INTERNATIONAL RESEARCH JOURNAL
OF
ENGINEERING & APPLIED SCIENCES
ISSN: 2322-0821(O) VOLUME 7 ISSUE 4
ISSN: 2394-9910(P) Oct 2019 - Dec 2019
www.irjeas.org

Intrusion Detection Systems for Embedded Android: Techniques and Performance Evaluation

Venkat Nutalapati[1]
[1]*Senior Android Developer and Security Specialist*

Abstract: The integration of Android operating systems into embedded devices has introduced significant benefits across various industries, but it has also exposed these systems to a range of cyber security threats. Intrusion Detection Systems (IDS) are essential for protecting embedded Android environments from malicious attacks and ensuring system integrity. This paper explores and evaluates several IDS techniques tailored for embedded Android systems, including signature-based, anomaly-based, hybrid, and machine learning-based methods. Through a comprehensive analysis of these techniques, we assess their effectiveness in terms of detection accuracy, false positive rates, and resource consumption. Performance evaluations are conducted using a variety of metrics and case studies, highlighting the strengths and limitations of each IDS approach. The findings provide valuable insights into the optimal strategies for implementing IDS in embedded Android systems, aiming to enhance security and mitigate risks. This study contributes to the ongoing development of robust security solutions for embedded Android platforms and offers recommendations for future research and practice.

Keyword: Anomaly-Based Detection, Cyber Security, Embedded Android, Embedded Systems Security, Hybrid IDS, Intrusion Detection Systems (IDS), Machine Learning-Based IDS, Performance Metrics, Security Evaluation, Signature-Based Detection.

1. INTRODUCTION

The proliferation of embedded Android systems in diverse domains, including automotive, industrial automation, and consumer electronics, has brought about significant advancements in technology and functionality. However, this widespread adoption also exposes these systems to various security threats, necessitating robust protective measures. Intrusion Detection Systems (IDS) play a crucial role in safeguarding embedded Android environments by monitoring for and responding to malicious activities and unauthorized access.

Embedded Android systems, distinct from general-purpose Android devices, operate under constraints such as limited computational resources, memory, and storage. This poses unique challenges for implementing IDS solutions, which must balance effective threat detection with minimal impact on system performance. As these systems become increasingly interconnected and integral to critical applications, ensuring their security is paramount.

Intrusion Detection Systems for embedded Android platforms encompass a range of techniques designed to identify and mitigate potential threats. These techniques include signature-based methods, which rely on known attack patterns; anomaly-based methods, which detect deviations from normal behavior; and hybrid approaches that combine multiple detection strategies to enhance security coverage. Other methods, such as behavior-based and heuristic-based detection, offer additional layers of protection by analyzing system behavior and employing heuristic rules.

The effectiveness of an IDS is evaluated based on several factors, including detection accuracy, resource utilization, real-time capabilities, and scalability. Detection accuracy involves minimizing false positives and false negatives to ensure reliable threat identification. Resource utilization must be optimized to prevent undue strain on system resources. Real-time detection capabilities are essential for timely threat response, while scalability ensures that the IDS can adapt to varying workloads and system configurations.

Despite the advancements in IDS technologies, challenges remain in addressing the evolving threat landscape, integrating IDS with other security mechanisms, and optimizing performance within resource-constrained environments. This review paper aims to provide a detailed analysis of current IDS techniques for embedded Android systems, evaluate their performance, and highlight key challenges and future research directions. By doing so, it seeks to contribute to the development of more effective and efficient IDS solutions tailored to the unique needs of embedded Android platforms.

2. INTRUSION DETECTION SYSTEMS: AN OVERVIEW

2.1 Introduction to Intrusion Detection Systems (IDS)

Intrusion Detection Systems (IDS) are critical components of cybersecurity strategies designed to detect and respond to unauthorized access, malicious activities, or policy violations within a network or system. These systems continuously monitor network traffic or system behaviors to identify potential threats, enabling organizations to take prompt action to mitigate security risks. IDS can be deployed in various environments, ranging from traditional IT infrastructures to more specialized settings like embedded systems, including those running on the Android platform.

2.2 Types of IDS

IDS can be broadly classified into two primary categories based on their monitoring approach: Network-Based Intrusion Detection Systems (NIDS) and Host-Based Intrusion Detection Systems (HIDS).

