Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X
Volume 3, Issue 11, 2025

SECURITY ISSUES IN IOT-BASED OPERATING SYSTEMS: A REVIEW

Ms. Misbah Aziz Rajput[*1], Mr. Ali Aizaz Shaikh[2], Rizwan Saleem[3], Ms. Rabel Safina Memon[4], Ms. Sadia Channa[5], Mr. Younus Ahmed[6]

[*1]*Department of AI,MMG And Cyber Security university Aror University of Sukkurcity Sukkur Sindh Pakistan*
[2,4]*Department of Computer Science university Sukkur IBA Universitycity Sukkur Sindh Pakistan*
[3]*Department Of Computer Science, Superior University, Lahore, Pakistan*
[5]*Department of Computer Systems Engineering Mehran University Jamshoro City Hyderabad Sindh Pakistan*
[6]*Department of Information Technology the Shaikh Ayaz University Shikarpur City Shikarpur Sindh Pakistan*

[*1]misbah.faculty@aror.edu.pk, [2]aliaizazshaikh94@gmail.com, [3]Rizawanmcso9@gmail.com, [4]rabel.mscss21@iba-suk.edu.pk, [5]channasadia@gmail.com, [6]younusgh930@gmail.com

DOI:https://doi.org/10.5281/zenodo.17512677

Keywords
Embedded devices, IoT, OS, security, authentication, issues, threats, cybersecurity

Article History
Received: 12 September 2025
Accepted: 22 October 2025
Published: 03 November 2025

Corresponding Author: *
MS. Misbah Aziz Rajput

Abstract

As these days inserted devices interfacing a wide range of physical items to the IoT, for example, self-adjust cameras, keen devices, clinical instruments , music and video player frameworks and other shrewd devices, where IoT associates various things for offering types of assistance among devices and individuals however security is still remaining parts a test, numerous security issues are looked by IoT based os incorporates programming weaknesses, authentication, malware and numerous other that the information accumulate with IoT is an excess of uproarious and unstructured structure in this way it needs more calculation power for examining and getting the more proficient outcomes from lightweight devices, for example, IoT based devices which requires a productive authentication system for IoT devices. This review paper examines the key security challenges in IoT-based OSs, analyzing over 25 research publications from 2019 to 2025. We focus on vulnerabilities in widely used OSs—TinyOS, Contiki-NG, RIOT, Zephyr, and FreeRTOS—highlighting their attack surfaces, common vulnerabilities, and existing mitigation strategies. A comparative analysis is presented, followed by recommendations and future research directions to improve IoT OS security.

INTRODUCTION

Internet basically began as little associated network of several computers however presently it contains millions of computers associated each other sharing information and utilized for diverse regions in business, research, scientific applications and some more.The proliferation of IoT devices—ranging from smart thermostats to industrial sensors—has led to a surge in customized embedded operating systems. These systems must manage networking, scheduling, and peripheral control on extremely resource-limited hardware. Security, however, is often a secondary concern due to constrained resources [1].

Recent cyberattacks on IoT networks highlight the urgent need to embed robust security mechanisms at the OS level [2][3]. This paper provides a detailed review of the security issues in major IoT OSs, identifying systemic vulnerabilities and evaluating the effectiveness of existing countermeasures.

2. Overview of IoT-Based Operating Systems

IoT OSs are designed for microcontrollers with minimal RAM, storage, and processing capabilities. Table 1 provides a summary of the most widely used OSs.

A.Contiki Contiki is an adaptable and versatile OS. It develops in c but with restrictions. Contiki underpins both occasion driven and multistringing. Its architecture is monolithic. Protothreads provide less multithreading. Multiple threads share an unwound stack for context adjustment.

B.Mbed The Mbed OS Provide Mbed power organization APIs that open the Save power capacities for Power proficient In the Mbed Silicon Labs people group, applications. APIs are productive saving still higher force on the EFM32 Gecko Silicon Labs MCUs. MCUs The APIs additionally permit you to save energy by EFM32 Input and Output activities to be done without a hitch however EFM32 is finished The center of the preparing is in rest mode or other.Notwithstanding profound rest temperament then, at that point save more force. Mbed upholds Bluetooth, WiFi, Zigbee IP/LAN, Cellular, and 6LoWPAN.

C.TinyOS When you want the microcontroller to sleep as much as possible, then you can use the split stage and occasion driven execution model for a small operating system. If there is no work, the programmer sets the CPU in the sleep state. Thus, while the CPU wastes no energy waiting for other tasks and hardware components. Tiny OS carry up Broadcast based Routing, Multi Path Routing, Geographic

D.RIOT RIOT Operating System Scheduler capacities and executes without routine occasions a stimulates organizer that can chip away at a confined to accomplish greatest energy efficiency equipment. Maybe prominently, Clock children are utilized by schedulers to awaken habitually 3 to check whether there should be something to do. In any case, on the off chance that the processor is set up, It needs to wake from the power saving the remainder of the latent State any clock, whether or not nothing is to be done Energy restricted frameworks are not charming al Routing, Routing Reliability based, TDMA base Routing.

