International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 7 July 2023, pp: 585-590www.ijaem.net ISSN: 2395-5252

Embedded System Hacking: Techniques, Tools, and Mitigation Strategies

Heng Zhi Yong[1], Teoh Chun Hwung[1], Mohamad Fadli Zolkipli[2]

[1] *Awang Had Salleh Graduate School, School of Computing, Universiti Utara Malaysia, Kedah, Malaysia*
[2] *School of Computing, Universiti Utara Malaysia, Kedah, Malaysia.*

Date of Submission: 15-07-2023 Date of Acceptance: 25-07-2023

ABSTRACT: Embedded systems play a critical role in today's interconnected world, powering various devices and enabling seamless communication and automation. However, the increasing connectivity and complexity of these systems have introduced new challenges in securing them from malicious attacks. The field of embedded system hacking has emerged as a crucial area of study, aiming to uncover vulnerabilities, explore exploitation techniques, and develop effective mitigation strategies. This study investigates the topic of embedded system hacking, focusing on techniques, tools, and mitigation strategies employed in this domain, with a specific emphasis on firmware vulnerabilities.

KEYWORDS: Embedded system, malicious attacks, mitigation strategies,system hacking, focusing on techniques, tools

I. INTRODUCTION

Embedded systems play a pivotal role in our interconnected world, powering a wide range of devices and enabling seamless communication and automation. From smart appliances and industrial control systems to medical devices and automotive systems, embedded systems have become an integral part of our daily lives. However, the rapid evolution of technology, coupled with increasing connectivity, has brought forth new challenges in securing these systems from malicious attacks. The field of embedded system hacking has emerged as a critical area of study, aiming to uncover vulnerabilities, explore exploitation techniques, and develop effective mitigation strategies [1].

The objective of the investigation is to provide a comprehensive examination of embedded system hacking, focusing on the techniques, tools, and mitigation strategies employed in this domain. By delving into the background, history, and current issues/threats surrounding embedded system security, we aim to shed light on the vulnerabilities that can be exploited by adversaries seeking to compromise these systems. Furthermore, we explore the various techniques and tools used by hackers to exploit these vulnerabilities, thereby emphasizing the importance of understanding the adversarial perspective.

Embedded systems face a multitude of security challenges, including firmware vulnerabilities, network-based attacks, physical tampering, and supply chain compromises. The ever-expanding attack surface and the increasing sophistication of adversaries necessitate a thorough understanding of these threats to design and implement robust security measures. By analyzing real-world case studies and incidents, will provide insights into the impact and consequences of embedded system hacking, highlighting the potential risks posed to critical infrastructure, personal privacy, and public safety.

To address these challenges, various techniques and tools have been developed to identify and mitigate vulnerabilities in embedded systems. From secure coding practices and firmware integrity verification to network security measures and hardware-level security features, an array of strategies have been proposed to safeguard these systems from hacking attempts. To delve into these mitigation strategies, discussing their strengths, limitations, and potential areas for improvement.

Through an exploration of embedded system hacking and the corresponding mitigation strategies, this study aims to raise awareness and foster a deeper understanding of the complex landscape of embedded system security. The intention is for this research to serve as a valuable resource to individuals involved in securing embedded systems, including

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 7 July 2023, pp: 585-590www.ijaem.net ISSN: 2395-5252

researchers,practitioners, and decision-makers, as they strive to protect against continuously evolving threats.

II. LITERATURE REVIEW

Embedded systems are pervasive in our modern society, playing a crucial role in various domains, including transportation, healthcare, industrial control, and smart homes. However, the increasing connectivity and complexity of embedded systems have exposed them to a myriad of security threats. This is to give a thorough insight of the background, timeline, and current problems and concerns surrounding embedded system hacking. Furthermore, it explores the techniques, tools, and mitigation strategies employed to safeguard these systems from malicious attacks.

1. Background

Embedded systems refer to computer systems integrated within devices, typically with limited resources and specific functionalities. They rely on microcontrollers or microprocessors to perform dedicated tasks, often in real-time environments. Over the years, embedded systems have evolved from simple standalone devices to interconnected systems, the Internet of Things' (IoT) supporting infrastructure [2].

2. History

The history of embedded system hacking can be traced back to the early days of computing. As technology advanced, so did the methods employed by hackers to exploit vulnerabilities in embedded systems. Early attacks targeted specific devices and focused on physical access or software manipulation. However, with the proliferation of networked embedded systems, new attack vectors emerged, making it possible to compromise devices remotely.

3. Issues/ Threats

a. Firmware vulnerabilities

Insecure firmware exposes embedded systems to various attacks, such as code injection, privilege escalation, and unauthorized access.

