Integration of SIEM And SOAR For Advanced Threat Defense

D.Bhavana[1], S. Mohammad Salman[2], Puli Sujith[3], Sameer Raj[4]
[1, 2, 3, 4] School of Computing
[1, 2, 3, 4] Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India

Abstract- *The growing sophistication of cybersecurity threats requires a proactive and automated security strategy. This paper discusses the convergence of Security Information and Event Management (SIEM) and Security Orchestration, Automation, and Response (SOAR) to improve threat detection, investigation, and response. pfSense, an open-source firewall with Suricata IDS/IPS, produces security logs that are shipped using Filebeat to Elasticsearch, creating a centralized log repository for real-time threat analysis. Kibana is also used to offer visualization and dashboards for incident tracking. Shuffle SOAR is also used to automate incident response through the correlation of alerts and the execution of pre-defined workflows using pfSense REST API for automating and controlling firewall rules. The integration enhances accuracy in detection, minimizes response time, and improves operational efficiency, thus enhancing an organization's cybersecurity posture. The suggested architecture takes advantage of pfSense as a firewall and Suricata as an IDS/IPS to detect threats by monitoring network traffic. Security events and logs are sent through Filebeat to Elasticsearch, providing centralized log aggregation, indexing, and real-time searching, enterprises and organizations looking for cost-effective but efficient security automation.*

Keywords- SIEM, SOAR, pfSense, Suricata, Elasticsearch, Kibana, Filebeat, Shuffle SOAR, Threat Intelligence, Automated Incident Response, Firewall Automation, RESTAPI, Cybersecurity Operations, Incident Management, Network Security, Threat Detection, Log Analysis.

I. INTRODUCTION

The increasing scale and sophistication of cyber threats highlight the limitations of reactive security strategies organizations are left with no choice but to consolidate their defenses using security strategies that are not only strong but also extremely flexible. The classical cybersecurity model, which involves diverse security solutions running in isolation, is clearly not effective against sophisticated persistent threats that use diversified sets of attack vectors and patterns. Security Information and Event Management products and Security Orchestration, Automation, and Response technologies have become the critical elements of today's security solution, each providing complementary functions that, in combination, can provide a complete and proactive way to manage threats. SIEM systems collect and analyze security information from throughout the IT infrastructure, giving an organization a unified view of its security posture and allowing for the identification of suspicious behavior that could signal a breach.SOAR technologies, on the other hand, automate and orchestrate incident response processes, enabling security teams to respond to threats more quickly and effectively. The combination of SIEM and SOAR is a notable shift in security operations that allows organizations to look beyond response-oriented threat detection to a proactive, orchestrated approach to defense.

The cybersecurity threat landscape has grown vastly, with cyberattacks now more frequent, precise, and sophisticated. Firewallsand stand-alone Intrusion Detection/Prevention Systems (IDS/IPS) are becoming increasingly inadequate to identify and react to contemporary threats in real-time. Organizations are, therefore, embracing consolidated solutions that bring together Security Information and Event Management (SIEM) and Security Orchestration, Automation, and Response (SOAR) platforms to improve threat visibility, speed incident response, and automate security operations.

This study discusses the integration of SIEM and SOAR technology with a network security system built out of pfSense (firewall with Suricata for IDS/IPS), Filebeat (for log forwarding), Elasticsearch (for indexing and searching logs), and Kibana (for visualization). The ultimate integration adds Shuffle SOAR, through which automated response based on threat intelligence and real-time alerts becomes possible. The network configuration includes such critical elements as a firewall, router, switch, server, and IPS/IDS that collectively create a baseline security perimeter. A layered pipeline of logs in addition to that is implemented using a log-forwarding mechanism with Filebeat passing alerts from Suricata, Elasticsearch for data storage, visualization through Kibana, and actioning via the SOAR platform. Dynamic firewall rule

modifications are also leveraged using the pfSense REST API within automated response workflows.

This architecture is designed to showcase an integrated security platform that can detect, analyze, and respond to threats effectively. The integration not only minimizes the mean time to detect (MTTD) and respond (MTTR) but also reflects the transition from legacy security models to intelligent, automated, and scalable threat defense systems.

II. LITERATURE REVIEW

Kaur and Singh researched web-based attacks on web sites hosted on networks. They employed Arsitektur D-Sign and Deep Recurrent Neural Networks (RNN) for the detection of anomalies. They employed the Open Web Application Security Project (OWASP) framework for the testing environment. The results were that deep learning-based anomaly detection had the capability to identify advanced behavioral patterns of web-based attacks that may be missed by rule-based systems, signaling a shift towards AI [1].