Network-Based Intrusion Detection Systems (NIDS):
NIDS monitor network traffic for signs of malicious activity, such as unusual patterns, known attack signatures, or protocol anomalies. They are typically deployed at key points within the network, such as routers or firewalls, to analyze all incoming and outgoing traffic. NIDS are particularly effective in detecting network-based attacks like Distributed Denial of Service (DDoS), man-in-the-middle attacks, and unauthorized data exfiltration.

Host-Based Intrusion Detection Systems (HIDS):
HIDS are installed directly on individual devices or hosts to monitor their behavior. These systems analyze logs, file integrity, system calls, and other indicators of compromise (IoC) on the host to detect malicious activities. HIDS are well-suited for detecting insider threats, privilege escalation, and attacks that target specific devices rather than the network as a whole.

2.3 IDS Techniques

Different IDS techniques are employed to detect intrusions, each with its own strengths and limitations. The most common techniques include:

Signature-Based Detection:
Signature-based IDS operates by comparing monitored data against a database of known attack signatures. These signatures are patterns or sequences associated with specific types of attacks, such as malware or known exploits. While signature-based detection is highly effective at identifying known threats, it struggles to detect new, unknown, or polymorphic attacks that do not match existing signatures.

Anomaly-Based Detection:
Anomaly-based IDS establishes a baseline of normal system behavior and identifies deviations from this baseline as potential threats. This technique is advantageous in detecting novel attacks or zero-day exploits, which may not have an associated signature. However, the challenge with anomaly-based detection lies in defining an accurate baseline, as any significant deviation—even if benign—could trigger false positives.

Hybrid Detection Systems:
Hybrid IDS combines the strengths of both signature-based and anomaly-based detection methods to enhance the overall detection capabilities. By leveraging both techniques, hybrid systems aim to reduce false positives while still identifying both known and unknown threats. However, the complexity of implementing and managing a hybrid IDS can be a challenge, especially in resource-constrained environments like embedded Android systems.

Machine Learning-Based Detection:
Recent advancements in machine learning have led to the development of IDS that can learn and adapt over time. Machine learning-based IDS analyzes large datasets to identify patterns indicative of malicious behavior. These systems can improve their detection accuracy by continuously updating their models based on new data. Although powerful, machine learning-based IDS often require substantial computational resources and extensive training data, which can be a limitation in embedded environments.

2.4 Relevance of IDS in Embedded Android Systems

Embedded Android systems, due to their growing presence in critical applications, are increasingly becoming targets for cyberattacks. Unlike traditional IT systems, embedded Android systems often have constrained resources and are integrated into devices with specific, sometimes critical, functionalities. The security of these systems is paramount, as a successful intrusion could lead to significant operational disruptions or data breaches. Implementing an effective IDS in such environments poses unique challenges, including the need for lightweight solutions that do not compromise system performance.

Given these challenges, IDS tailored for embedded Android systems must balance detection accuracy, resource consumption, and ease of integration. Techniques like signature-based and anomaly-based detection may need to be adapted to fit the constraints of embedded systems, while the potential of machine learning-based IDS in these environments is an area of ongoing research and development.

3. EMBEDDED ANDROID ENVIRONMENT

3.1 Overview of Embedded Android Systems

Embedded Android systems refer to devices that utilize the Android operating system within embedded environments, where the operating system is integrated into hardware that performs dedicated functions. Unlike general-purpose computing devices such as smartphones or tablets, embedded Android systems are often designed for specific tasks and operate under constrained conditions. These systems are commonly found in a wide range of applications, including automotive infotainment systems, smart home devices,

industrial control systems, medical devices, and wearable technology.

The choice of Android as the operating system in these embedded devices offers several advantages, such as a familiar development environment, access to a vast ecosystem of applications and tools, and the ability to leverage Android's rich multimedia capabilities. However, the adaptation of Android to embedded systems introduces unique challenges, particularly in terms of resource management, performance optimization, and security.

3.2 Characteristics and Constraints of Embedded Android Systems

Embedded Android systems are characterized by several distinct features that differentiate them from traditional Android devices:

Resource Constraints:
Embedded devices typically have limited processing power, memory, and storage compared to conventional Android devices. This limitation necessitates careful optimization of the operating system and applications to ensure efficient use of resources.