Firmware serves as the software component that runs on the embedded system's hardware, providing low-level control and functionality. It acts as the bridge between the hardware and higher-level software, enabling the system to perform its designated tasks. However, vulnerabilities in firmware can expose the system to exploitation and compromise its security.

One prevalent issue with firmware is the lack of proper security measures during its development and deployment. Insufficient validation of firmware updates and inadequate secure coding practices can lead to vulnerabilities that adversaries can exploit. Attackers may attempt to inject malicious code into the firmware, compromising its integrity and allowing unauthorized functionality or unauthorized access to the embedded system. Such code injection attacks can enable privilege escalation, where an attacker gains elevated privileges within the system, potentially bypassing access controls and gaining control over critical functionalities.

b. Network-based attacks

Network-based attacks pose significant threats to the security of embedded systems, exposing them to various risks such as unauthorized access, data breaches, and manipulation of system functionality. Interconnected embedded systems are susceptible to network-based attacks, including eavesdropping, man-in-the-middle attacks, and distributed denial-of-service (DDoS) attacks.[3].

c. Physical attacks

Adversaries can target the physical components of embedded systems, such as tampering with hardware, extracting sensitive information, or bypassing security measures. The security and integrity of embedded systems, especially cyber-physical systems (CPS), are significantly at risk from physical attacks. As CPS become increasingly autonomous and interconnected, they are susceptible to exploitation by malicious actors who can manipulate physical components to deceive the system into performing dangerous actions. While many existing works focus on preventing and detecting attacks, the critical question arises: "What should be done after detecting an attack?" This article delves into the problem of attack response and proposes novel real-time recovery techniques to enhance the security of CPS [4].

d. Supply chain attacks

Supply chain attacks pose significant risks to the security of embedded systems, making them vulnerable to unauthorized access, data breaches, and malicious manipulation. These attacks occur when adversaries exploit vulnerabilities in the supply chain to compromise the integrity and security of the components or software used in embedded systems. Supply chain attacks can occur at various stages of the supply chain, including the

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 7 July 2023, pp: 585-590www.ijaem.net ISSN: 2395-5252

manufacturing, assembly, distribution, or software development processes. Attackers may infiltrate the supply chain to introduce malicious components, tamper with hardware or firmware, or inject malicious code into the software. These compromised elements can then be integrated into the embedded systems, providing a backdoor for unauthorized access or control [5].

4. Techniques

a. Reverse Engineering

Hackers utilize reverse engineering techniques to understand the inner workings of embedded systems. By analyzing firmware, protocols, and software binaries, they uncover vulnerabilities and discover potential attack vectors. Furthermore, machine learning algorithms play a significant role in hardware trust and assurance. By leveraging machine learning techniques, researchers can train models to detect anomalies, identify counterfeit components, and enhance the overall security of hardware systems. Large datasets may be analysed by machine learning techniques, and patterns found there may point to possible security holes or malicious hardware changes. This proactive approach can aid in the identification and prevention of security breaches in embedded systems [6].

b. Fault Injection

Fault injection involves intentionally introducing faults or errors into an embedded system to observe its response and identify vulnerabilities. Techniques such as voltage glitching, electromagnetic interference (EMI), and clock manipulation are employed to manipulate the system's behaviour and exploit weaknesses. [7]

c. Side-Channel Attacks

According to [8], side-channel attacks exploit various channels, such as power consumption, electromagnetic radiation, timing, and cache utilization, to infer information about cryptographic keys or other sensitive data. It emphasizes that side-channel attacks can be particularly challenging to detect and defend against.

d. Buffer Overflows

Buffer overflow attacks target vulnerabilities in input validation and memory handling, allowing adversaries to overwrite critical data or inject malicious code into an embedded system's memory. This technique is commonly used to gain unauthorized access or execute arbitrary commands [9].

e. Software-Defined Radio Attacks

Software-defined radio (SDR) attacks leverage the programmable nature of embedded systems with wireless communication capabilities. Hackers can manipulate radio frequencies, intercept or modify wireless communications, and perform attacks like replay attacks or signal jamming [10].