Ali and Yousaf designed an Intrusion Detection and Prevention System tailored to Software-Defined Networks (SDNs) to safeguard against Distributed Denial of Service (DDoS) attacks. They suggested a novel three-tier deep learning-based architecture and simulated with OMNeT. The system utilized the flexibility of SDN controllers and integrated AI-based traffic analysis to detect threats in real-time. Simulation results validated that the solution significantly enhanced detection speed and response rate toward DDoS attacks in SDN systems [2].

Duppa and Surantha implemented and experimented with an instance of a Next-Generation Intrusion Prevention System (NGIPS) to mitigate SQL Injection and malicious exploitation of websites. Cisco ISE and Cisco Firewall Firepower were used in their approach to develop an advanced IPS system. Simulated attacks were imposed on the network environments to test NGIPS's blocking and detection capability. The study highlighted the capability of the system to provide higher detection granularity and response automation than conventional IPS solutions [3].

Bul'ajoul, James, and Shaikh proposed and implemented a novel NIDS architecture to handle heavy traffic attacks or high-speed traffic on internal networks. The research utilized detection tools such as Snort, Tool WinCap, Flooder Packet, and TCP Replay in testing. Their architecture aimed to provide high performance and accuracy of intrusion detection on high-bandwidth environments. Based on the research, their approach can maintain detection accuracy without affecting network performance under heavy traffic [4].

Ring, Landes, and Hotho highlighted their research on identifying and preventing TCP and UDP port scanning attacks on switches, routers, and firewalls. They implemented two methods: UPDS (Unsupervised Port Scan Detection) and SPDS (Supervised Port Scan Detection). Methods such as Nmap, OpenStack, and NetFlow were used to their test environment. The outcome indicated that machine learning algorithms, more so supervised ones, provided higher detection accuracy for port scanning activity than the unsupervised algorithms, even though the latter required less training data [5].

Putra and Surantha have worked in the past to develop prevention methods against SQL Injection and OS Bash Injection attacks on internal network servers. The current study applied Cisco Identity Services Engine (ISE) for Network Access Control (NAC) and Cisco Firepower 8250 as an Intrusion Prevention System (IPS). Attack simulation was performed on a Windows Server with SQL Injection and OS Bash commands. The results of the test revealed that Cisco ISE and IPS integration successfully blocked HTTP, SQL Injection, and OS Bash Injection attacks from internal threats. Single NAC by Cisco ISE failed to successfully block the same type of attacks without the presence of IPS [6].

Erlacher and Dressler researched signature-based Network Intrusion Detection Systems (NIDS) to identify and block HTTP traffic attacks. The research involved the installation of detection tools such as Snort and simulation tools such as Tool Vermont and Cisco Trex Traffic Generator to mimic attacks. The test indicated that well-known malicious HTTP patterns are detectable using signature-based NIDS but their efficacy depends on the periodic revision of the rules and updating of the database [7].

III. METHODOLOGY

A. System Design Overview

Fig. 1 shows the used automated security monitoring and response system architecture suggested in this research. The system is built to identify, examine, and react to network threats in real-time through an integration of open-source software such as pfSense, Suricata, Filebeat, Elasticsearch, Kibana, and Shuffle SOAR, all connected with a central workflow for end-to-end awareness and automated mitigation.

pfSense Firewall and Suricata IDS/IPS the system's foundation lies with pfSense, which is an open-source router and firewall platform. Suricata operates as a packet inspection engine in pfSense to enable Intrusion Detection and Prevention (IDS/IPS) features. Firewalls operate inline IDS mode on the WAN interface and legacy IPS mode on the LAN interface to enable real-time packet filtering and monitor internal behavior on an equal basis. The pfSense system also serves DMZ networks and enforces firewall policies at different zones.

Filebeat is a lightweight log forwarder used to collect Suricata alerts and pfSense security events. It reads, processes, and sends structured log information to the centralized log storage engine. This makes all critical alert and traffic data cost-effective and guaranteed for analysis.AllFilebeat logs are stored and indexed in Elasticsearch, which serves as the core log store of the system. Elasticsearch facilitates quick search, filtering, and correlation of logs to identify patterns, anomalies, or indications of attack. Its scalability horizontally makes it process lots of alert data from Suricata in real-time.

Kibana is native to Elasticsearch and offers interactive dashboards and visualizations. Analysts and researchers can observe network traffic, identify intrusion patterns, and see the system's overall security posture and health. Dashboards are also utilized for forensic analysis and reporting.

