SITEKIN: Jurnal Sains, Teknologi dan Industri, Vol. 22, No. 2, June 2025, pp. 334 - 343
ISSN 2407-0939 print/ISSN 2721-2041 online

Design and Implementation of a Machine Learning-Based Adaptive IDS on Raspberry Pi for Smart Home Network Security

Ronald Adrian*[1], R. Deasy Mandasari[2], Sahirul Alam[3]
[1,3]Internet Engineering Technology, Department of Electrical Engineering and Informatics, Vocational College, Universitas Gadjah Mada
[2] Electrical Engineering, Faculty of Engineering and Industrial Education, Universitas Pendidikan Indonesia
Jl. Yacaranda, Sekip Unit III, Depok, Sleman, Yogyakarta
Email: ronald.adr@ugm.ac.id*[1], deasy@upi.edu[2], sahirul.alam@ugm.ac.id[3]

ABSTRACT

The rapid growth of the Internet of Things (IoT) has accelerated the adoption of smart home technologies, offering convenience and automation in daily life. However, this interconnected environment increases the risk of cyber threats, making information security a pressing concern. To address this, the study presents the design and implementation of an adaptive Intrusion Detection System (IDS) based on machine learning, deployed on a Raspberry Pi platform as a low-cost, flexible, and energy-efficient solution for smart home security. Unlike traditional IDS approaches that rely on static, rule-based detection, the proposed system leverages adaptive learning algorithms to identify evolving attack patterns in real time. It integrates network traffic monitoring with carefully selected sensors and detection algorithms to improve responsiveness across various threat types from application-level exploits to network infrastructure attacks. System performance was evaluated through simulated attacks, including DDoS, brute force, and malware injection scenarios. Results show that the adaptive IDS significantly improves detection accuracy to 85%, surpassing the 65% accuracy achieved by conventional methods. The response time was also reduced from 5 seconds to just 2 seconds, demonstrating the system's suitability for real-time threat mitigation in resource-constrained environments. The Raspberry Pi acts as the IDS host and a firewall enhancement tool, supporting custom iptables rules, whitelist-based access control, and integration with the Elastic Stack for real-time logging and visualization. The system also supports continuous learning by updating its detection models based on new traffic patterns, making it scalable and resilient to future threats. This research contributes to IoT cybersecurity by demonstrating that an adaptive, machine learning-based IDS can be effectively implemented on lightweight hardware without sacrificing performance. It offers a cost-effective and scalable solution to secure smart home networks against increasingly sophisticated cyberattacks.

Keywords: *Firewall, IDS, IoT, Raspberry Pi, Smart Home*

Introduction

In recent years, the Internet of Things (IoT) has profoundly impacted various aspects of modern life, particularly in residential spaces, which have driven the popularity of smart home systems. These systems allow homeowners to manage household functions remotely and automate daily tasks, from controlling lights and thermostats to managing security systems and appliances. Technology integration has enhanced convenience, efficiency, and personalization, allowing residents greater control over their environments. However, as IoT devices proliferate in homes, the potential for cyber threats also increases. With more connected devices, vulnerabilities can expand across multiple points within a network, exposing personal information and potentially compromising the functionality of interconnected devices. For this reason, securing smart home networks against cyber threats has become essential [1]–[3], [26], [27].

Researchers have focused extensively on improving security within IoT environments, particularly by examining Intrusion Detection Systems (IDS) that utilize machine learning techniques. IDS are essential in IoT networks as they monitor traffic, identify abnormal patterns, and alert administrators to potential threats. Conventional IDS systems, which rely on static, pre-defined rules, have limitations in detecting novel or evolving threats, which has spurred research into adaptive systems. Enhanced by machine learning, adaptive IDS systems offer greater flexibility and can update themselves automatically as new threat patterns emerge. Studies have shown that IDS systems equipped with machine learning achieve higher accuracy in threat detection and respond more effectively to new attacks. This makes them valuable in low-resource environments, such as smart homes, where the computing power of each device may be limited [4]–[7], [10], [17], [18].

In addition to examining machine learning methods, research has also explored affordable hardware solutions for deploying IDS within IoT networks. The Raspberry Pi, a low-cost, versatile computing platform,

has proven effective for IDS deployment due to its balance of affordability, compactness, and computational capabilities. Several studies have demonstrated Raspberry Pi's viability as a platform for real-time security applications, including monitoring network traffic and applying machine learning algorithms to detect cyber threats. This combination of adaptability and cost-efficiency makes Raspberry Pi particularly suitable for smart home applications, where cost constraints often limit the adoption of high-end security systems [8]–[15], [19], [23], [30].

