ISSN: 2754-6705

Journal of Mathematical & Computer Applications

SCIENTIFIC Research and Community

Review Article

Open Access

IoT Firmware Security Automation: QEMU-Based Fuzzing, CVSS Scoring, and Mender OTA Updates

Sandhya Guduru

Masters in Information Systems Security, Software Engineer - Technical Lead, USA

ABSTRACT

Firmware security is a growing concern in IoT environments, as many devices are shipped with outdated or vulnerable software. Traditional security methods, such as manual testing and patching, are time-consuming and ineffective against the increasing complexity of firmware architectures. This research proposes an automated approach to firmware security using QEMU-based emulation, AFL++ fuzzing, and CVSS scoring to identify vulnerabilities. Additionally, secure OTA updates are implemented through Mender, ensuring compliance with The Update Framework (TUF) and leveraging Ed25519 cryptographic signatures for protection. By integrating these technologies, this framework enhances IoT security by automating both vulnerability detection and firmware updates, reducing the risks of cyberattacks.

***Corresponding author**

Sandhya Guduru, Masters in Information Systems Security, Software Engineer - Technical Lead, USA.

Received: December 08, 2023; **Accepted:** December 15, 2023, **Published:** December 23, 2023

Keywords: IoT Security, Firmware Vulnerabilities, QEMU Emulation, AFL++ Fuzzing, CVSS Scoring, OTA Updates, Mender, The Update Framework (TUF), Ed25519 Signatures, Automated Security Testing

Introduction

The rapid growth of Internet of Things (IoT) devices has introduced significant security challenges, particularly concerning firmware protection. Many IoT devices operate on outdated firmware with weak security measures, making them susceptible to cyberattacks. Attackers exploit these vulnerabilities to gain unauthorized access, manipulate device functionality, or exfiltrate sensitive data. Addressing these threats requires a proactive security approach that automates firmware vulnerability detection and remediation.

Current security solutions often rely heavily on manual testing, which is inefficient for large-scale IoT deployments. Traditional methods struggle to keep pace with the evolving complexity of firmware architectures, leaving many vulnerabilities unpatched. Moreover, many IoT manufacturers fail to implement secure over-the-air (OTA) update mechanisms, increasing the risk of compromised firmware installations. Without robust cryptographic verification, attackers can inject malicious firmware, leading to device hijacking and persistent security threats [1].

To tackle these challenges, this research presents an automated firmware security framework. The proposed solution integrates QEMU-based emulation for dynamic firmware analysis, AFL++ fuzzing for vulnerability detection, and the Common Vulnerability Scoring System (CVSS) for assessing security risks. Additionally, Mender is utilized for secure OTA updates, ensuring compliance with The Update Framework (TUF) and employing Ed25519 signatures to prevent unauthorized modifications. By automating these security processes, this approach enhances IoT firmware protection, reducing exposure to cyber threats while ensuring the integrity and reliability of firmware updates.

Literature Review

IoT Firmware Security and Attack Vectors

IoT devices are highly vulnerable to firmware-based attacks, including buffer overflows, code injection, and firmware tampering. Research has shown that attackers exploit these weaknesses to gain unauthorized control over devices and execute malicious code. For example, a study on TrustZone-M based IoT devices found that stack-based buffer overflows and return-oriented programming (ROP) attacks can be used to bypass security measures and compromise embedded systems Similarly, another study introduced BaseSAFE, a framework for fuzzing cellular basebands in embedded systems [2]. This research demonstrated how heap-based buffer overflows could be exploited to manipulate firmware, highlighting the importance of proactive security measures [3]. These findings emphasize the need for automated security tools to detect and fix firmware vulnerabilities. Manual testing methods are insufficient for handling the increasing complexity of modern IoT firmware, making automated solutions essential for securing devices against evolving threats.

QEMU Based Emulation for Firmware Analysis

QEMU is a widely used tool in security research because it can emulate various hardware architectures, allowing researchers to test firmware without needing physical devices. This capability enables the analysis of embedded systems in a controlled environment, facilitating the identification of vulnerabilities. For instance, the Firmware Analysis Toolkit (FAT) leverages QEMU to automate the emulation of firmware images, assisting security researchers in detecting and analyzing potential security issues in IoT and embedded device firmware [4].