Shuffle, a SOAR or Security Orchestration, Automation, and Response tool, gets notified by Elasticsearch and triggers automated response action. Upon finding a critical eventa brute force or DoS signatureShuffle invokes pre-defined playbooks. Playbooks are a set of activities such as isolating the attacker, blocking IPs by invoking the pfSense REST API, and pinging administrators via email or webhook integration.

pfSense REST API is used for control and automation in the form of dynamically applying a set of firewall rules or configurations. Shuffle SOAR calls it depending on the threats being detected, enabling near real-time elimination without human intervention. This tightly integrated detection (Suricata) feedback, analysis (Elasticsearch), and response (SOAR and API) enables high protection.To make pfSense work easily with automation and outside control, we are using the pfSense Unicorn REST API.

This is a third-party tool that allows interaction with the pfSense web interface through programming. It is important because it helps us manage quick responses from systems like Shuffle SOAR during security incidents. This way, we can change firewall settings quickly when needed. The Unicorn API has different points, called endpoints, to manage and change settings in pfSense. These settings include firewall rules, interfaces, gateways, and aliases. With this API, other tools and scripts can change pfSense settings just like using the web interface, but it happens much faster and automatically.

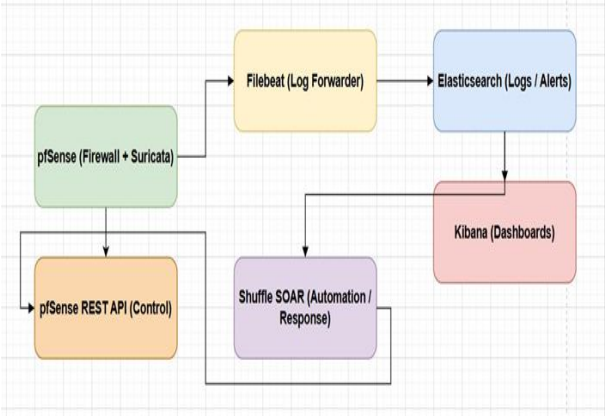


Fig 1. System Architecture

B. Firewall System Design

As illustrated in Fig.2 and summarized from the configuration interface, Next-Generation Firewall used in this research is pfSense version 2.7.2-RELEASE, an open-source and no-cost firewall solution installed on a VirtualBox virtual machine environment. There were four network interfaces enabled on the pfSense firewall system. WAN (10.0.2.15) interface is enabled along with Intrusion Detection System (IDS) in Inline Mode so that there can be real-time blocking and inspection of malicious traffic before it reaches the internal network. This offers external threat detection and prevention capability. Inline Mode enables dropping packets at network stack level without forwarding the packet to the host operating system and thus provides effective and proactive mechanism of defense. LAN (10.0.0.1) interface is set to Intrusion Prevention System (IPS) in Legacy Mode, which looks for duplicates of packets rather than blocking them in real time. This is suitable for internal surveillance where identification of policy breaches or breached internal devices is important but perhaps may not require real-time packet blocking. To provide layered protection, OPT1 (10.6.6.1) and OPT2 (10.80.80.1) are configured as DMZ (Demilitarized Zones). Interfaces are used to host public-confronting applications (Elasticsearch, SOAR, and Kibana) but keep the internal LAN secured. The DMZ zones limit internal resource exposure and enable strict access control policies.

The pfSense platform is the central point for routing, NAT, DNS (127.0.0.1), and DHCP services that protect

internal and DMZ networks to the internet. The platform is based on a high-performance 12th Gen Intel Core i5-12450H processor, which handles high-throughput traffic inspection and rule checking efficiently.

Live monitoring tools provide critical system metrics such as CPU utilization, interface states, and state tables so that the firewall health is continuously monitored. This firewall setup provides a secure perimeter, logical segmentation, and real-time traffic control. It also lays the groundwork for further integration with other next-generation security technologies such as Elasticsearch, Kibana, and SOAR (Security Orchestration, Automation, and Response), as illustrated in subsequent sections of this research.