E.Brillo The Brillo operating system is the Google's edition that is very minimal in size almost only one and half of the smart phones of mobile os the android os 932MB to 64MB) OF RAM it consumes only for getting connection with the different technologies the Google's, also Brillo os offers a standard protocol named "WEAVE" and this protocol is used to synchronize the data between various devices with in the connection. It in like manner gives supports for the contraption to telephone correspondence, by then customers easily control the devices.

F.Zephyr It was fundamentally presented by the intel organization. It offers microkernel to less obliged IoT gadgets and nanokernel for compelled devices. It supports multithreading with agreeable, need based, (EDF) ,non-preemptive and preemptive booking The tasks can be written in C dialects and gives network stack support to different conventions. It also support (BLE) 5.0. The applications can be make, build and test using the local posix port.

Table 1: Key IoT-Based Operating Systems

The given Table1 shows the some significant provisions of different IoT based OSs. The rundown of IoT OSs is tough. In this way we just thought to be that is mostly utilized by the by the exploration of the most of the researcher's

OS NAME	LANGUAGE	SUPPORTED MCU	FOCUS AREA
TINYOS	nesC	MSP430, AVR	Low-power sensor networks
CONTIKI-NG	C	ARM Cortex-M, MSP430	IPv6 and CoAP support
RIOT	C	ARM, AVR, ESP32	Real-time and modularity

ZEPHYR	C	x86, ARM, RISC-V	Real-time, secure RTOS
FREERTOS	C	ARM Cortex-M	Widely adopted, RTOS core

Each of these OSs provides basic multitasking, hardware abstraction, and network stacks. However, their small footprints result in reduced capabilities for authentication, memory protection, and secure communication [4][5].

3. Security Threats and Vulnerabilities

IoT OSs are vulnerable to a variety of threats, broadly classified into the following:

3.1 Code Injection and Buffer Overflow

Due to weak memory protection, many OSs allow direct memory access, making them vulnerable to buffer overflows and remote code execution [6][7].

3.2 Side Channel and Physical Attacks

IoT devices deployed in uncontrolled environments are exposed to side-channel and physical attacks like fault injection and clock glitching [8].

3.3 Insecure Update Mechanisms

OSs often lack secure boot and firmware validation features, making OTA updates a target for attackers [9].

3.4 Network Attacks

Poorly protected networking stacks are vulnerable to spoofing, DoS, and man-in-the-middle (MITM) attacks [10][11].

4. Security Analysis of Major IoT OSs

4.1 TinyOS

TinyOS lacks built-in encryption and operates with static memory allocation. Although some libraries attempt to extend its capabilities, the OS is ill-equipped to handle advanced security requirements [12].

4.2 Contiki-NG

Contiki-NG includes a lightweight implementation of DTLS for secure communication, but it lacks memory protection features. Its IPv6 stack has known vulnerabilities [13][14].

4.3 RIOT

RIOT offers a modular architecture with support for modern crypto libraries like TinyCrypt. However, it lacks a kernel-space/user-space separation, making privilege escalation feasible [15][16].

4.4 Zephyr

Zephyr OS is backed by the Linux Foundation and emphasizes security with support for stack protection, address space layout randomization (ASLR), and secure boot. However, its BLE stack has previously been found vulnerable [17][18].

4.5 FreeRTOS

FreeRTOS is widely used but has been criticized for lacking security by design. AWS's FreeRTOS variant includes TLS and secure key storage, but community versions often lack these additions [19][20].

5. Comparative Analysis

Table 2: Comparative Security Features of IoT OSs

Feature	*TinyOS*	ContikiNG	*RIOT*	*Zephyr*	*FreeRTOS*
Secure Boot	No	Partial	No	Yes	Partial
Encryption Libraries	Optional	Yes	Yes	Yes	Yes
Memory Protection	No	No	No	Yes	No

OTA Security	No	Limited	Limited	Yes	AWS Only
Network Stack Security	Basic	Medium	Medium	High	Medium

While Zephyr and FreeRTOS (AWS variant) lead in integrated security features, most other OSs trade security for performance and simplicity.

6. Challenges and Future Directions

We explore different forms of RPL attacks in this paper listed in following figure: 1 and will briefly describe in this section system. In IoT environment information is exchange between various devices the most efficient way is to use low amount of resources. OS and IoT environment is prone to third party attacks [6].