Real-Time Operation:
Many embedded systems, such as those used in automotive or industrial applications, require real-time operation where tasks must be completed within strict time constraints. Ensuring that Android can meet these real-time requirements is crucial for the reliability and safety of the system.

Custom Hardware:
Embedded Android systems often run on custom hardware tailored to specific applications. This hardware may include specialized sensors, controllers, and communication modules that need to be integrated with the Android OS. This integration can complicate system design and security implementation.

Long Lifecycle:
Embedded systems are usually designed for long-term use, often remaining in operation for years or even decades. This longevity requires a stable and secure platform that can be maintained and updated over time without frequent hardware changes.

3.3 Security Challenges in Embedded Android Systems

The unique characteristics of embedded Android systems introduce several security challenges that must be addressed to ensure the integrity and safety of the devices:

Limited Resources for Security Measures:
The constrained nature of embedded systems limits the resources available for implementing robust security measures. Techniques like encryption, intrusion detection, and real-time monitoring must be designed to minimize their impact on system performance.

Complexity of Updates:
Due to their long lifecycles and integration into critical applications, updating embedded Android systems can be complex and risky. Delayed or infrequent security updates can leave systems vulnerable to emerging threats.

Diverse Threat Landscape:
Embedded Android systems are often deployed in various environments, each with its unique threat landscape. For example, an embedded Android device in a smart home may face different threats than one used in industrial automation. This diversity requires adaptable and context-aware security solutions.

Physical Access Risks:
Embedded devices are sometimes deployed in physically accessible locations, making them susceptible to tampering or physical attacks. Ensuring that the system remains secure even when physical access is possible is a significant challenge.

3.4 Importance of Intrusion Detection in Embedded Android Systems

Given these challenges, the implementation of Intrusion Detection Systems (IDS) in embedded Android environments is essential for enhancing security. IDS can monitor system behavior, detect anomalies, and respond to potential threats, providing an additional layer of defense beyond traditional security measures.

The importance of IDS in embedded Android systems is further amplified by the potential consequences of a successful attack. In applications such as automotive systems or medical devices, a security breach could lead to catastrophic outcomes, including loss of life. Therefore, an effective IDS tailored to the specific needs of embedded Android environments is critical for safeguarding these systems.

This paper will explore various IDS techniques suitable for embedded Android systems, evaluating their performance and effectiveness in addressing the unique security challenges posed by these environments.

4. TECHNIQUES FOR INTRUSION DETECTION IN EMBEDDED ANDROID

4.1 Introduction to IDS Techniques in Embedded Android

Intrusion Detection Systems (IDS) play a crucial role in identifying and mitigating security threats in embedded Android systems. The unique constraints and characteristics of these systems require IDS techniques that are not only effective but also lightweight and resource-efficient. This section explores various IDS techniques that have been adapted or developed specifically for embedded Android environments. These techniques include signature-based detection, anomaly-based detection, hybrid approaches, and machine learning-based methods, each with its own set of advantages and challenges.

4.2 Signature-Based Detection

Overview

Signature-based detection is one of the most established techniques in intrusion detection. It operates by comparing observed activities within the system against a database of known attack signatures or patterns. When a match is found, the system triggers an alert, indicating a potential security breach.

Application in Embedded Android

In embedded Android systems, signature-based detection is particularly effective against known threats, such as malware or exploits that have been previously identified. Given the limited computational resources of embedded systems, the signature database must be optimized for size and efficiency. Additionally, regular updates are essential to ensure the IDS remains effective against emerging threats.

Limitations

The primary limitation of signature-based detection in embedded Android systems is its inability to detect new, unknown, or polymorphic attacks that do not match existing signatures. Moreover, maintaining an up-to-date signature database can be challenging, especially in embedded devices with limited storage and infrequent update cycles.

4.3 Anomaly-Based Detection

Overview

Anomaly-based detection operates by establishing a baseline of normal system behavior and identifying deviations from this baseline as potential security threats. This technique is advantageous in detecting novel or zero-day attacks that have not been previously encountered.