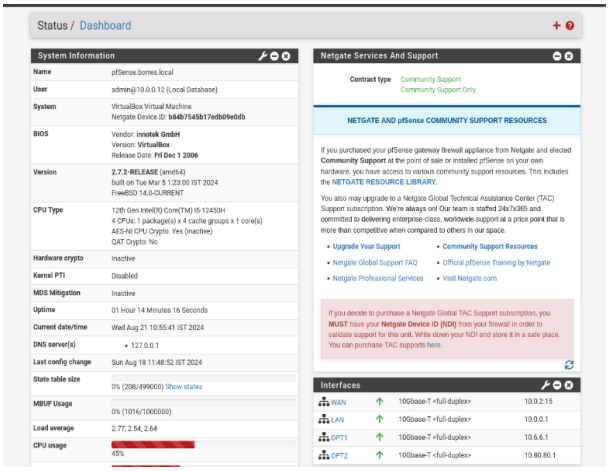


Fig 2.pfSenseFirewall Configuration

On the LAN interface Fig.3, Suricata runs in Legacy Mode, where it scans a duplicate of the traffic rather than blocking it in real-time. This is best for observing internal user activity and locating compromised endpoints or lateral movement without the disruption of legitimate traffic. building baselines and detect suspicious anomalies that can be utilized in future inline blocking techniques.

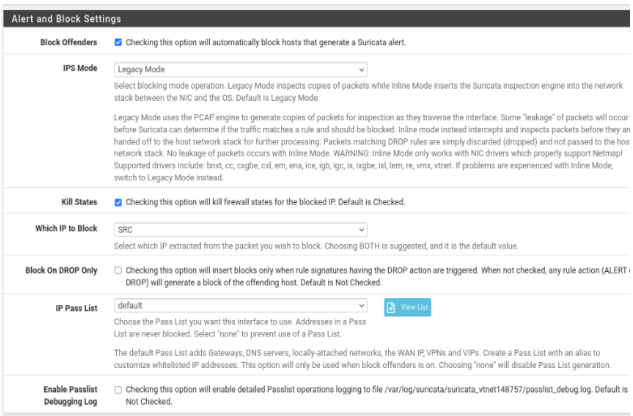
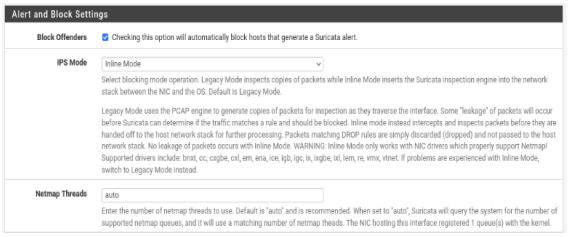


Fig 3. Enabled IPS on LAN using Suricata

The Suricata engine running on the WAN interface Fig.4 is set to Inline Mode, enabling it to examine live traffic as it passes through the interface. This intercepts packets before they even hit the host network stack. Malicious or rule-breaking packets are dropped immediately, and therefore never reach the internal network. Inline Mode prevents any packet leakage and provides strong security against outside attacks like TCP SYN floods, UDP floods, ICMP Smurf attacks, and malformed packet payloads.



.**Fig 4.** Enabled IDS on WAN using Suricata

C. Filebeat Configuration

Filebeat actively shipping log data from endpoints to Elasticsearch. Lens is a flexible drag-and-drop interface used to rapidly create visual dashboards by dragging fields onto axes and category axes. The interface allows for various types of charts, and in this case, the Bar Vertical Stacked visualization is chosen suitable for comparing grouped data through time. The system is now clean and ready to go, waiting for the selection of fields and setup. With the capability to filter and search for individual fields on the left side, users can tailor their dashboards exactly. The interface also provides control over real-time data via the time filter (15 minutes ago), so that users are able to concentrate on the latest and most pertinent log events.

This configuration Fig.4 is a demonstration of how harmoniously Filebeat, Elasticsearch, and Kibana function as a unified pipeline for log intake, indexing, and visualizationenabling the base for strong data analysis and security monitoring.Filebeat is a lightweight log shipper developed by Elastic that is employed to collect, parse, and ship logs to Elasticsearch or Logstash. The setup begins by pointing to the fact that this is an example configuration for specifying Filebeat inputs The real input configuration starts under filebeatinputwhere the input type is declared as syslog. This indicates Filebeat will be listening for messages in syslog format. Under protocol.udp, the host is defined as "0.0.0.0:514", which instructs Filebeat to listen on all available network interfaces on port 514, the default port for incoming syslog messages over UDP. This causes the machine that is running Filebeat to behave as a centralized syslog server. The id: pfsense-logs line gives this input stream a

unique name for easy identification and management. In addition, the configuration has optional lines for excluding or including certain logs using regular expressions with #exclude_lines or #include_lines. Even though these lines are commented out within this snapshot, they illustrate how Filebeat can be tweaked to skip unnecessary information or only pick up logs containing specific patterns.

