55:57,252 - INFO - Trusted process: MsMpEng.exe (Microsoft Corporation)
2024-08-21 07:55:57,254 - INFO - Trusted process: Code.exe (Microsoft Corporation)
2024-08-21 07:55:57,258 - INFO - Trusted process: svchost.exe (Microsoft Corporation)
2024-08-21 07:55:57,258 - INFO - Suspicious process: ACCSvc.exe (Acer Incorporated)
2024-08-21 07:55:57,265 - INFO - Suspicious process: armsvc.exe (Adobe Systems Incorporated)
2024-08-21 07:55:57,270 - INFO - Trusted process: AppleMobileDeviceService.exe (Apple Inc.)
2024-08-21 07:55:57,273 - INFO - Trusted process: mDNSResponder.exe (Apple Inc.)
2024-08-21 07:55:57,277 - INFO - Suspicious process: IntelCpHDCPSvc.exe (Intel Corporation)
2024-08-21 07:55:57,282 - INFO - Suspicious process: httpd.exe (Apache Software Foundation)
2024-08-21 07:55:57,283 - INFO - Suspicious process: xTendUtility.exe (Rivet Networks LLC)
```

Figure 5 : Process that runs on your device with the help of API [6].

CSV:



	A	B	C	D	E	F	G	H	I	J
1	PID	Process N	Source IPs	Legitimate	Suspiciou	Network Packets				
2	0	System id	['172.16.1	['13.105.7	[]	72				
3	4	System	['169.254.	[]	[]	17				
4	172		[]	[]	[]	0				
5	228	Registry	[]	[]	[]	0				
6	1280	Spotify.ex	[]	[]	[]	0				
7	1644	Spotify.ex	['::', '0.0.0	['35.186.2	[]	13				
8	2460	Spotify.ex	[]	[]	[]	0				
9	2584	igfxCUISer	[]	[]	[]	0				
10	3148	nvcontain	[]	[]	[]	0				
11	3172	Spotify.ex	[]	[]	[]	0				
12	3564	NVDisplay	[]	[]	[]	0				
13	4000	NVDisplay	[]	[]	[]	0				
14	5980	igfxEM.exe	[]	[]	[]	0				
15	6252	vmnat.exe	[]	[]	[]	0				
16	6272	vmware-u	[]	[]	[]	0				
17	6528	ACCSvc.ex	[]	[]	[]	0				
18	6540	armsvc.ex	[]	[]	[]	0				
19	6564	IntelCpHD	[]	[]	[]	0				
20	6572	httpd.exe	['::', '0.0.0	[]	[]	2				
21	6596	xTendUtili	[]	[]	[]	0				
22	6604	Bridge_Se	['127.0.0.:	[]	[]	2				
23	6620	OneApp.IC	['::', '0.0.0	[]	[]	2				
24	6628	DtsApo4S	[]	[]	[]	0				
25	6636	nssm.exe	[]	[]	[]	0				
26	6644	GTFidoSer	[]	[]	[]	0				
27	6676	KillerAnaly	[]	[]	[]	0				
28	6684	jhi_service	['::1']	[]	[]	1				

Fig 6: Final CSV What the Ongoing Tasks [7].

I. CONCLUSION

The implementation of a real-time accident detection system leveraging advanced computer vision and deep learning technologies is crucial for the accurate and timely identification of traffic accidents as they occur. This capability is essential for initiating swift emergency responses, mitigating the impact of accidents, and ultimately enhancing road safety. The system features an automated notification mechanism that ensures emergency services receive immediate alerts, complete with detailed information about the accident's severity and location. Such responsiveness is vital for reducing response times, enabling quicker medical intervention, and potentially saving lives. Furthermore, the project's emphasis on integrating the accident detection system with existing emergency response frameworks highlights the importance of achieving high accuracy in detection and ensuring seamless communication between systems. By improving the speed and efficiency of accident detection and response, this innovative approach has the potential to significantly enhance road safety, resulting in better outcomes for accident victims and contributing to a reduction in fatalities and severe injuries.

II. FUTURE ENRICHMENT

the project aims to incorporate comprehensive system monitoring by expanding its coverage to include additional aspects of system security, such as file system changes, registry modifications, and other indicators of compromise. This enhancement will provide a more holistic view of system integrity and potential threats. Additionally, the introduction of automated responses to detected threats will be a key focus. This could involve implementing actions like blocking malicious IP addresses, terminating suspicious processes, or quarantining affected files to neutralize threats swiftly and effectively.

III. REFERENCES

[1] “Stouffer, K., Falco, J., & Scarfone, K. (2011). Guide to Industrial Control Systems (ICS) Security. National Institute of Standards and Technology (NIST).”

[2] "Feng, H., Qiu, M., Hu, X., & Khan, I. (2020). A Comprehensive Study on Cyber Threat Intelligence in Cybersecurity. IEEE Access, 8, 29839-29850.

[3] "Sommer, R., & Paxson, V. (2010). Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. IEEE Symposium on Security and Privacy

[4] "An Intelligent Cyber Threat Intelligence System Using Machine Learning Techniques

[5] "Automated Cyber Threat Intelligence: An Efficient Approach Using Python

[6] "An Advanced Threat Intelligence Framework Using Python and Machine Learning

[7] "Improving Cybersecurity Posture Using Threat Intelligence Platforms.

[8] Cybersecurity Risk Assessment using Threat Intelligence and Machine Learning

[9] Using Python for Real-Time Cyber Threat Intelligence AnalysisJ. Zhao, Z. Yi, S. Pan, Y. Zhao, Z. Zhao, F. Su, and

[10] G. H. Golash and S. N. Tiwari, “Deep Learning for Computer Vision: A Brief Review,” International Journal of Computer Applications, vol. 175, no. 9, pp. 6–10, 2021.

[11] V. Chandola, A. Banerjee, and V. Kumar, “A Survey on Machine Learning Techniques for Cyber Security,” ACM Computing Surveys, vol. 54, no. 7, pp. 1–36, 2021.

[12] A. S. G. V. Prasad and S. Bhattacharyya, “A Comprehensive Survey on Deep Learning in Computer Vision: Applications, Techniques, and Challenges,” Journal of Computer Science and Technology, vol. 35, no. 3, pp. 553–577, 2020.

[13] Y. G. Kim, K. H. Lee, and J. W. Lee, “Towards Effective Deep Learning for Cybersecurity,” IEEE Transactions on Dependable and Secure Computing, vol. 16, no. 3, pp. 500–511, 2019.

[14] R. G. K. Rao and S. P. B. Mallya, “Real-time Detection of Network Anomalies Using Deep Learning,” Journal of Network and Computer Applications, vol. 146, pp. 1–11, 2021.

[15] A. S. B. K. Verma and M. S. Y. Gohar, “Anomaly Detection in Computer Systems Using Deep Learning,” International Journal of Information Security, vol. 21, no. 5, pp. 499–514, 2022.