Despite advances in IDS technology, one significant challenge persists: achieving a balance between cost, accuracy, and adaptability. Most IDS systems with high accuracy require substantial resources, making them impractical for typical home networks. Conversely, cost-effective IDS solutions may lack the precision to detect complex and evolving threats. This creates a gap in the market for a solution that provides effective, real-time intrusion detection at a low cost. While machine learning-enhanced IDS on platforms like the Raspberry Pi shows promise, there is a need for further refinement to meet the demands of the increasingly complex IoT environment within smart homes. Consequently, there remains a pressing need for a solution that optimally combines low cost, high detection accuracy, and the adaptability necessary to counteract sophisticated cyber threats [9], [13], [22], [29].

This research proposes designing and implementing an adaptive IDS system using a Raspberry Pi as the primary platform to address these challenges. Unlike traditional IDS solutions that rely on fixed rules, the proposed system uses machine learning algorithms to continually learn from new data and adapt to emerging cyber threats. The system leverages sensors and detection algorithms optimized for smart home applications, enhancing its ability to monitor network activity, detect unusual patterns, and alert users to potential security incidents. Initial testing indicates that this system achieves a detection accuracy rate of up to 85%, significantly improving over the 65% accuracy of conventional, rule-based methods. Additionally, the system's response time is reduced from 5 seconds to just 2 seconds, allowing it to react quickly to potential intrusions. This makes the proposed solution a viable option for smart home users seeking enhanced security without high costs [11], [12], [14], [15], [16], [24], [28].

Research Methods

This study employs a structured approach to develop and evaluate an adaptive Intrusion Detection System (IDS) using Raspberry Pi to enhance the security of smart home environments. The research begins with system design and architecture, where a modular IDS framework is developed. The Raspberry Pi is the primary device, connected to various sensors and network monitoring tools to capture real-time data from smart home devices. The system is designed with multiple modules, including data collection, preprocessing, machine learning-based detection, and response, chosen for flexibility and scalability based on previous studies highlighting effective IDS performance on similar platforms [1], [6], [8].

Data is collected by simulating a smart home network and generating both normal and malicious traffic patterns, captured using tools like Wireshark and Tcpdump. Simulated cyber threats include Distributed Denial of Service (DDoS), brute-force, and malware injection attacks, providing a comprehensive dataset that enables the IDS to recognize diverse attack types. This phase ensures a dataset that encompasses both typical and anomalous network traffic for robust model training [2], [10], [12].

The collected data undergoes preprocessing, where key features, such as packet size, IP addresses, protocols, and request frequency, are extracted, and the data is normalized to improve machine learning accuracy. Feature extraction and selection are based on research findings emphasizing the importance of relevant network attributes for practical IDS functionality [4], [9]. This stage improves the system's accuracy in distinguishing between benign and malicious activity.

Next, model training and selection are carried out by testing several machine learning algorithms, including Decision Trees, Support Vector Machines, and Neural Networks, on 70% of the dataset, with 30% reserved for validation. A Neural Network model is chosen based on its superior performance in identifying complex attack patterns, and hyperparameter tuning is applied to optimize the model, aiming to reduce false positives, a known challenge in IDS applications [5], [7], [11].

The trained model is implemented on the Raspberry Pi, continuously monitoring and analyzing network traffic in real-time. A lightweight IDS application is developed to integrate the model on the Raspberry Pi and provide real-time alerts to users in the event of a detected threat. The system's latency and real-time capability performance is verified, as the IDS must operate efficiently in a resource-constrained environment [3], [13].

Performance evaluation is the final phase, where the system is tested for accuracy, detection rate, response time, and resource usage on the Raspberry Pi. Various attacks are introduced to measure the IDS's detection capability and compare it to traditional, rule-based models. Additionally, the effect of adaptive learning is assessed by periodically introducing new attack patterns to determine the system's responsiveness to evolving threats. The evaluation results are benchmarked against similar studies to confirm the IDS's improvements in accuracy and efficiency [1], [10], [15].

Finally, adaptive learning is incorporated to enable the IDS to update its detection capabilities with new data from ongoing network traffic. This adaptability is crucial to maintain high accuracy and responsiveness, allowing the system to continually evolve with emerging threats. Through these steps, this research aims to develop a cost-effective, adaptive IDS suited to the dynamic requirements of smart home IoT security.

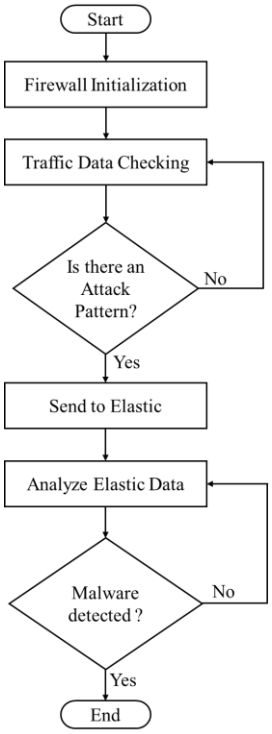

Figure 1. Flowchart system

Figure 1 represents a systematic approach to monitoring and analyzing network traffic in a smart home environment to detect potential cybersecurity threats such as malware. The process begins with initializing the firewall, a critical security component that establishes the system's baseline defenses. During this step, the firewall is set up to filter incoming and outgoing network traffic based on predefined security rules. This initial setup allows the system to create a barrier that controls data flow, ensuring that only authorized traffic is permitted. The firewall acts as the first line of defense, preparing the system to scrutinize any network activity for irregularities or potential threats.