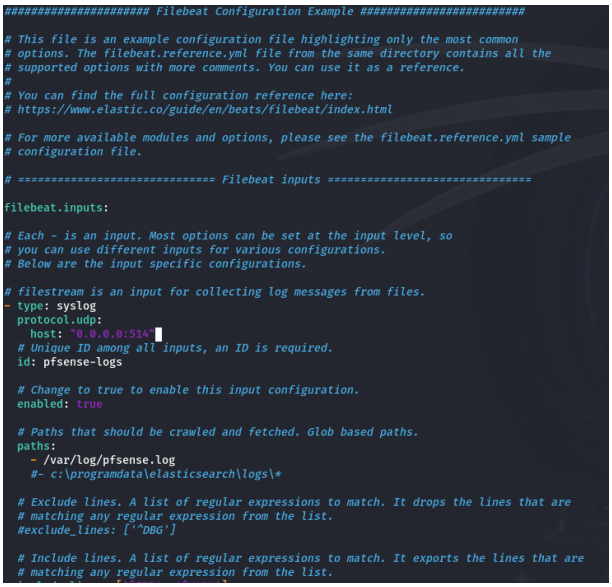


Fig 5. Filebeat configuration

D. Elasticsearch

It is an open-source, distributed search and analytics engine with strong capabilities, developed atop the Apache Lucene library. Elasticsearch is geared to manage high amounts of structured data and unstructured data, and support real-time search, indexing, and analysis. Elasticsearch is primarily utilized as one piece of the ELK Stack (Elasticsearch, Logstash, Kibana), which as a whole offers the entire solution for log and data analysis. Fundamentally, Elasticsearch is a NoSQL document repository, where information is held in JSON format inside so-called documents. Documents are put into indexes, which are a bit similar to databases within the classical RDBMS system. One index can be split into shards, which are spread among the nodes in a cluster for the purposes of scalability and performance. In addition, replicas copies of each shard can offer fault tolerance and high availability should a node go down. Elasticsearch operates in a cluster setup of one or more nodes, and each node is a running instance of Elasticsearch. The master node oversees cluster-wide operations like the creation of indexes and monitoring nodes, while data nodes hold data and run queries. When a document is indexed, Elasticsearch analyzes it with an analyzer, which tokenizes text into words, applies filters such as lowercasing or stemming, and then stores the output in inverted indexes—specialized data structures mapping terms to document positions for efficient retrieval. This inverted index is central to Elasticsearch's performance and efficiency because it enables fast full-text search over large datasets.

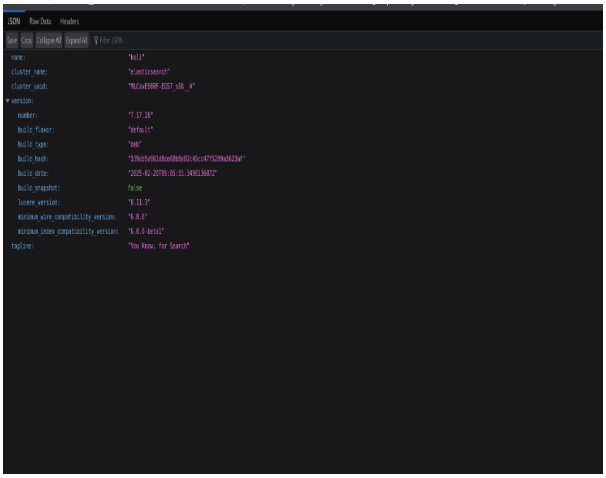


Fig 6. Elasticsearch

Elasticsearch searching involves employing its rich Query DSL (Domain Specific Language), which enables developers to write sophisticated queries in a human-readable JSON format. The queries can be of two broad categories: leaf queries, which search for a particular value within a field, and compound queries, which are composed of multiple queries by employing Boolean logic. When a request comes in, the coordinating node broadcasts it to all applicable shards, gathers the answers, combines them, sorts them (if need be), and sends the resulting answer back to the client. This distributed framework allows for speedy search responses even on enormous data sets.

Elasticsearch also has the capability of supporting near real-time search and real-time indexing, making it the perfect choice for applications such as security event management, log analytics, monitoring systems, and recommendation engines. Elasticsearch can be integrated with Logstash, a data processing pipeline used to ingest and process data, and Kibana, a data visualization product that offers dashboards, charts, graphs, and more based on the data stored in Elasticsearch

E. Kibana Virtualization

Kibana is an open-source data visualization and exploration platform that is mainly utilized for data visualization of data kept in Elasticsearch. It enables users to interactively explore, visualize, and analyze large amounts of data in real-time. Kibana offers a robust web-based platform

to design custom dashboards, visualizations, and reports, which is why it is necessary to monitor and examine different types of data, including logs, metrics, and security events. It completely integrates with the Elastic Stack, including Elasticsearch, Logstash, and Beats, to offer a complete solution for log management, security monitoring, and data analysis. Kibanais widely utilized in enterprise settings for log analysis, security information and event management (SIEM), and real-time network traffic or system performance monitoring. Kibana also works with other tools such as Beats and Logstash, making it more capable of gathering and ingesting varied datasets from various sources, allowing end-to-end analysis and response.