AFL++ Fuzzing for Vulnerability Detection

AFL++ is an advanced tool used to test software security by automatically feeding programs with varied inputs to uncover hidden bugs and vulnerabilities. It enhances traditional fuzzing methods by employing improved strategies to explore different execution paths within a program, thereby increasing the likelihood of detecting flaws. This approach has proven effective in identifying security issues in real-world software applications [5]. When combined with emulated environments, such as those provided by QEMU, AFL++ becomes even more powerful. This integration allows for thorough testing of IoT device firmware without the need for physical hardware, facilitating the discovery of vulnerabilities that could be exploited by attackers. For instance, the FirmAFL framework utilizes this combination to efficiently detect security flaws in IoT firmware [5].

CVSS Scoring for Firmware Vulnerabilities

Research analyzing vulnerabilities from 2005 to 2019 found that most security flaws could be exploited over networks, required little effort to attack, and needed minimal authentication. This shows why CVSS scoring is important—it helps security teams focus on the most dangerous threats first [6].

By using CVSS, organizations can better understand their security risks and apply fixes where they are needed most, improving overall protection against cyber attacks.

OTA Update Security in IoT Devices

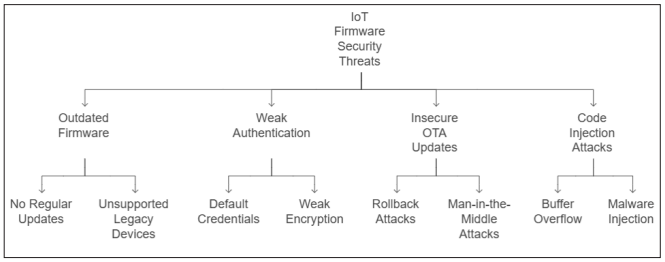
Over-the-air (OTA) updates are essential for maintaining and enhancing IoT devices, allowing manufacturers to deploy new features and security patches remotely. However, these updates can introduce risks such as rollback attacks and unauthorized modifications if not properly secured. To mitigate these risks, implementing robust cryptographic measures is crucial. One effective approach is the use of digital signatures to verify the authenticity and integrity of firmware updates. For instance, the Edwards-curve Digital Signature Algorithm (EdDSA), particularly the Ed25519 variant, offers high-performance signing and verification suitable for constrained IoT devices. Ed25519 is known for its speed and security, making it a preferred choice for ensuring that only authorized firmware is installed on devices [7].

Additionally, integrating frameworks like The Update Framework (TUF) enhances the security of OTA updates. TUF provides a flexible and robust architecture designed to protect software update systems from various attacks, including those that compromise repositories or signing keys. By incorporating TUF, organizations can ensure the integrity and authenticity of firmware updates, even in the face of partial key compromises.

Implementing these cryptographic techniques and frameworks requires careful consideration of the resource constraints inherent in many IoT devices. However, the benefits of enhanced security and trust in the update process far outweigh the challenges, making them essential components of a comprehensive IoT security strategy.

Problem Statement

Figure 1: IoT Firmware Security Threats



Many Internet of Things (IoT) devices operate with outdated or insecure firmware, making them susceptible to cyberattacks. Hackers can exploit these vulnerabilities to gain unauthorized access, steal sensitive data, or disrupt device operations. For instance, a 2021 Microsoft review highlighted increasing attacks focusing on IoT device firmware and BIOS due to significant lapses in firmware security support [8].

Traditional security measures rely heavily on manual testing, which is time-consuming and impractical for large-scale IoT deployments. As firmware complexity increases, these conventional approaches struggle to detect all vulnerabilities, leaving many security risks unaddressed. Automated tools, such as MetaEmu, an architecture-agnostic emulator synthesizer, offer a more efficient solution by enabling rehosting and security analysis of automotive firmware across multiple architectures [9].

Another significant challenge is the lack of secure, automated firmware update mechanisms. Many IoT manufacturers do not implement robust over-the-air (OTA) update processes, exposing devices to persistent threats. Without proper cryptographic verification, attackers can alter firmware updates to install malicious software. Implementing secure OTA update solutions, as discussed in a comprehensive review of IoT firmware vulnerabilities and auditing techniques, can mitigate these risks by ensuring timely and authenticated firmware updates [10].

