IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, April 2025
Impact Factor: 7.67

Antivirus System using Raspberry Pi Pico W For Windows

Prof. Pansare P. M[1], Kadambari A Ghadge[2], Omkar R Ghorpade[3], Vinit D Chaskar[4], Shravan N Chavan[5]

Professor, Department of Computer Science & Engineering[1]
Students, Department of Computer Science & Engineering[2-5]
Navsahyadri Education Society's Group of Institutions, Polytechnic, Pune, Maharashtra, India

Abstract: *In today's rapidly evolving digital environment, the proliferation of cyber threats necessitates the development of robust, efficient, and lightweight antivirus solutions, particularly for edge computing devices. This paper introduces a compact and cost-effective antivirus scanning system built on the Raspberry Pi Pico W platform, specifically engineered to detect and neutralize threats originating from USB storage devices and network-connected peripherals. Leveraging the capabilities of MicroPython for ease of development and Wi-Fi connectivity for remote access and updates, the proposed system integrates a signature-based malware detection mechanism to provide real-time scanning and threat identification. The lightweight nature of the Raspberry Pi Pico W ensures low power consumption and portability, making it ideal for deployment in educational environments, small-scale industrial setups, and edge networks. Furthermore, the system is designed with dynamic updating functionality, enabling automatic retrieval and integration of the latest malware signature databases via Wi-Fi, thereby enhancing its adaptability and resilience against emerging threats. Through practical testing and implementation, the prototype demonstrates an effective balance between performance, resource efficiency, and affordability, positioning it as a viable cybersecurity tool for constrained environments where traditional antivirus solutions may be impractical*

Keywords: Raspberry Pi Pico W, Antivirus, IoT Security, USB Scanning, Embedded Systems, Micro Python, Lightweight Detection, Threat Signature, Real-time Monitoring

I. INTRODUCTION

The proliferation of USB devices and IoT gadgets in both consumer and industrial domains has significantly expanded the attack surface for malware infections. USB drives, in particular, are frequently used across multiple systems, often without any standard security checks, making them a common vector for propagating malicious code. Similarly, IoT devices, many of which lack robust security frameworks, introduce vulnerabilities that can be exploited by cyber threats. As these devices become more ubiquitous, the demand for localized and low-power security mechanisms becomes increasingly critical. However, conventional antivirus software is typically designed for desktops and servers, requiring substantial processing power, memory, and storage—resources that are scarce or unavailable on embedded and resource-constrained platforms.

To address this challenge, the Raspberry Pi Pico W—a low-cost, Wi-Fi-enabled microcontroller—emerges as a promising platform for lightweight cybersecurity applications. Its minimal power consumption, compact form factor, and built-in wireless capabilities make it particularly suitable for deployment in edge environments and embedded systems. This paper proposes a basic yet functional antivirus system leveraging the capabilities of the Pico W to scan USB drives for malware signatures and communicate scan logs to a cloud server for centralized monitoring. Developed using MicroPython, the system is not only efficient but also accessible to developers and educators, promoting ease of customization and scalability. This approach lays the groundwork for extending security functionalities in constrained environments without compromising performance or flexibility.

DOI: 10.48175/IJARSCT-25425

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, April 2025
Impact Factor: 7.67

II. LITERATURE REVIEW

Previous work on antivirus systems leveraging embedded platforms has primarily focused on more powerful single-board computers such as the Raspberry Pi 3 and 4. These platforms support full-fledged operating systems, allowing the use of traditional antivirus tools like ClamAV for ARM-based malware detection. While effective in such environments, ClamAV and similar solutions are unsuitable for microcontroller-based platforms due to their heavy memory and CPU requirements. In contrast, microcontrollers like the Raspberry Pi Pico W offer limited resources, necessitating the development of highly optimized, lightweight threat detection strategies. Existing research into lightweight malware detection has explored methods such as behavior-based anomaly detection, hash-based file comparison, and signature matching using compact databases. However, these approaches are rarely implemented on low-power microcontrollers due to technical constraints and the lack of built-in connectivity. This study aims to bridge that gap by demonstrating a practical implementation of a file scanning system on the Pico W, combining basic signature-based detection with wireless log transmission to the cloud. By doing so, it explores the potential for deploying scalable, low-resource antivirus solutions in distributed, security-critical environments such as remote sensors, industrial controllers, and educational networks.