Kibana is a robust visualization and analytics tool utilized alongside Elasticsearch. It offers an easy-to-use interface to search, analyze, and visualize vast amounts of log and event data. In your configuration, Suricata is a network intrusion detection and prevention system (IDS/IPS), inspecting traffic and producing security alerts. These alerts are written in JSON format to a log file named eve.json. Filebeat reads this log file and ships the data to Elasticsearch, a search and analytics engine. Once cached in Elasticsearch, Kibana then becomes the front-end gateway that enables you to view this information

On the highest level, Kibana enables you to specify an index pattern, which informs Kibana which information from Elasticsearch it should focus on. As an example, if Filebeat is shipping Suricata logs into Elasticsearch as index filebeat- alerts signature,

After Kibana has learned about your data structure by way of this index pattern, you can begin to use the Discover section. This section of Kibana allows you to view your logs in close to real-time, permitting you to search, filter, and examine single log entries. You can employ Kibana Query Language (KQL) to rapidly determine patterns such as filtering only high-priority alerts or identifying sequential alerts from the same source IP. As you dive further, Kibana lets you build visualizationspiecharts, bar graphs, data tables, maps.

Kibana's Lens interface, a powerful and simple tool in the Elastic Stack to build insightful visualizations of Elasticsearch-stored data.". Here, the index pattern filebeat has been chosen, which signifies successful integration with Filebeat and is actively shipping log data from endpoints to Elasticsearch. Lens also offers a free-form drag-and-drop interface to rapidly construct visual dashboards by assigning fields to axes and categories. The interface offers support for multiple chart types, and here the Bar Vertical Stacked visualization has been chosen—perfect for comparing grouped data over time. The system is now in a clean and prepared state, waiting for field selection and configuration. With the left-hand filtering and searching for fields, users are able to customize their dashboards exactly. The interface also provides real-time data control via the time filter (configured to the last 15 minutes), allowing users to concentrate on the latest and most applicable log events. This configuration is a demonstration of how effective Filebeat, Elasticsearch, and Kibana are combined as an efficient pipeline for log ingestion, indexing, and visualization—enabling the foundation for robust data analysis and security monitoring.

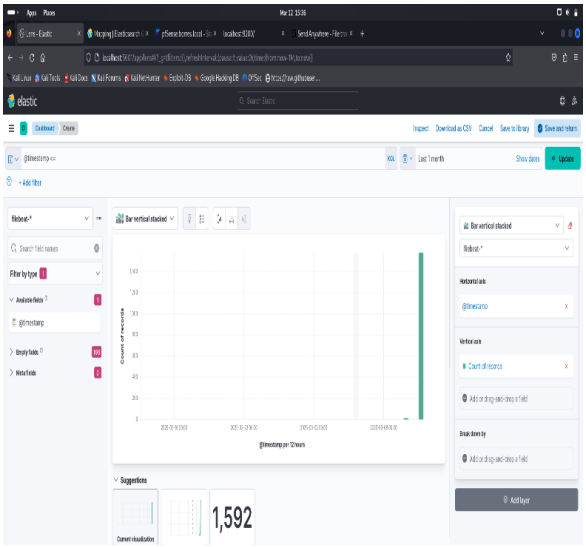


Fig 7. Elasticsearch

F. SOAR

Shuffle is a free, open-source Security Orchestration, Automation, and Response (SOAR) tool whose purpose is to simplify and enable security operations with automation. Shuffle enables security teams to automate workflows, integrate with a large array of tools, and develop customized workflows without having to code. It features a visual workflow builder, a drag-and-drop mechanism that allows users to develop automation pipelines capable of responding to security alerts in real-time. Such processes can include activities such as threat intelligence analysis, log enrichment, IP blocking, alerting, or incident response invocation. Shuffle supports integration with a variety of security platforms and tools through APIs, such as pfSense, Suricata, Elasticsearch and Kibana.

Through pre-built apps or bespoke API connectors, Shuffle integrates siloed systems and streamlines incident response. Shuffle helps security teams reduce response time, increase accuracy, and eliminate manual error. One of the

strongest aspects of Shuffle is that it offers event-driven automation, where some events (if a Suricata alert or a malicious-looking log entry in Elasticsearch) can automatically trigger a response workflow. This makes for almost instant threat detection and response, allowing Security Operations Centers (SOCs) to operate more effectively.