To address these issues, automating firmware security processes is essential. Techniques such as QEMU-based emulation, AFL++ fuzzing, and CVSS scoring can enhance the detection of vulnerabilities. Additionally, secure OTA update frameworks can safeguard firmware from unauthorized modifications, thereby strengthening the overall security of IoT ecosystems.

Challenges in IoT Firmware Security

Securing IoT firmware is complex due to several technical and operational challenges. One major issue is the diversity of IoT hardware and software environments. Unlike traditional computing systems, IoT devices use a wide range of processors, architectures, and operating systems, making it difficult to apply a single security solution across all devices.

Another challenge is the limited computing power of many IoT devices. Unlike computers or smartphones, IoT devices often have minimal processing capabilities and memory, making it difficult to implement robust security mechanisms such as real-time encryption or advanced threat detection. This limitation allows attackers to exploit vulnerabilities more easily, as security measures must be lightweight to avoid slowing down device performance [11].

Additionally, firmware updates are not always implemented securely. Many IoT manufacturers prioritize cost and convenience over security, leading to weak update mechanisms that lack proper authentication. This increases the risk of attacks such as firmware tampering and rollback attacks, where an attacker forces a device to install an older, vulnerable version of the firmware. Studies have shown that implementing cryptographically signed updates can reduce these risks, but adoption remains inconsistent across the industry [12].

Finally, firmware security automation faces challenges related to scalability and accuracy. Many automated tools struggle to identify vulnerabilities in complex firmware environments, leading to false positives or missed security flaws. Without efficient automation, large-scale IoT deployments remain vulnerable due to the sheer volume of devices requiring security assessments. Emerging tools like architecture-agnostic emulators and intelligent fuzzing techniques have shown promise in addressing these issues, but their adoption is still in progress [13].

These challenges highlight the urgent need for improved security frameworks that can handle the complexity of IoT firmware. The next section explores how automation, emulation, and secure update mechanisms can help overcome these issues.

Proposed Solution

To tackle the security challenges in IoT firmware, we propose an automated system that integrates advanced tools and methodologies:

QEMU-Based Firmware Emulation

QEMU is an open-source emulator that allows running firmware in a virtual environment, eliminating the need for physical hardware. By emulating IoT devices with QEMU, we can analyze firmware behavior and identify vulnerabilities in a controlled setting. This approach enables comprehensive testing and debugging without the constraints of physical devices [14].

Firmware Security Testing Process

To streamline IoT firmware security, we propose an automated workflow integrating QEMU-based emulation, AFL++ fuzzing, CVSS scoring, and secure OTA updates. The diagram below illustrates this process, showing how vulnerabilities are detected, assessed, and patched in a structured manner. This automated pipeline eliminates manual testing inefficiencies, ensuring firmware remains secure throughout its lifecycle.

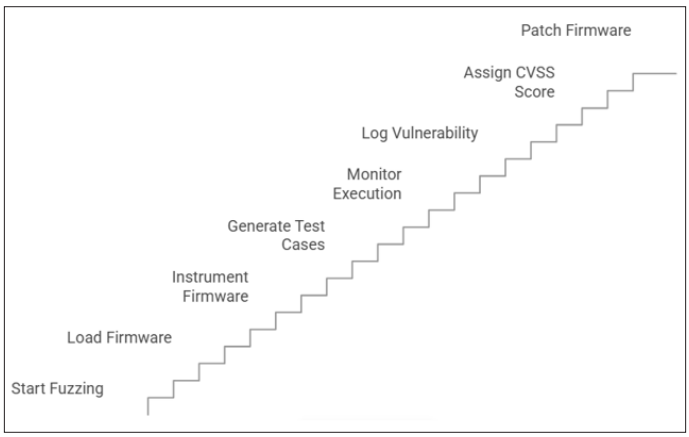
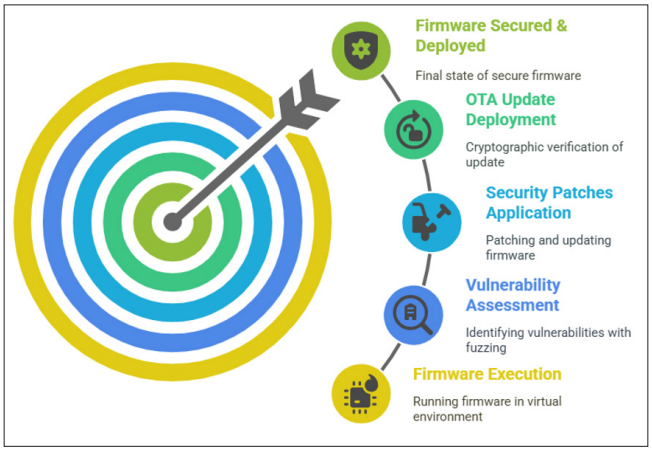