5. Tools

a. JTAG Debugging Interfaces

JTAG (Joint Test Action Group) debugging interfaces provide access to the debugging and testing features of embedded systems. They allow hackers to interact with the system's internal components, analyze firmware, and potentially exploit vulnerabilities. [11]

b. Hardware Emulators

Hardware emulators simulate the behavior of specific hardware components or entire embedded systems. They enable hackers to test and debug software in a controlled environment, facilitating the identification of vulnerabilities and potential exploitation techniques. [12]

c. Software Analysis Frameworks

Software analysis frameworks provide tools and libraries for analyzing the behavior and security of embedded system software. These frameworks often include static code analysis, dynamic analysis, and fuzzing capabilities to detect vulnerabilities and assess system security [13] [14]

6. Mitigation Strategies

a. Secure coding practices

Developing solid firmware and software for embedded devices by putting secure coding standards and recommendations into practice. According to [15], adhere to secure coding principles such as least privilege, separation of duties, and principle of least astonishment. Regularly update and patch dependencies to avoid known vulnerabilities.According to [16], to prevent common vulnerabilities such as buffer overflows, injection attacks, and insecure cryptographic implementations. This includes input validation, proper error handling, secure memory management, and using secure coding libraries.

b. Firmware integrity verification

Using methods like firmware encryption and digital signatures to safeguard the authenticity and integrity of embedded system firmware [17].

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 7 July 2023, pp: 585-590www.ijaem.net ISSN: 2395-5252

c. Network security measures

Implementing secure communication protocols, access control mechanisms, and intrusion detection systems to protect networked embedded systems. [18]

d. Hardware security features

Incorporating hardware-level security measures, such as secure elements, trusted platform modules (TPMs), and hardware-based encryption, to enhance the overall security posture of embedded systems. [19]

III. ISSUES AND CHALLENGES IN PENETRATION TESTING

a. Lack of Standardization

The lack of standardized protocols and interfaces in embedded systems poses challenges for penetration testing. Each embedded system may have unique communication protocols, proprietary firmware, and custom interfaces, requiring testers to invest significant effort in understanding and reverse engineering these components.

b. Real-Time Constraints

Many embedded systems operate in real-time environments, with strict timing requirements. Penetration testing in such scenarios needs to consider the impact on system functionality and ensure that critical operations are not disrupted or delayed. Testing methodologies and tools must be adapted to accommodate real-time constraints without affecting the system's performance or reliability [20]

c. Physical Access Challenges

In some cases, embedded systems may be deployed in physically challenging or hostile environments, making physical access for testing purposes difficult or impossible. Remote testing methods and techniques need to be developed to address such scenarios and ensure comprehensive assessment of system security [21]

d. Lack of Awareness and Expertise

Embedded system security, including penetration testing, is a specialized field that requires specific knowledge and expertise. There is a shortage of skilled professionals with deep understanding and experience in this domain. Building and maintaining a skilled workforce capable of conducting effective penetration testing for embedded systems is a significant challenge [22].

IV. DISCUSSION

a. Vulnerabilities and Exploitation Techniques

The various vulnerabilities that embedded systems are susceptible to, including firmware weaknesses, network-based attacks, physical tampering, and supply chain compromises. Furthermore, to delve into the different exploitation techniques employed by hackers, such as reverse engineering, fault injection, side-channel attacks, buffer overflows, and software-defined radio attacks. Understanding these vulnerabilities and techniques is crucial for developing effective mitigation strategies. [23]

b. Impact on Critical Infrastructure and Public Safety

Critical infrastructure sectors like energy, transportation, and healthcare are seriously at risk from embedded system hacking. the possible repercussions of successful assaults on these systems, such as the interruption of crucial services, the compromise of private information, and risks to public safety. By analyzing real-world incidents and case studies, emphasize the importance of safeguarding embedded systems to mitigate these risks [24]

c. Mitigation Strategies and Best Practices

The mitigation strategies and best practices employed to protect embedded systems from hacking attempts. The importance of secure coding practices, firmware integrity verification, network security measures, and hardware-level security features. Furthermore, the challenges associated with implementing these strategies, such as resource limitations, compatibility issues, and the need for continuous testing and updates [25]

d. Ethical and Legal Considerations

Embedded system hacking raises ethical and legal considerations that must be addressed. The importance of responsible disclosure, collaboration between security researchers and system manufacturers, and adherence to relevant regulations and compliance requirements. Additionally, highlight the need for establishing clear guidelines and frameworks to ensure ethical conduct in embedded system security research [26].

V. CONCLUSION

The field of embedded system hacking poses significant challenges and demands comprehensive knowledge of the techniques, tools,

and mitigation strategies employed to protect these systems from malicious attacks. Throughout this study, this explored various aspects of embedded system hacking, focusing on key issues and threats such as firmware vulnerabilities, network-based attacks, physical attacks, supply chain attacks, and the techniques of reverse engineering.

Embedded systems play a pivotal role in our interconnected world, powering a wide range of devices and enabling seamless communication and automation. However, their rapid evolution and increasing connectivity have exposed them to new and sophisticated threats. Insecure firmware can lead to code injection, privilege escalation, and unauthorized access, highlighting the need for secure coding practices, firmware integrity verification, and timely patch management.