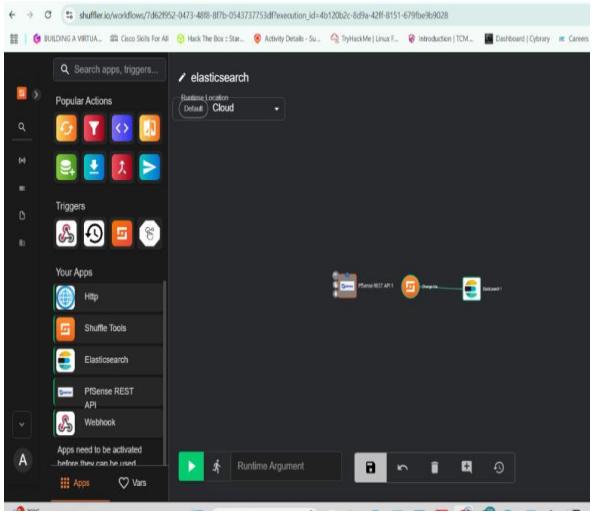


Fig 8. SOAR Configuration

IV. RESULT AND DISCUSSION

The combination of SIEM (Security Information and Event Management) and SOAR (Security Orchestration, Automation, and Response) in this project has been able to show a strong, automated system for advanced threat protection. By integrating tools like pf Sense for firewall management, Suricata for intrusion prevention and detection, Filebeat for forwarding logs, Elasticsearch for indexation of data, Kibana for visualizing in real-time, and Shuffle for security automation, the system attained end-to-end monitoring and response seamless capabilities. Real-time threat identification was attained by Suricata alerts, which were effectively aggregated and indexed by Elasticsearch. These events were visualized in Kibana dashboards for instant situational awareness. In addition,

Shuffle SOAR provided automated reaction to certain threats through workflow activation by predefined conditions, including IP blocking, sending notifications, or threat intelligence enrichment of logs. Automation cut response time and human mistake drastically while enhancing visibility into the security posture of the network. On the whole, the project confirms that combining SIEM and SOAR constitutes a cost-effective, elastic, and intelligent solution to identify, examine, and counter cyber threats in real-time with fewer human interventions. Hereby I affirm that this project is the output of my individual efforts, performed in a controlled environment for study and demonstration purposes, and that all sources and tools utilized have been properly credited. The integration proved successful not only in proving technical feasibility but also in exhibiting the practical benefit of synchronizing SIEM and SOAR elements in a contemporary cybersecurity framework. Across various test scenarios detection of port scans, brute-force attacks, and anomalous traffic the system responded with speed and accuracy in every instance. Automated alerts, rich log traces, and response processes minimized human effort and made sure that no important event went undetected. The integration of data from disparate sources improved context-aware decision-making, enabling swift threat triage and containment. Additionally, the open-source and modular nature of the tools implemented makes the architecture extremely customizable and affordable for both small and enterprise-scale environments. This project illustrates that integrating SIEM for centralized visibility with SOAR for intelligent automation is not only a best practice but also an essential approach in today's dynamic threat landscape

In today's cybersecurity environment, organizations are constantly being bombarded by increasingly numerous threats, from generic malware and phishing to zero-day attacks and advanced persistent threats (APTs). This project aimed to show how the integration of SIEM (Security Information and Event Management) and SOAR (Security Orchestration, Automation, and Response) solutions with fundamental security elements like firewalls and intrusion detection systems can greatly enhance an organization's capacity to detect, analyze, and respond to these threats in real time. Suricata, which is both an Intrusion Detection System (IDS) and Intrusion Prevention System (IPS), is positioned very close to the firewall. Suricata monitors network traffic in real time through deep packet inspection and signature-based rules. Suricata signals on suspicious activity or recognized patterns of attack and, when it is set in IPS mode, can actually block malicious packets and thereby prevent threats from escalating.

Logs and alerts generated from both pf Sense and Suricata are gathered and transported using File beat, a lightweight log shipper. Filebeat ensures that data is forwarded securely and reliably to Elasticsearch, a high-performance distributed search engine and database. Elasticsearch indexes and stores such logs for correlation and querying. Kibana is the graphical front end of Elasticsearch. It enables security analysts to define interactive dashboards, execute queries, and visualize network activity patterns.