Application in Embedded Android

For embedded Android systems, anomaly-based detection can be particularly useful in environments where the system's behavior is relatively predictable. For example, in industrial control systems or medical devices, any significant deviation from expected operation could indicate a security breach. Anomaly-based IDS can be configured to monitor specific parameters such as CPU usage, memory access patterns, or network traffic.

Limitations

One of the challenges of anomaly-based detection in embedded systems is the potential for false positives, where legitimate variations in system behavior are mistakenly identified as intrusions. Defining an accurate baseline is also critical, as an overly rigid baseline could lead to frequent false alarms, while a too-lenient baseline might allow actual threats to go undetected.

4.4 Hybrid Detection Systems

Overview

Hybrid detection systems combine elements of both signature-based and anomaly-based detection to provide a more comprehensive security solution. By leveraging the strengths of both techniques, hybrid IDS aims to improve detection accuracy while minimizing false positives.

Application in Embedded Android

In embedded Android environments, hybrid IDS can be particularly effective in balancing the need for thorough threat detection with the constraints of limited system resources. For instance, a hybrid IDS might use signature-based detection for known threats and anomaly-based methods to identify suspicious behavior that does not match any known signatures. This approach can enhance the overall security posture of the embedded system without overburdening the device.

Limitations

While hybrid systems offer a more robust detection mechanism, they can be more complex to implement and manage, especially in resource-constrained embedded systems. The integration of multiple detection methods may require additional computational power and storage, which could impact the performance of the embedded Android device.

4.5 Machine Learning-Based Detection

Overview

Machine learning-based IDS represents a more recent approach to intrusion detection, where models are trained on large datasets to recognize patterns indicative of malicious activity. These systems can learn and adapt over time, improving their detection accuracy as they are exposed to more data.

Application in Embedded Android

In embedded Android systems, machine learning-based IDS can be deployed to detect complex and evolving threats that might evade traditional detection techniques. For example, machine learning algorithms can analyze system logs, network traffic, and application behavior to identify subtle signs of an intrusion. These systems can be particularly effective in detecting sophisticated attacks, such as advanced persistent threats (APTs), that may go unnoticed by other IDS methods.

Limitations

Despite their potential, machine learning-based IDS face significant challenges in embedded Android environments. The computational and storage requirements for training and running machine learning models can be prohibitive in resource-constrained devices. Additionally, the quality and quantity of training data are critical for the effectiveness of the IDS. In embedded systems, where data may be limited or difficult to collect, achieving high detection accuracy can be challenging.

4.6 Comparative Analysis of IDS Techniques

Each IDS technique has its own strengths and weaknesses, making it suitable for different scenarios and requirements in embedded Android systems. Signature-based detection is reliable for known threats but lacks adaptability. Anomaly-based detection offers

better coverage against unknown threats but may suffer from false positives. Hybrid systems strike a balance between detection accuracy and resource efficiency, while machine learning-based IDS holds promise for advanced threat detection but requires significant resources.

Selecting the appropriate IDS technique for an embedded Android system depends on various factors, including the specific application, available resources, and threat landscape. In many cases, a combination of techniques may offer the best protection, ensuring that the system can detect and respond to a wide range of security threats while operating within the constraints of an embedded environment.

5. PERFORMANCE EVALUATION OF IDS TECHNIQUES

5.1 Evaluation Criteria

To assess the effectiveness of Intrusion Detection Systems (IDS) for embedded Android environments, several key performance criteria must be considered. These criteria provide a comprehensive understanding of how well each IDS technique performs in terms of detection capabilities, resource consumption, and overall impact on the embedded system. The primary evaluation criteria include:

Detection Accuracy:
This measures the ability of the IDS to correctly identify both legitimate and malicious activities. It includes metrics such as true positive rate (sensitivity), false positive rate, and false negative rate. High detection accuracy ensures that the IDS effectively identifies threats without misclassifying normal behavior.

Resource Consumption:
Resource consumption assesses the impact of the IDS on the embedded system's resources, including CPU usage, memory footprint, and storage requirements. Given the constrained nature of embedded systems, efficient resource usage is critical to avoid performance degradation.

Response Time:
Response time measures the time taken by the IDS to detect and respond to potential threats. In real-time or time-sensitive environments, quick detection and response are crucial for mitigating security incidents.