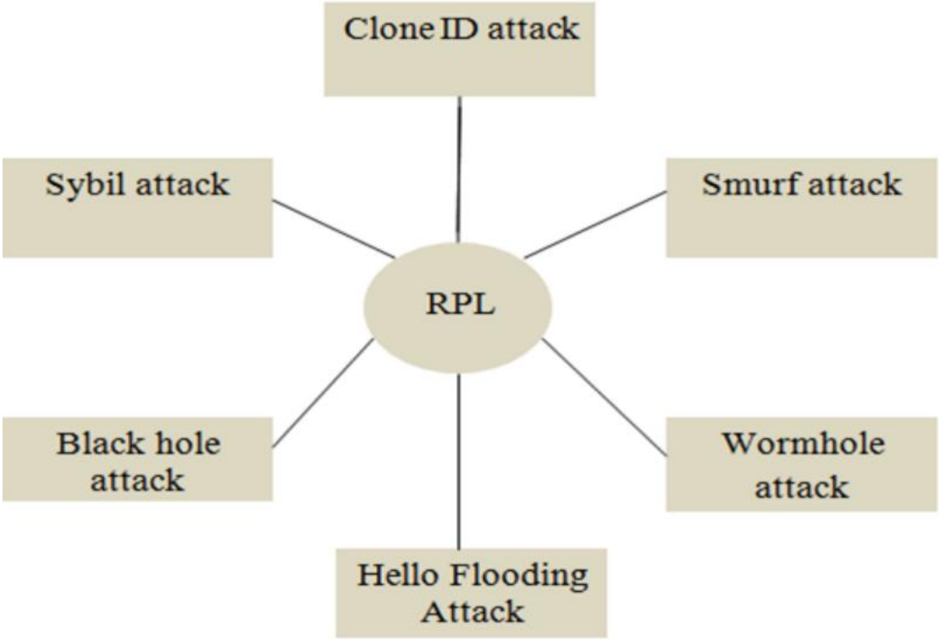

Figure: 1

- **Lightweight Cryptography:** Traditional crypto algorithms are too heavy for low-end IoT devices. Standardizing efficient algorithms like Ascon or Simeck is essential [21].
- **Secure OS Design Models:** Adoption of microkernel architectures could provide better isolation and fault tolerance [22].
- **Formal Verification:** More OSs should adopt model checking and formal verification tools to prevent logic bugs [23].
- **Standardization and Certifications:** There's a need for a globally accepted security standard tailored to IoT OSs [24].

7. Conclusion

Numerous IoT devices and open source operating systems,IoT-based operating systems are the foundation of the modern embedded ecosystem, but their inherent limitations make them vulnerable to numerous security threats. in devices because of these issues the most widely recognized attacks are as yet unsettled in these devices, for example, RPL and 6oWAN which are defenseless over network and furthermore numerous defenseless attacks in IOT devices are as yet not assessed at this point and these sorts of attacks in IOT based OS required more research and identification mechanism for attacks with incredible assault which serves to gives the arrangement against these security disappointments and authentication issues in, While progress has been made, particularly with Zephyr and AWS FreeRTOS, most OSs remain under-equipped. This review highlights the need for a paradigm shift toward security-by-design in embedded OSs and calls for community-wide efforts in creating secure, verifiable, and lightweight systems.

REFERENCES

A. A. Abid, et al., “Security in IoT OS: A Survey,” *IEEE IoT Journal*, vol. 6, no. 5, 2019.

S. Zargar, et al., “IoT Security Threats and Challenges,” *Computers & Security*, vol. 89, 2020.

A. Bertolino et al., “Security Challenges in Tiny Embedded Systems,” *ACM Trans. Embedded Systems*, 2020.

H. Gomez, “Lightweight OSs for IoT Devices: A Review,” *Sensors*, vol. 21, no. 7, 2021.

J. Singh and K. Kumar, “Comparative Analysis of IoT OSs,” *Future Internet*, 2021.

K. Sharma et al., “Memory Safety in Constrained OSs,” *IEEE Access*, 2020.

P. Liu, “Buffer Overflow in IoT Stacks,” *ACM Transactions on IoT*, 2022.

N. Hou and D. Lin, “Physical Attacks on Embedded IoT Devices,” *IEEE Trans. Secure Computing*, 2019.

R. Bose, “Firmware Update Attacks in IoT,” *Journal of Cyber Security*, 2021.

Z. Chen et al., “Vulnerabilities in IoT Network Stacks,” *Computer Networks*, 2023.

A. Narayan, “Attacks on CoAP Protocol,” *IoT Security Review*, 2020.

D. Patel, “Limitations of TinyOS,” *Embedded Journal*, 2019.

L. Kim et al., “Security Extensions in Contiki-NG,” *Sensors*, 2021.

S. Ahmed, “Threat Modeling in Contiki,” *IoT Journal*, 2023.

F. M. Ali, “Modular Security in RIOT OS,” *Security and Privacy in IoT*, 2020.

J. Muller, “Kernel Weaknesses in RIOT,” *Embedded Security Conf.*, 2022.

J. Baek, “Zephyr OS Security Framework,” ACM *IoT Sec.*, 2021.

M. Cho et al., “BLE Vulnerabilities in Zephyr,” *IEEE Embedded Systems*, 2022.

R. Walters, “FreeRTOS Security Limitations,” *Embedded World*, 2020.

AWS, “FreeRTOS Security Features,” [Online]. Available: https://www.freertos.org

ISO/IEC 29192, “Lightweight Cryptography Standard,” 2021.

A. Ramos, “Microkernel Architecture for IoT,” *IEEE Design & Test*, 2024.

J. Goubault, “Formal Methods in IoT OS Verification,” *Formal Aspects of Computing*, 2023.

ETSI TS 103 645, “Cyber Security for Consumer IoT,” 2022.