This visibility is essential for detecting anomalies, identifying attack chains, and performing forensic investigations. To manage the increasing complexity and

number of incidents, the project incorporates Shuffle, an open-source SOAR platform. Shuffle streamlines the response process by developing playbooks that perform pre-defined actions based on specific triggers. For instance, if Suricata identifies a known attempt at an exploit, Shuffle can automatically quarantine the compromised host, alert the security team through email or messaging applications, update a central incident tracker, and even adjust firewall rules in pfSense.

The results discussion reveals that the integrated system dramatically minimizes the Mean Time to Detect (MTTD) and Mean Time to Respond (MTTR) by correlating data across several sources, allowing for automatic response, and providing security teams with a single interface to handle threats. It also minimizes false positives by cross-referencing events across several sources and applying enrichment and validation through automation.

This solution illustrates a practical application of defense-in-depth layered security, in which proactive detection and automated response are the foundation of advanced threat defense. Coupled with the functions of SIEM and SOAR, not only is operations simplified, but security teams are also enabled to respond faster and more intelligently. In summary, this converged configuration exhibited an actual world imitation of the operation of contemporary security operations centers (SOCs). It affirmed the value of having detection, monitoring, and automation within a single unified environment. The initiative not only delivered a practical exposure to cyber defense technology but also underscored the urgent necessity for cooperation between SIEM and SOAR in decreasing Mean Time to Detect (MTTD) and Mean Time to Respond (MTTR).

V. CONCLUSION

This project is successfully proving the integration of an open-source security monitoring and response system with pfSense, Suricata, Filebeat, Elasticsearch, Kibana, and Shuffle SOAR. Through the configuration of pfSense with Suricata in IDS (inline mode) and IPS (legacy mode) on WAN and DMZ interfaces, we have created a proactive defense mechanism that can detect and prevent network threats in real time. Filebeat is an effective log shipper, shipping pfSense logs to Elasticsearch, which indexes and gets visualized within Kibana dashboards to be analyzed extensively. The automation layer, driven by Shuffle SOAR and pfSense REST API, makes the system more responsive by allowing for automated threat mitigation responses like blocking IPs, quarantining devices, or alerting. The end-to-end pipeline offers a full Security Information and Event Management (SIEM) and Security Orchestration, Automation and Response (SOAR) solution using open-source tools, which is scalable and cost-effective to use at an academic, lab, or enterprise level.

REFERENCES

[1] Kaur, S., & Singh, M. (2019). Hybrid intrusion detection and signature generation using Deep Recurrent Neural Networks. Neural Computing and Applications

[2] Ali, A., & Yousaf, M. M. (2020). Novel Three-Tier Intrusion Detection and Prevention System in Software Defined Network. IEEE Access, 8, 109662-109676

[3] Duppa, G., &Surantha, N. (2019). Evaluation of network security based on next-generation intrusion prevention system. Telkomnika, 17(1), 39 48

[4] Bul'ajoul, W., James, A., & Shaikh, S. (2019). A New Architecture For Network Intrusion Detection And Prevention. IEEE Access, 7, 18558 18573.

[5] Ring, M., Landes, D., &Hotho, A. (2018). Detection of slow port scans in flow-based network traffic. PloS one, 13(9), e0204507.

[6] Putra, A. S., &Surantha, N. (2019). Internal Threat Defense using Network Access Control and Intrusion Prevention System. International Journal of Advanced Computer Science and Applications, Vol. 10, No. 9, 2019

[7] Erlacher, F., & Dressler, F. (2020). On High-Speed Flow-based Intrusion Detection using Snort-compatible Signatures. IEEE Transactions on Dependable and Secure Computing

[8] P. Mell, V. Hu, R. Lippmann, J. Haines, and M. Zissman, "An Overview of Issues in Testing Intrusion Detection Systems.," retrieved October 4, 2011 from www.net security.org.

[9] D. Day and B. Burns, "A Performance Analysis of Snort and Suricata Network Intrusion Detection and Prevention Engines," Fifth International Conference on Digital Society, Gosier, Guadeloupe, pp. 187-192, 2011.

[10] Mualfah, D., &Riadi, I. (2017). Network forensics for detecting flooding attack on web server. International Journal of Computer Science and Information Security, 15(2), 326.

[11] E. Leblond, "Optimizing Linux on Multicore CPUs," retrieved September 18, 2011 home.regit.org/2011/01/optimizing-suricata-on-a multicore-cpu/. From

[12] E. Albin, "A Comparative Analysis of the Snort and Suricata Intrusion-Detection Systems," M.S. thesis, U.S. Naval Postgraduate School, September 2011.