Figure 2: Firmware Security Testing Process

AFL++ Fuzzing for Vulnerability Detection

AFL++ is an enhanced version of the American Fuzzy Lop (AFL) fuzzer, designed to detect security flaws by injecting unexpected inputs into software and monitoring for anomalies. When combined with QEMU, AFL++ can effectively test emulated firmware, uncovering hidden vulnerabilities that might be missed by traditional testing methods. This integration allows for automated and thorough security assessments.

The following diagram illustrates the AFL++ fuzzing workflow, demonstrating how it detects vulnerabilities in firmware by injecting test cases and monitoring execution for crashes.



As shown in the diagram, the process begins by loading the firmware into the QEMU emulator. AFL++ then instruments the firmware, generates test cases, and injects inputs to uncover security flaws. If a crash is detected, the vulnerability is logged, assigned a CVSS score, and patched before retesting. This automated workflow enhances firmware security by identifying and fixing vulnerabilities efficiently.

CVSS Scoring for Assessing Vulnerabilities

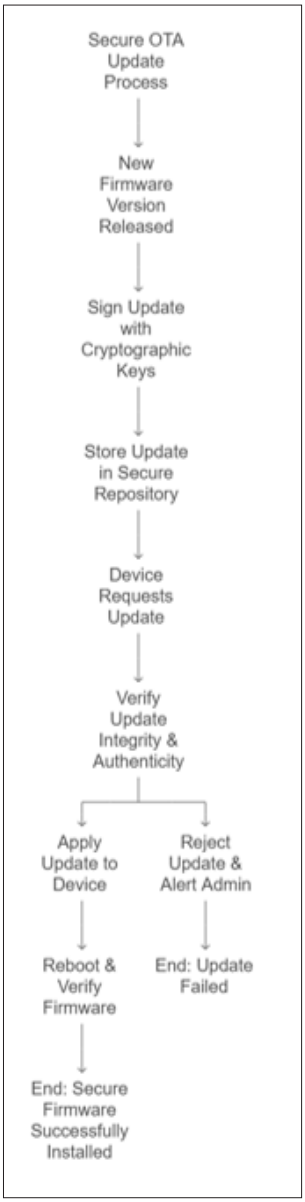
Once vulnerabilities are identified, it's crucial to assess their severity to prioritize remediation efforts. The Common Vulnerability Scoring System (CVSS) provides a standardized framework for evaluating the impact and exploitability of security flaws. By applying CVSS scores to detected vulnerabilities, organizations can effectively prioritize patches and allocate resources to address the most critical issues first.

Secure OTA Updates with Mender and TUF Compliance

After identifying and assessing vulnerabilities, deploying secure firmware updates is essential to mitigate risks. Mender is an open-source tool that facilitates over-the-air (OTA) firmware updates for IoT devices, ensuring that updates are delivered reliably and securely. Integrating Mender with The Update Framework (TUF) enhances the security of the update process by protecting against various attacks, such as rollback and mix-and-match attacks, ensuring that only authenticated and integrity-verified updates are applied to devices.

Secure OTA Update Process with Mender and TUF Compliance

Ensuring that firmware updates are securely delivered and verified is crucial for IoT security. The diagram below outlines the secure OTA update process, from signing new firmware to final verification, preventing unauthorized modifications and attacks.