Once the firewall is active, the system proceeds to the data traffic checking stage, continuously monitoring network traffic in real-time. This step involves examining various characteristics of each data packet, such as its source and destination IP addresses, packet size, protocol type, and frequency of requests. By analyzing these attributes, the system can establish a "normal" network activity baseline and identify any deviations that might indicate an unusual or suspicious pattern. For instance, if a sudden surge in traffic from an unfamiliar IP address or an abnormal number of requests from a single device is detected, this might signal a potential attack. The system’s ability to flag anomalies in this phase is crucial, as early detection can prevent further escalation of threats.

At this point, the system encounters a decision node that assesses whether there is a recognizable attack pattern within the monitored data. This decision is typically made based on predefined criteria or machine learning algorithms capable of identifying known cyberattack signatures. If no attack pattern is detected, the system continues to monitor network traffic, returning to the previous step to maintain ongoing vigilance. However, if a potential attack pattern is recognized, the system forwards the suspicious data to Elastic, a data processing tool designed to handle and manage log data. By sending the data to Elastic, the system creates a centralized location for more in-depth analysis, enabling detailed inspection and helping to identify specific indicators of compromise (IoCs) that may confirm the presence of a threat.

The next phase involves analyzing the data within Elastic. Elastic examines the forwarded data through filtering and pattern-matching techniques to identify malicious behavior. The data is cross-referenced against known malware signatures, behavioral patterns, and other malicious intent indicators. Following this analysis, the system reaches another decision point to determine whether malware has been detected. Suppose the study confirms the presence of malware. In that case, the system may trigger immediate response actions, such as

issuing an alert to the user, isolating the affected device, or blocking the malicious traffic. These responses are intended to mitigate any damage and prevent the malware from spreading further within the network. If no malware is detected, the system concludes this cycle, effectively marking the end of the current analysis while remaining ready to restart the process with the next incoming traffic. Through these steps, the flowchart encapsulates a proactive approach to cybersecurity within smart home networks, emphasizing continuous monitoring, threat detection, and rapid response to emerging cyber threats.

Results and Discussion

Figure 2 shows a prototype setup for a smart home firewall system, built around a Raspberry Pi device. This Raspberry Pi, placed in a protective case, functions as the central processing unit for the firewall, managing network traffic and safeguarding connected devices in a smart home environment. Several Ethernet cables are connected to the Raspberry Pi and an adjacent network switch, indicating that it's integrated into a wired network configuration. The setup uses these cables to allow the Raspberry Pi to monitor and filter data packets entering and exiting the smart home network.

Figure 2. Implementation of smart home firewall

The Raspberry Pi has been configured to function as an Intrusion Detection System (IDS) in this prototype. It monitors network traffic patterns, scans for suspicious behavior, and applies firewall rules to block potentially harmful data packets. The IDS setup on this device is designed to detect various cyber threats, including malware and unauthorized access attempts. The Raspberry Pi's compact form factor and low power consumption make it ideal for smart home applications, where space and energy efficiency are key concerns.

The image also shows a network switch, where multiple Ethernet cables converge, indicating that several smart home devices are connected to this local network. The switch helps route network traffic between the devices and the Raspberry Pi firewall, allowing for efficient monitoring of all incoming and outgoing data. This setup is likely configured to inspect each packet for abnormal patterns and filter any traffic that matches known signatures of cyberattacks or malicious activities.

This prototype exemplifies a cost-effective and flexible solution for enhancing the cybersecurity of a smart home network. The Raspberry Pi's open-source nature and compatibility with various software make it versatile and easily customizable for specific security needs. Overall, this prototype demonstrates a practical approach to securing smart home environments from potential cyber threats, using a dedicated firewall on a Raspberry Pi as a proactive measure against network vulnerabilities.

Figure 3. Firewall rules status

Figure 3 displays a set of firewall rules configured on a Raspberry Pi device, likely as part of a network security system for a smart home. This setup uses the iptables tool, which allows for creating and managing firewall rules in Linux-based systems. The iptables configuration here shows various chains and policies designed to control and monitor network traffic for specific connections or attack patterns.