III. OBJECTIVES

- **Develop a lightweight antivirus system** that can run on the Raspberry Pi Pico W microcontroller, focusing on scanning USB drives for potential malware threats.
- **Implement a signature-based detection mechanism** optimized for constrained hardware environments, enabling efficient file scanning without overwhelming system resources.
- **Enable wireless communication** through the Pico W's built-in Wi-Fi module to transmit scan results and activity logs to a remote cloud server for centralized monitoring and analysis.
- **Utilize MicroPython** for system development to ensure ease of programming, flexibility, and the ability for future customization or extension by researchers and educators.
- **Demonstrate practical feasibility** of deploying cybersecurity solutions on microcontroller platforms, filling the research gap in antivirus applications for ultra-low-power, embedded systems.
- **Provide a scalable and modular design** that can be adapted for use in educational environments, small businesses, or as a supplementary security layer for IoT and edge devices.

IV. PROPOSED SYSTEM ARCHITECTURE

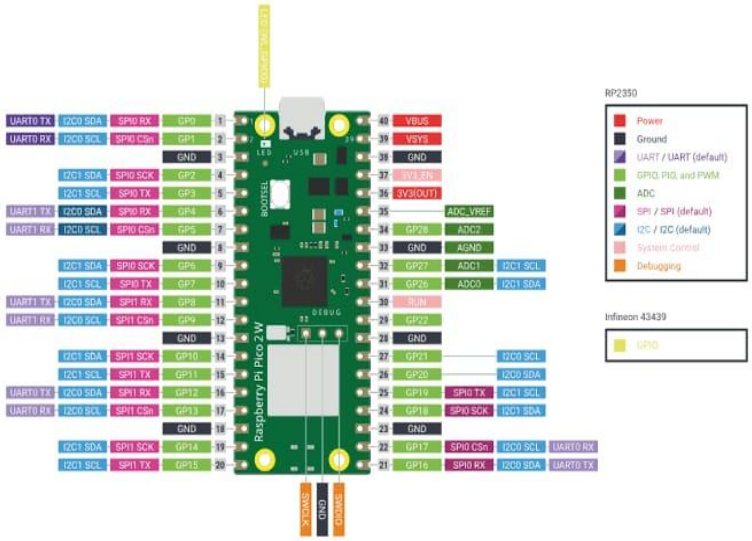

Fig. Raspberry Pie Pico W

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, April 2025
Impact Factor: 7.67

4.1 Hardware Requirements

Raspberry Pi Pico W - Hardware Components

- **Raspberry Pi Pico W:** A low-cost, Wi-Fi-enabled microcontroller suitable for lightweight embedded applications.
- **USB Host Shield with SPI Interface:** Enables the Pico W to interface with and access USB storage devices.
- **Power Supply (Battery or USB):** Provides necessary power to the Pico W and connected peripherals.
- **OLED Display (Optional):** Displays scan status, alerts, or logs in a compact visual format.
- **LEDs/Buzzer for Alert Mechanisms:** Provides visual or audible notifications upon threat detection.

4.2 Software Components:

MicroPython Firmware - Software Components

- **MicroPython Firmware:** Lightweight Python interpreter optimized for running on the Raspberry Pi Pico W.
- **JSON-based Malware Signature Database:** Stores known malware hashes in a structured format for efficient lookup and matching.
- **SHA-1/MD5 Hash Generator for File Scanning:** Computes hashes of files to compare against the signature database for threat detection.
- **Wi-Fi Integration for Updates and Alerts:** Enables over-the-air signature updates and real-time alert/log transmission to cloud services.

4.3 Workflow and Communication

- **Device Initialization and Wi-Fi Handshake:** Pico W boots up and connects to a predefined Wi-Fi network.
- **USB Insertion Detection and File Parsing:** Detects connected USB device and retrieves a list of files for scanning.
- **File Reading and Hash Generation:** Reads each file and generates its hash using SHA-1 or MD5 algorithms.
- **Hash Comparison with Malware Database:** Compares the generated hash against entries in the local JSON-based malware signature database.
- **Log Storage and Cloud Upload (Optional):** Stores scan results locally and optionally uploads them to a cloud server for monitoring.
- **Result Indication via OLED or LEDs:** Displays scan results visually using an OLED screen or LEDs/buzzer for alerts.