By combining QEMU-based emulation, AFL++ fuzzing, CVSS scoring, and secure OTA updates with Mender and TUF compliance, our proposed solution offers a comprehensive and automated approach to enhancing IoT firmware security. This integration not only streamlines the vulnerability detection and assessment process but also ensures that remediation efforts are effectively and securely implemented, thereby strengthening the overall security posture of IoT devices.

Conclusion

Firmware security is a critical challenge in IoT environments, as many devices are shipped with outdated or vulnerable firmware. Traditional security approaches, which rely on manual testing and reactive patching, are not sufficient to address the growing number of threats. To mitigate these risks, an automated solution is necessary.

This research proposes a security framework that integrates QEMU-based firmware emulation, AFL++ fuzzing, and CVSS scoring to systematically identify vulnerabilities. Additionally, secure OTA updates using Mender and TUF compliance ensure that firmware updates are protected against unauthorized modifications. By automating both vulnerability detection and secure firmware updates, this approach enhances the overall security of IoT devices.

Implementing this solution can help manufacturers and security teams proactively defend against cyber threats, reducing the risk of device takeovers, data breaches, and service disruptions. Future research should explore ways to further improve automation, integrate machine learning for vulnerability detection, and expand compatibility with a wider range of IoT hardware.

References

1. X Feng, X Zhu, QL Han, W Zhou, SWen et al. (2022) Detecting Vulnerability on IoT Device firmware: a Survey IEEE/CAA. Journal of Automatica Sinica 1-17.
2. L Luo, Y Zhang, C C Zou, X Shao, Z Ling et al. (2020) On Runtime Software Security of TrustZone-M based IoT Devices. arXiv.org: https://arxiv.org/abs/2007.05876.
3. D Maier, L Seidel, S Park (2020) BaseSAFE: baseband sanitized fuzzing through emulation. AMC DL 122-132.
4. Attify (2022) GitHub -attify/firmware-analysis-toolkit: Toolkit to emulate firmware and analyse it for security vulnerabilities.

5. Y Zheng, A Davanian, H Yin, C Song, H Zhu et al. FIRM-AFL: High-Throughput Greybox Fuzzing of IoT Firmware via Augmented Process Emulation. https://www.cs.ucr.edu/~heng/pubs/FirmAFL.pdf?.
6. A Gueye, P Mell (2021) A Historical and Statistical Study of the Software Vulnerability Landscape arXiv.org https://arxiv.org/abs/2102.01722?.
7. H Gupta, P Van Oorschot (2019) "Onboarding and Software Update Architecture for IoT Devices." https://www.scs.carleton.ca/~paulv/papers/PST2019-gupta.pdf?.
8. O Media (2022) Understanding UEFI Firmware Update and Its Vital Role in Keeping Computing Systems Secure-Embedded Computing Design Embedded Computing Design https://embeddedcomputing.com/technology/security/software-security/understanding-uefi-firmware-update-and-its-vital-role-in-keeping-computing-systems-secure.
9. Z Chen, SL Thomas, FD Garcia (2022) MetaEmu: An Architecture Agnostic Rehosting Framework for Automotive Firmware arXiv.org https://arxiv.org/abs/2208.03528?.
10. M Lezzi, M Lazoi, A Corallo (2018) Cybersecurity for Industry 4.0 in the current literature: A reference framework Computers in Industry 97-110.
11. G M Richardson, I Douair, SA Cameron, L Maron, M D Anker (2021) Ytterbium (II) Hydride as a Powerful Multielectron Reductant Chemistry (Weinheim an der Bergstrasse, Germany) Autumn 144-148.
12. G Acquaviva, N Katirci (2022) Dynamical analysis of logarithmic energy-momentum squared gravity arXiv.org, https://arxiv.org/abs/2203.01234.
13. A Costin, A Zarras, A Francillon (2015) "Automated Dynamic Firmware Analysis at Scale: A Case Study on Embedded Web Interfaces," arXiv.org, https://arxiv.org/abs/1511.03609.
14. F Bellard (2005) QEMU, a Fast and Portable Dynamic Translator.," ResearchGate 41-46.

Copyright: ©2023 Sandhya Guduru. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.