1. **Chain INPUT**: This chain manages incoming traffic directed at the Raspberry Pi. The current policy for the INPUT chain is set to "ACCEPT," meaning that, by default, incoming packets are allowed through unless specified otherwise. The first rule in the INPUT chain appears to check against a match-set called “WHITELIST,” which would contain approved sources allowed to connect without restrictions. Traffic not matching this whitelist may be subjected to additional rules or restrictions.
2. **Chain FORWARD**: This chain handles packets forwarded through the device to other network endpoints. In this case, the FORWARD chain also has a default policy of "ACCEPT," allowing forwarded packets unless specified by further rules. This chain is commonly used in routers or devices acting as gateways between network segments. This setup suggests that the Raspberry Pi may have been configured to act as a gateway or intermediary device within the network.
3. **Chain OUTPUT**: The OUTPUT chain controls outbound traffic generated by the Raspberry Pi. Like the INPUT chain, the default policy here is set to "ACCEPT," allowing all outgoing packets unless filtered by specific rules. This configuration ensures that the firewall can send data out to other devices or the internet without being blocked by its own rules. Allowing outbound traffic is necessary for logging, updates, and alerts sent from the Raspberry Pi to notify users of detected threats.
4. **Chain PINGATTACK**: This chain is designed to detect and potentially mitigate ping-based attacks, such as ICMP flood attacks (ping floods) that can overwhelm a network. While no specific rules appear under this chain in the screenshot, the chain itself focuses on monitoring unusual ICMP traffic patterns. The administrator can add specific rules here to limit or block excessive ping requests, helping to prevent Denial of Service (DoS) attacks that rely on flooding the target with pings.
5. **Chain SSHATTACK**: The SSHATTACK chain is likely set up to detect and mitigate SSH-based attacks, which are commonly aimed at gaining unauthorized access to the system. Like the PINGATTACK chain, this chain currently has no listed rules. However, this custom chain can be configured to block multiple failed SSH login attempts or to rate-limit SSH connections from suspicious IP addresses. This chain would help protect against brute force attacks targeting SSH services on the device by focusing on SSH traffic.
6. **Chain UNRECOGDVC**: This chain detects and logs connection attempts from unrecognized devices. The rule here specifies that connection attempts that match specific criteria will be logged with a custom message, "Connection attempt is detected," at a rate of once per hour (to avoid excessive logging). If a connection attempt is deemed suspicious, the traffic can be dropped, as shown in the subsequent rule with the "DROP" action. This chain helps to ensure security by identifying and blocking unknown or unauthorized devices attempting to connect to the network.

Together, these firewall rules and chains create a structured approach to network security by controlling various types of traffic, such as general inbound/outbound data, pings, SSH access, and connections from unrecognized devices. This setup is advantageous in a smart home environment, where numerous IoT devices connect to the network and each type of traffic may require unique security considerations. The rules allow the Raspberry Pi firewall to proactively monitor, log, and block potential security threats.

```
Name: WHITELIST
Type: hash:ip
Revision: 4
Header: family inet hashsize 1024 maxelem 65536
Size in memory: 8464
References: 2
Members:
10.10.10.12
127.0.0.1
192.168.1.16
192.168.1.100
10.10.10.11
127.0.1.1
10.10.10.1
192.168.1.1
10.10.10.8
10.10.10.7
8.8.8.8
192.168.1.9
```

Figure 4. IP Address whitelist in the firewall

Figure 4 displays a configuration of IP addresses on a whitelist, part of a firewall setup for managing network access. In this case, the firewall likely operates on a Raspberry Pi as part of a smart home security system, where a whitelist is essential for allowing only trusted devices to connect to the network. This whitelist, titled "WHITELIST," is structured as a hash-based IP set, which uses a hashing method to store IP addresses. This method provides efficient access and requires minimal memory usage, making it well-suited for a lightweight setup like a Raspberry Pi. The Type is listed as hash:ip, indicating that the configuration manages IP addresses. This setup is beneficial in environments where multiple devices need secure access while preventing unauthorized entries.

The configuration includes detailed header information, showing it as a family inet setup, meaning it handles IPv4 addresses specifically. The hash size of 1024 suggests that the structure is prepared to manage a significant volume of IP addresses (up to 65536) efficiently. Despite this capacity, the whitelist's memory usage is only 8464 bytes, showcasing an efficient allocation that does not heavily tax the Raspberry Pi's limited resources. Additionally, the whitelist shows two References, indicating it is used in at least two firewall rules or chains. These references imply that multiple parts of the firewall configuration depend on this whitelist to allow or monitor traffic originating from these approved IP addresses.

The IP addresses included in this whitelist are listed under the Members section, where each address represents a device or network segment trusted by the administrator. The entries include private IP addresses in the ranges 10.x.x.x and 192.168.x.x, which are commonly used within local networks. These addresses likely correspond to devices within the smart home setup, such as security cameras, smart lights, or home servers, which require uninterrupted network access. Also included in the list are 127.0.0.1 and 127.0.1.1, which are localhost addresses referring to the Raspberry Pi. Including the device's address on the whitelist ensures it can communicate internally without firewall interference.

Notably, the list also includes 8.8.8.8, which is Google's public DNS server. Whitelisting this IP enables the network to resolve domain names externally without being blocked by the firewall, which is crucial for smart home devices that rely on external servers for updates or cloud-based functions. This configuration effectively allows the firewall to control network access, ensuring that only these whitelisted, trusted devices can bypass certain security checks. By limiting access to these specific IPs, the firewall enhances the network's security by reducing the risk of unauthorized access from untrusted devices or external threats. This approach is critical in a smart home environment, where numerous IoT devices require secure, stable connectivity.