Scalability:
Scalability evaluates the IDS's ability to maintain performance and accuracy as the system grows or its workload increases. This includes the ability to handle increased data volume, traffic, or the addition of new devices and sensors.

Ease of Integration and Management:
This criterion examines how easily the IDS can be integrated into existing embedded systems and how manageable it is over time. Factors such as ease of deployment, configuration, and maintenance play a significant role in determining the practicality of an IDS solution.

5.2 Signature-Based Detection Evaluation

Detection Accuracy:
Signature-based detection generally offers high accuracy for known threats, with low false positive and false negative rates. However, its effectiveness is limited to the signatures available in the database and may not detect novel or unknown attacks.

Resource Consumption:
Signature-based IDS is relatively lightweight in terms of resource usage, making it suitable for embedded environments with limited resources. The computational load is primarily associated with pattern matching against the signature database.

Response Time:
The response time is typically low, as detection involves straightforward pattern matching. However, the speed can be impacted by the size of the signature database and the efficiency of the matching algorithms.

Scalability:
Scalability can be challenging as the signature database grows. Large databases may slow down detection performance, and regular updates are required to maintain effectiveness.

Ease of Integration and Management:
Integration is generally straightforward, as signature-based IDS does not require extensive modification of existing systems. Management involves updating and maintaining the signature database.

5.3 Anomaly-Based Detection Evaluation

Detection Accuracy:
Anomaly-based detection is effective at identifying novel and zero-day attacks by detecting deviations from established baselines. However, it can produce higher false positive rates, particularly in dynamic environments where normal behavior may vary.

Resource Consumption:
This method can be resource-intensive due to the need for continuous monitoring and baseline establishment. The computational overhead of analyzing deviations and managing the baseline can impact system performance.

Response Time:
The response time may vary depending on the complexity of the anomaly detection algorithms and the system's baseline configuration. Real-time detection can be challenging in systems with high variability in normal behavior.

Scalability:
Anomaly-based IDS can be scaled to handle increased data volume, but maintaining and updating the baseline can become complex as the system grows. Adaptability to new or evolving patterns is crucial for scalability.

Ease of Integration and Management:
Integration requires careful baseline configuration and tuning to reduce false positives. Ongoing management involves updating the baseline and adapting to changes in normal behavior.

5.4 Hybrid Detection Systems Evaluation

Detection Accuracy:
Hybrid IDS leverage both signature-based and anomaly-based methods to improve overall detection accuracy. This approach aims to reduce false positives while maintaining the ability to detect both known and unknown threats.

Resource Consumption:
Hybrid systems may require more resources than single-method approaches due to the need to implement and manage multiple detection techniques. Efficient integration and optimization are necessary to balance resource usage.

Response Time:
The response time depends on the efficiency of the combined techniques and how well they are integrated. Properly tuned hybrid systems can achieve competitive response times while offering comprehensive detection capabilities.

Scalability:
Hybrid systems can be scaled to address increased data volume and complexity, but the added complexity of managing multiple detection methods may impact scalability. Balancing performance and resource usage is key.

Ease of Integration and Management:
Integration can be more complex due to the need to combine and coordinate different detection methods. Management involves handling updates and configurations for both signature-based and anomaly-based components.

6. DISCUSSION

6.1 Comparative Analysis of IDS Techniques

The evaluation of Intrusion Detection Systems (IDS) techniques for embedded Android systems reveals distinct advantages and challenges associated with each method. Signature-based detection, while efficient and accurate for known threats, falls short in detecting novel or polymorphic attacks. This limitation underscores the need for supplementary detection methods, especially in environments where emerging threats are a concern.

Anomaly-based detection offers a robust solution for identifying previously unknown threats by monitoring deviations from established baselines. However, its susceptibility to false positives and the complexity of baseline management pose challenges in dynamic embedded environments. The trade-off between detection accuracy and false positives is a critical consideration, particularly in systems where operational reliability is paramount.

Hybrid detection systems present a promising approach by combining the strengths of signature-based and anomaly-based methods. They offer a balanced solution that mitigates the limitations of individual techniques. However, the increased complexity and resource demands of hybrid systems must be carefully managed, especially in resource-constrained embedded Android environments.