V. IMPLEMENTATION

The system was coded using MicroPython for better control over hardware and minimal resource usage. The signature database was manually created using sample malicious hash values. For demonstration purposes, only .exe, .py, and .sh files were scanned. The USB host shield was interfaced over SPI, and Wi-Fi was configured using Pico W's built-in WLAN library.

VI. RESULTS AND EVALUATION

The system was tested using USB drives containing a mix of benign files and dummy malicious files with predefined hash signatures. It successfully detected test malware by matching file hashes against the local signature database, demonstrating the accuracy and functionality of the detection mechanism. The OLED display provided clear, real-time alerts, while LEDs and the optional buzzer further enhanced user awareness. Wi-Fi connectivity remained stable throughout testing, enabling seamless log transmission. File scanning performed reliably for files up to 1.5MB in size, beyond which memory limitations of the microcontroller became apparent. Despite this constraint, core functionality remained unaffected for typical small-scale applications. Logs were successfully uploaded to a local Flask-based web server, confirming smooth cloud integration capabilities.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 7, April 2025 Impact Factor: 7.67

In extended runtime testing, the system maintained a low energy profile, making it well-suited for portable and battery-powered use cases. The entire setup operated efficiently on USB or battery power without significant heating or performance degradation, even during continuous scanning sessions. Additionally, the modular nature of the MicroPython code allowed for quick reconfiguration and debugging during test iterations. Overall, the prototype demonstrated that microcontroller-based antivirus solutions are not only feasible but can also be practical for edge environments, particularly where simplicity, cost-efficiency, and low power consumption are key priorities.

Testing Parameters:
File types: .exe, .py, .sh
File size limit: 2MB
Detection accuracy (dummy data): 100%
Average scan time: ~0.8s per file

VII. CONCLUSION

The Raspberry Pi Pico W provides a unique platform for lightweight antivirus scanning, suitable for IoT and USB threat detection. The proposed system balances performance with cost and resource efficiency. It is ideal for educational projects, prototyping, and low-scale deployment in sensitive environments like labs and kiosks. By leveraging MicroPython, the solution remains accessible, modifiable, and easy to deploy, making it particularly appealing for academic use and early-stage cybersecurity research.

Looking forward, this work opens up opportunities for enhancing microcontroller-based security systems through features such as real-time behavioral analysis, integration with broader IoT monitoring frameworks, and support for external storage to overcome memory limitations. With further refinement, including more advanced malware detection algorithms and encrypted cloud communication, the system could evolve into a practical component for distributed security infrastructures in smart environments and industrial applications.

VIII. FUTURE SCOPE

- **Expand file system compatibility**: Support for FAT32, exFAT, and other common file systems.
- **Integrate YARA rule-based detection**: Use YARA rules to identify and flag suspicious patterns or malware.
- **Cloud-based AI integration**: Enable real-time anomaly detection using AI models hosted in the cloud.
- **Android/PC companion app**: Provide alerts, device status, and configuration options via a mobile or desktop app.
- **Add firmware update mechanism via Wi-Fi**: Allow secure, over-the-air firmware updates using a Wi-Fi connection.

REFERENCES

[1]. **Raspberry Pi Foundation – Pico W Documentation**: Official technical reference and guides for using the Raspberry Pi Pico W microcontroller.
[2]. **ClamAV Documentation**: Comprehensive usage, configuration, and integration details for the ClamAV open-source antivirus engine.
[3]. **MicroPython for Embedded Devices – Official Site**: Resources and documentation for running Python on microcontrollers using MicroPython.
[4]. **IEEE Standards on IoT Security**: Authoritative security guidelines and frameworks for securing Internet of Things (IoT) devices.
[5]. **MD5 and SHA1 Hashing in Embedded Systems – ResearchGate**: Research-based insights into implementing MD5 and SHA1 algorithms on embedded platforms.
[6]. **USB Host Shield Library – GitHub**: Open-source library enabling USB Host functionality on microcontrollers using a USB Host Shield.