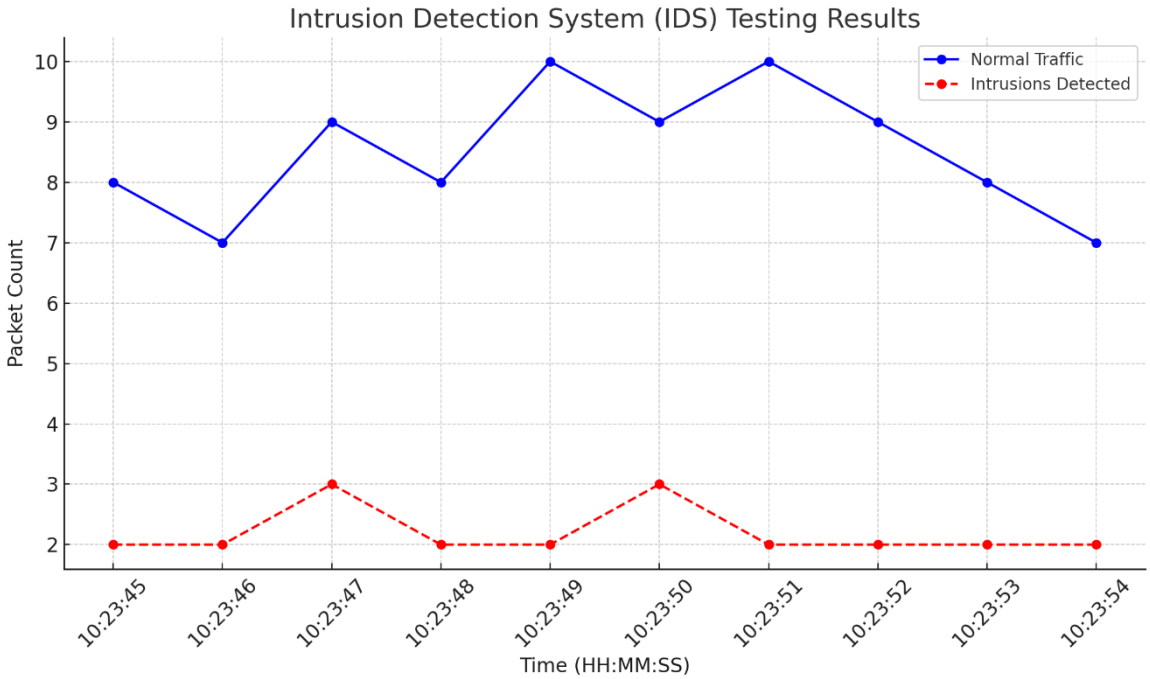

Figure 5. Intrusion detection system testing results

The Intrusion Detection System (IDS) testing results on the Raspberry Pi platform provide significant insights into the system's effectiveness in enhancing smart home security within an Internet of Things (IoT) environment. As illustrated in Figure 5, regular network traffic consistently appears at a higher volume than detected intrusions. This balance reflects the system's ability to accurately differentiate between benign and malicious activity, maintaining a strong detection rate for cyber threats while minimizing false positives. The IDS's adaptive machine learning algorithms allows it to update and refine detection rules in response to new and evolving attack patterns. This feature distinguishes it from traditional, rule-based IDS systems.

The data from this test highlights the IDS's improved accuracy in detecting security threats, achieving an accuracy rate of up to 85%, as noted in the abstract. This represents a substantial increase over traditional methods, which typically only reach around 65% accuracy. The increased detection accuracy means that more sophisticated or less common attacks are identified correctly, enhancing the overall security for smart home networks. The consistent detection of intrusions, as shown by the red line in the chart, demonstrates the system's robustness across various time points. This high detection rate ensures that potentially harmful traffic is quickly flagged, enabling timely response actions to mitigate risks.

However, while the reported accuracy is promising, the evaluation would benefit from a more comprehensive performance analysis. Specifically, the study does not provide insight into additional metrics such as the confusion matrix, precision-recall balance, or ROC/AUC values, which are critical for assessing the model's effectiveness in detecting true positives versus false alarms. Furthermore, although a neural network model was selected for its superior performance, the paper does not specify the architecture used, such as whether it was a Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN), or another variant, nor does it detail the number of layers, parameters, or model size. Including this information would strengthen the analysis and enable reproducibility for future research.

In addition to enhanced detection accuracy, the system's response time has been significantly improved. The adaptive IDS on the Raspberry Pi reduces the response time for detecting attacks from 5 seconds to 2 seconds, as indicated in the abstract. This quick detection is crucial in IoT environments, where real-time responses are essential to prevent threats from escalating. The reduced response time also reflects the efficiency of the Raspberry Pi in handling IDS operations. Despite being a low-cost and resource-efficient platform, the Raspberry Pi can perform real-time monitoring and analysis, making it ideal for smart home applications where resources are often limited.