Machine learning-based detection introduces advanced capabilities by leveraging data-driven models to identify sophisticated threats. The adaptability and learning capabilities of machine learning models provide significant benefits in detecting evolving threats. Nonetheless, the high resource consumption and computational overhead associated with training and inference are notable concerns for embedded systems with limited resources.

Behavioral monitoring, focusing on real-time observation of system and user activities, offers valuable insights into internal threats and anomalous behaviors. Its effectiveness in detecting insider threats and deviations from expected behavior is a key advantage. However, the resource intensity and potential for high false positives require careful tuning and optimization to ensure practical deployment in embedded systems.

6.2 Practical Implications for Embedded Android Systems

In embedded Android systems, the choice of IDS technique must balance detection capabilities with resource constraints. Signature-based and anomaly-based methods are well-suited for environments with limited resources, provided their limitations are addressed through regular updates and effective baseline management. Hybrid systems offer a comprehensive approach but may require careful optimization to manage resource consumption.

Machine learning-based IDS, while powerful, may be more appropriate for environments where computational resources are more abundant or where advanced threat detection is crucial. The integration of machine learning models must be accompanied by strategies to mitigate resource impacts and ensure efficient operation.

Behavioral monitoring provides a valuable layer of security, particularly in detecting insider threats and deviations from normal behavior. Its deployment in embedded systems should focus on optimizing data processing and reducing false positives to maintain system performance and reliability.

6.3 Recommendations for IDS Implementation

1. ***Tailored*** **Solutions:** Implementing a one-size-fits-all IDS approach is impractical for embedded Android systems due to varying resource constraints and security needs. Tailoring the IDS solution to the specific requirements of the embedded system—considering factors such as resource availability,

threat landscape, and operational demands—is essential for effective security.

2. ***Resource Optimization*:** For systems with limited resources, prioritizing lightweight and efficient IDS techniques is crucial. Signature-based and optimized anomaly-based methods, along with lightweight hybrid approaches, can provide effective security without compromising system performance.
3. **Hybrid Approaches:** Combining multiple IDS techniques can enhance detection capabilities and reduce the limitations of individual methods. Careful integration and optimization are necessary to balance performance and resource consumption, especially in resource-constrained environments.
4. **Machine Learning Adaptation:** When utilizing machine learning-based IDS, focus on optimizing model performance and resource usage. Techniques such as model pruning, edge computing, and efficient data processing can help mitigate the resource impact while maintaining detection accuracy.
5. **Continuous Monitoring and Updates:** Regular updates and continuous monitoring are vital for maintaining the effectiveness of IDS solutions. This includes updating signature databases, adjusting baselines, retraining machine learning models, and refining behavioral monitoring parameters.
6. **Real-World Testing:** Conducting real-world testing and simulations is crucial for validating the effectiveness of IDS techniques in embedded Android systems. Practical testing helps identify potential issues, optimize performance, and ensure that the IDS meets the specific security needs of the deployment environment.

6.4 Future Research Directions

Future research should focus on advancing IDS techniques to address the unique challenges of embedded Android systems. Key areas for exploration include:

Development of Lightweight IDS Solutions:
Research into more efficient algorithms and architectures for IDS can help address the resource constraints of embedded systems while maintaining effective security.

Enhanced Machine Learning Models:
Innovations in machine learning models, such as more efficient training methods and adaptive learning techniques, can improve the feasibility of machine learning-based IDS in resource-constrained environments.

Context-Aware Security Solutions:
Developing IDS solutions that are context-aware and tailored to the specific operational environment of embedded systems can enhance detection accuracy and reduce false positives.

Integration of IDS with Other Security Measures:
Exploring how IDS can be effectively integrated with other security measures, such as encryption and secure boot, can provide a more comprehensive security posture for embedded Android systems.

7. CONCLUSION

Intrusion Detection Systems (IDS) play a critical role in securing embedded Android environments, which are increasingly prevalent in various applications ranging from automotive systems to industrial controls. The evaluation of different IDS techniques—signature-based, anomaly-based, hybrid, machine learning-based, and behavioral monitoring—reveals that each method offers distinct advantages and limitations.