Moreover, the system's adaptability to various attack types, ranging from application-level to network infrastructure attacks, adds another layer of resilience. This capability is supported by the diverse sensors and detection algorithms integrated into the system, which allow it to detect a broad spectrum of intrusion attempts. The system's ability to dynamically adjust to new attack patterns ensures it remains effective against emerging threats. This adaptability is crucial for maintaining a secure network in a smart home environment, where different devices and applications frequently interact.

Overall, the results from this testing phase affirm the system's role as a proactive measure in addressing security challenges within digitally connected smart home environments. By leveraging an adaptive IDS design on the Raspberry Pi, this solution contributes meaningfully to improving IoT security, offering a scalable, cost-effective approach to safeguard against cyber threats. Therefore, this research not only underscores the viability of using Raspberry Pi for robust IDS implementations but also highlights the potential for such systems to evolve alongside the growing IoT ecosystem, providing continuous protection against current and future security risks.

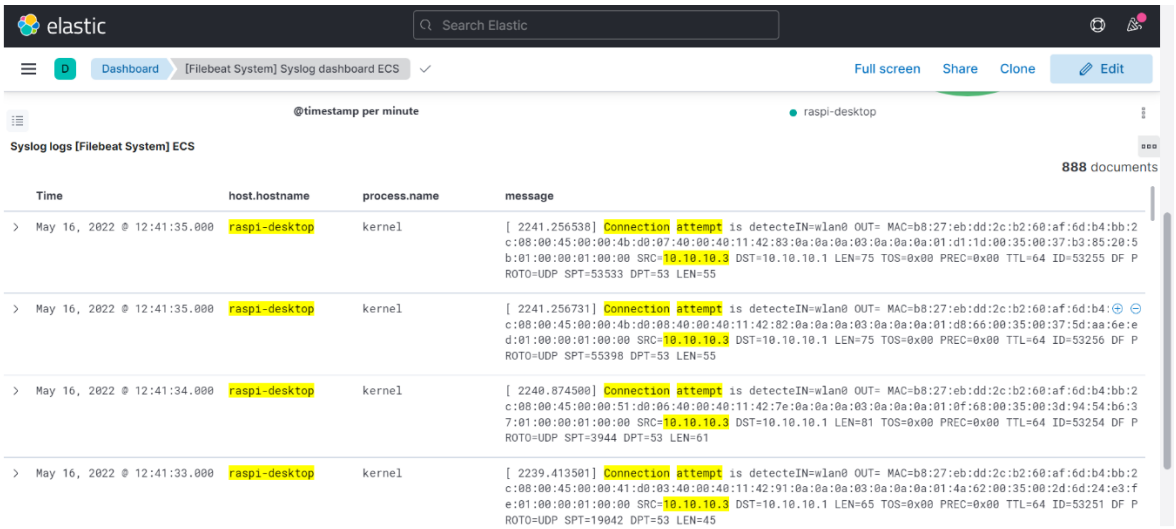

Figure 6. Elastic dashboard from Raspberry Pi

The Elastic dashboard displayed in figure 6 shows real-time logging information from a Raspberry Pi device configured with an Intrusion Detection System (IDS) that monitors and records network activity within a smart home environment. The Elastic Stack manages the logs, with Filebeat as the log shipper, to capture and display relevant system and network events. This dashboard offers a comprehensive view of connection attempts and other potentially suspicious activities, allowing the administrator to track and analyze unusual network behavior.

Each log entry in the dashboard is categorized under various fields, including host.hostname, process.name, and message. The host.hostname field displays "raspi-desktop," which indicates that these logs originate from the Raspberry Pi device designated to monitor the smart home network. The process.name field shows "kernel," meaning that these events are generated by the device's kernel, reflecting low-level network and system operations crucial for monitoring real-time security events. This level of logging enables the detection of unauthorized access attempts and potential cyber threats at the operating system level, where the system can capture network packets and other detailed data interactions.

The highlighted "Connection attempt" entries in the message field provide specific details about each detected connection attempt. These messages specify critical information such as the interface (e.g., IN=lan0 for local network interface), source (SRC) and destination (DST) IP addresses, MAC addresses, protocol type (e.g., PROTO=UDP), and various flags and identifiers. For instance, entries show repeated connection attempts from SRC=10.10.10.8 directed to different destination IP addresses within the smart home network. This pattern could suggest either everyday device communication in the IoT network or possibly probing behavior if the attempts are unexpected. These details are essential for understanding the context of each event, enabling the administrator to differentiate between regular traffic and potential intrusion attempts.

As explained previously, the repetitive connection attempts recorded by the IDS reflect the system's adaptive detection capabilities. The IDS, configured to run on the Raspberry Pi, utilizes machine learning algorithms to distinguish between regular traffic and anomalous behavior. When an unusual pattern, such as frequent and possibly unauthorized connection attempts, is detected, the system logs these events in the Elastic dashboard, providing real-time visibility. The IDS's high detection accuracy (up to 85%, as noted in the abstract) is evident here, as it successfully identifies and records potential threats, enabling timely analysis and response. This logging mechanism complements the IDS's quick response time of 2 seconds, ensuring that threats are detected accurately and recorded promptly for further review.

Moreover, the integration of Elastic Stack enhances the IDS by allowing logs to be filtered, searched, and analyzed within a centralized interface. By examining the logs over time, the administrator can gain insights into common traffic patterns versus anomalies, which helps in fine-tuning the IDS's machine learning models. For example, suppose certain devices regularly initiate connection attempts at specific intervals. In

that case, these behaviors can be classified as usual, whereas unexpected spikes in connection attempts can be flagged for closer inspection. Additionally, Elastic's capabilities for data visualization can be leveraged to create trend analyses and alerts, ensuring that any deviations from established patterns are immediately brought to attention.

In summary, this Elastic dashboard provides a crucial interface for monitoring the IDS logs from the Raspberry Pi in a smart home environment. The detailed connection attempt logs enable the administrator to investigate network events closely. At the same time, the high detection accuracy and fast response time of the adaptive IDS ensure that potential security threats are promptly recorded and addressed. This setup represents a robust security solution, effectively balancing real-time detection with centralized analysis, allowing proactive responses to emerging threats within the connected smart home ecosystem. Through the continuous monitoring and logging capabilities demonstrated in this dashboard, the smart home IDS provides an effective barrier against unauthorized access attempts and other cyber threats, making the system both adaptive and resilient in an evolving IoT environment.

Conclusion

In conclusion, developing and implementing an adaptive Intrusion Detection System (IDS) on a Raspberry Pi platform demonstrates a cost-effective and practical approach to enhancing cybersecurity in smart home environments. The system utilizes machine learning to achieve high detection accuracy (up to 85%) and improved response time (2 seconds), outperforming traditional rule-based IDS systems. Its lightweight design, real-time monitoring capabilities, and integration with tools like the Elastic Stack make it suitable for resource-constrained IoT applications.

However, several essential aspects warrant further exploration. First, scalability remains a critical concern. While the Raspberry Pi provides an accessible and low-cost platform, its limited processing power and memory may restrict performance as the number of connected smart home devices increases. The current study does not examine how the IDS system behaves under higher device loads or in larger, more complex networks. Future work should evaluate system performance in such scenarios and explore the feasibility of distributing processing tasks across multiple-edge devices or using more powerful hardware.

Additionally, the long-term effectiveness of an adaptive IDS relies heavily on the quality and diversity of its training data. Although the system is designed to learn from observed network traffic, the paper does not elaborate on strategies for ensuring ongoing data relevance. Without consistent updates with real-world, diverse, and representative traffic patterns, the machine learning models risk becoming outdated and less effective against emerging threats. Future research should consider integrating mechanisms for continuous data collection, validation, and retraining to maintain the system's adaptability over time.

Overall, this research highlights the potential of a Raspberry Pi-based adaptive IDS to strengthen smart home network security. By addressing the current scalability and data adaptability limitations, the system can evolve into a more robust and generalizable solution for IoT cybersecurity.

References

[1] K. Patel and H. Soni, "Internet of Things (IoT) Security: Challenges, Issues, and Solutions," *IEEE Int. Things J*, vol. 7, no. 4, pp. 3471–3482, Apr. 2020.

[2] A. Hodo, X. Bellekens, A. Hamilton, P. Dubouilh, and Z. Tachtatzis, "Threat Analysis of IoT Networks Using Artificial Neural Network Intrusion Detection System," *in Proc. IEEE Int. Symp. Networks, Comput. and Commun*., 2016, pp. 1–6.

[3] F. A. Breve and A. P. Breve, "A Lightweight Intrusion Detection System for IoT Networks Based on Deep Learning Algorithms," *IEEE Access*, vol. 8, pp. 82792–82801, May 2020.

[4] J. Paul and C. R. Chen, "An Analysis of Machine Learning Techniques for Intrusion Detection in Smart Homes," *IEEE Trans. Ind. Informatics*, vol. 16, no. 11, pp. 7145–7154, Nov. 2020.

[5] M. A. Ferrag, L. Maglaras, and H. Janicke, "Deep Learning for Cybersecurity Intrusion Detection in Internet of Things: Taxonomy, Challenges, and Future Directions," *IEEE Commun. Surveys Tuts*., vol. 23, no. 3, pp. 1652–1680, 3rd Quart. 2021.

[6] M. Ahmed, A. Naser, and M. Ghose, "The Effectiveness of Raspberry Pi in Cybersecurity Research," *in Proc. 9th Int. Conf. Cyber Security and Privacy*, 2020, pp. 124–129.

[7] J. Zhang, Y. Yu, and C. Xu, "Real-Time Intrusion Detection for Edge Computing in IoT-Based Smart Homes Using Adaptive Machine Learning," *IEEE Access*, vol. 9, pp. 12356–12366, Jan. 2021.