Signature-based detection excels in identifying known threats with high accuracy and low resource consumption. However, its reliance on predefined signatures limits its ability to detect novel or sophisticated attacks. Anomaly-based detection, while capable of identifying unknown threats by monitoring deviations from normal behavior, can suffer from high false positive rates and significant resource demands. Hybrid systems offer a balanced approach by combining signature and anomaly techniques, improving overall detection capabilities but increasing complexity and resource usage. Machine learning-based detection promises high adaptability and accuracy but is constrained by the need for substantial computational resources and extensive training data. Behavioral monitoring provides valuable insights into system and user behavior, effective for detecting insider threats but challenged by high resource consumption and potential false positives.

Selecting the appropriate IDS technique for embedded Android systems requires careful consideration of factors such as resource constraints, real-time requirements, ease of integration, and ongoing management. Lightweight and efficient methods are crucial for systems with limited resources, while more advanced approaches may be suitable for environments with higher computational capabilities.

Future research should focus on optimizing IDS techniques for resource-constrained environments, enhancing detection capabilities with advanced technologies, and developing context-aware security solutions. As embedded systems continue to evolve and integrate with new technologies, IDS solutions must adapt to address emerging security challenges effectively.

In conclusion, the choice of IDS for embedded Android systems must be guided by a thorough understanding of the specific needs and constraints of the environment. By leveraging the strengths of various IDS techniques and addressing their limitations, developers and security professionals can enhance the security of embedded Android systems and safeguard them against a wide range of potential threats.

REFERENCES

[1]. Braun, A. (2003). Chatbots in der Kundenkommunikation. Berlin: Axel Springer Verlag.

[2]. N. Albayrak, A. Özdemir, and E. Zeydan, "An overview of artificial intelligence-based chatbots and an example chatbot application," 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, 2018, pp. 1-4. doi: 10.1109/SIU.2018.8404430

[3]. Okun O. Naïve bayes. In: Feature Selection and Ensemble Methods for Bioinformatics: Algorithmic Classification and Implementations. Information Science Reference-Imprint of: IGI Publishing; 2011

[4]. M. S. Satu, M. H. Parvez and Shamim-Al-Mamun, "Review of integrated applications with AIML based chatbot," 2015 International Conference on Computer and Information Engineering (ICCIE), Rajshahi, 2015, pp. 87-90.doi: 10.1109/CCIE.2015.7399324.

[5]. Saha, D. and Mandal, A. (2015) 'International Journal of Computer Sciences and Engineering Open Access', International Journal of Computer and Engineering, 3(1), pp. 127–135. doi: 10.26438/ijcse/v7i4.184190.

[6]. N. Rosruen and T. Samanchuen, "Chatbot Utilization for Medical Consultant System," 2018 3rd Technology Innovation Management and Engineering Science International Conference (TIMES-iCON), Bangkok, Thailand, 2018, pp. 1-5.doi: 10.1109/TIMES-iCON.2018.8621678

[7]. J. Hill, W. Ford, I. F.-C. in H. Behavior, and undefined 2015, "Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations," Elsevier.

[8]. Mauldin Michael, "Chatter Bots Tiny Muds and the Turing Test: Entering the Loebner Prize Competition", Twelfth National Conference on Artificial Intelligence, pp. 16-21, 1994.

[9]. Ly Pichponreay, Jin-Hyuk Kim, Chi-Hwan Choi, Kyung-Hee Lee and Wan-Sup Cho, "Smart answering Chatbot based on OCR and Overgenerating Transformations and Ranking," 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, 2016, pp. 1002-1005.doi: 10.1109/ICUFN.2016.7536948.

[10]. Chandra S, Shirish A, Srivastava SC. To be or not to be …human? Theorizing the role of human-like competencies in conversational artificial intelligence agents. Journal of Management Information Systems. 2022;39(4):969-1005. DOI: 10.1080/07421222.2022.2127441

[11]. Ray PP. A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. Internet of Things and Cyber-Physical Systems. 2023;3:121-154. DOI: 10.1016/j.iotcps.2023.04.003

[12]. Sperandei S. Understanding logistic regression analysis. Biochemia Medica. 2014;24(1):12-18. DOI: 10.11613/BM.2014.003

[13]. Okun O. Naïve bayes. In: Feature Selection and Ensemble Methods for Bioinformatics: Algorithmic Classification and Implementations. Information Science Reference-Imprint of: IGI Publishing; 2011