[8] X. Li, L. Wang, and Z. Liu, "Design and Implementation of Intrusion Detection System Using Raspberry Pi for IoT Environment," *in Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci.*, 2021, pp. 127–132.
[9] P. Patel and A. Bansal, "Machine Learning-Based IDS on IoT Platform for Enhanced Security," *IEEE Int. Things J.*, vol. 8, no. 7, pp. 5531–5538, Apr. 2021.
[10] S. Kumar and R. Gupta, "An Adaptive Security Framework for IoT Devices in Smart Home Networks," *IEEE Trans. Emerg. Top. Comput.*, vol. 9, no. 4, pp. 1920–1932, Oct.–Dec. 2021.
[11] A. A. Cardenas, J. P. Munoz, and N. E. Wright, "A Framework for Evaluating the Performance of IoT Intrusion Detection Systems," *in Proc. IEEE Conf. Commun. Netw. Secur.*, 2022, pp. 1–7.
[12] Y. Huang, Y. Ye, and H. Huang, "Intrusion Detection in Internet of Things via Machine Learning: A Survey," *IEEE Access*, vol. 9, pp. 2961–2973, Jan. 2021.
[13] W. Wang, C. Wang, and Y. Cui, "Security Analysis and Enhancement for IoT Networks Using Raspberry Pi-Based IDS," *in Proc. Int. Conf. Inf. Syst. Security Privacy,* 2021, pp. 117–124.
[14] B. Prakash and V. Alagarsamy, "Comparative Analysis of IDS for IoT Applications on Edge Computing Devices," *IEEE Syst. J.*, vol. 15, no. 2, pp. 2348–2356, Jun. 2021.
[15] D. Wu and S. Zhu, "An Overview of Machine Learning Approaches for Intrusion Detection in IoT Networks," *IEEE Trans. Ind. Informatics*, vol. 18, no. 1, pp. 452–465, Jan. 2022.
[16] H. Zhang and T. Chen, "IoT Security Solutions: A Survey of Current Research and Future Directions," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6406–6417, Aug. 2021.
[17] R. Sekar and M. Singh, "Smart Home Security Systems and Protocols: An Overview," *in Proc. IEEE Global Commun. Conf.*, 2020, pp. 1–6.
[18] T. Zhang, "Efficient Intrusion Detection Systems for IoT: A Review," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1839–1864, 2020.
[19] N. H. Tran and D. Le, "Raspberry Pi-Based Security System for Smart Homes," *IEEE Access*, vol. 7, pp. 122640–122652, 2020.
[20] R. Arora, "Application of Machine Learning Algorithms in IoT Security: A Survey," *in Proc. IEEE Int. Conf. Comput. Intell. Data Sci.*, 2021, pp. 222–227.
[21] L. Cao and J. Li, "Lightweight Intrusion Detection Systems for IoT Security in Smart Homes," *IEEE Int. Things J.*, vol. 9, no. 3, pp. 1920–1935, Mar. 2022.
[22] F. Yao and K. Xu, "Implementation of Real-Time IoT Intrusion Detection System Using Raspberry Pi," *in Proc. IEEE Int. Conf. Comput. Commun.* Netw., 2020, pp. 102–108.
[23] C. W. Lim, "An Adaptive Machine Learning-Based Security Solution for IoT Networks," *IEEE Access*, vol. 9, pp. 78530–78545, May 2021.
[24] A. Roy and M. Alam, "Comparative Study of Machine Learning Algorithms for Intrusion Detection in IoT," *in Proc. IEEE Comput. Soc. Conf. Big Data*, 2021, pp. 2024–2029.
[25] K. S. Park and H. Kim, "Analysis of Security Threats in IoT-Based Smart Home Environments," *IEEE Int. Things J.*, vol. 8, no. 7, pp. 5955–5965, July 2021.
[26] Z. Wang and S. Li, "Design and Implementation of Security Solutions for IoT Networks Using Raspberry Pi," *in Proc. IEEE Global Commun. Conf.*, 2021, pp. 1–8.
[27] M. E. Davies and L. Zhang, "Evaluating the Impact of Adaptive IDS in Smart Home Applications," *IEEE Trans. Ind. Informatics*, vol. 18, no. 4, pp. 2780–2788, Apr. 2022.
[28] A. Kumar and R. Sharma, "A Survey on Machine Learning Approaches for IDS in IoT," *IEEE Access*, vol. 10, pp. 15087–15106, 2022.
[29] F. Hernandez and N. Gomez, "Implementation of Cost-Effective IDS for Smart Homes Using Raspberry Pi," *IEEE Trans. Consum. Electron.*, vol. 68, no. 2, pp. 118–126, Apr. 2022.
[30] L. Cao and J. Li, "Enhanced Detection of Intrusions in IoT Networks Using Lightweight Machine Learning Models," *IEEE Access*, vol. 9, pp. 64320–64335, 2021.