By raising awareness of these issues and exploring the various mitigation strategies available, this study aims to empower individuals involved in securing embedded systems. Researchers, practitioners, and decision-makers can use the knowledge gained from this study to protect against evolving threats and ensure the security, integrity, and reliability of embedded systems in our interconnected world.

As the field of embedded system hacking continues to evolve, further study is needed to address emerging threats, develop more efficient mitigation strategies, and enhance the resilience of embedded systems. By staying vigilant, collaborating, and continually improving our understanding of embedded system security, this can effectively protect critical infrastructure, personal privacy, and public safety.

In conclusion, embedded system hacking presents significant challenges, but with the appropriate techniques, tools, and mitigation strategies in place, this can secure these systems and ensure a safer and more resilient digital landscape.

VI. ACKNOWLEDGEMENT

The authors would like to thank all School of Computing members who were involved in this study. This study was conducted for the purpose of Ethical Hacking & Penetration Testing Research Project. This work was supported by Universiti Utara Malaysia.

REFERENCES

[1] S. Ravi, A. Raghunathan, P. Kocher and S. Hattangady, “Security in embedded systems: Design challenges,” ACM Transactions on Embedded Computing Systems (TECS), vol. 3, no. 3, pp. 461-491, 2004.

[2] Y. Joobeom, F. Rustamov, K. Juhwan and S. Younghoo, “Fuzzing of Embedded Systems: A Survey,” ACM Computing Surveys, vol. 55, no. 7, pp. 1-33, 2022.

[3] A. O. Aseeri, “A Problem-tailored Adversarial Deep Neural Network-Based Attack Model for Feed-Forward Physical Unclonable Functions,” ACM Transactions on Design Automation of Electronic Systems, vol. 28, no. 4, pp. 1-18, 2023.

[4] Z. Lin, L. Pengyuan, K. Fanxin, C. Xin, O. V. Sokolsky and I. Lee, “Real-time attack-recovery for cyber-physical systems using linear-quadratic regulator,” ACM Transactions on Embedded Computing Systems (TECS), vol. 20, no. 5s, pp. 1-24, 2021.

[5] T. Mishra, T. (. Chantem and R. M. Kepke Gerdes, “Survey of control-flow integrity techniques for real-time embedded systems,” ACM Transactions on Embedded Computing Systems (TECS), vol. 21, no. 4, pp. 1-32, 2022.

[6] U. J. Botero, R. Wilson, L. Hangwei, M. T. Rahman, M. A. Mallaiyan, F. Ganji, N. Asadizanjani, M. M. Tehranipoor, D. L. Woodard and D. J. Forte, “Hardware trust and assurance through reverse engineering: A tutorial and outlook from image analysis and machine learning perspectives,” ACM Journal on Emerging Technologies in Computing Systems (JETC), vol. 17, no. 4, pp. 1-53, 2021.

[7] B. Yuce, N. F. Ghalaty, C. Deshpande, H. Santapuri, C. Patrick, L. Nazhandali and P. R. Schaumont, “Analyzing the fault injection sensitivity of secure embedded software,” ACM Transactions on Embedded Computing Systems (TECS), vol. 16, no. 4, pp. 1-25, 2017.

[8] N. Belleville, D. Couroussé, K. Heydemann and H.-P. Charles, “Automated software protection for the masses against side-channel attacks,” ACM Transactions on Architecture and Code Optimization (TACO), vol. 15, no. 4, pp. 1-27, 2018.

[9] M. L. Chaim, D. S. Santos and D. S. Cruzes, “What do we know about buffer overflow detection?: A survey on techniques to detect a persistent vulnerability,” International Journal of Systems and Software Security and

Protection (IJSSSP), vol. 9, no. 3, pp. 1-33, 2018.

[10] Y. Kumar, G. Jajoo, A. Kumar and S. K. Yadav, "Implementation of modulation classifier over software defined radio," IET Communications, vol. 14, no. 9, pp. 1467-1475, 2020.

[11] L. Kuen-Jong, L. Zheng-Yao and Y. Shih-Chun, "A secure JTAG wrapper for SoC testing and debugging," IEEE Access, vol. 10, pp. 37603-37612, 2022.

[12] C. Spensky, A. Machiry, N. Redini, C. Unger, G. Foster, E. Blasband, H. R. Okhravi, C. Kruegel and G. Vigna, "Conware: Automated modeling of hardware peripherals," in Proceedings of the 2021 ACM Asia conference on computer and communications security, 2021, May.

[13] L. Yufeng, "Embedded controller Visual Configuration software design analysis," in Proceedings of the 2021 3rd International Conference on Software Engineering and Development, 2021, November.

[14] N. Aaraj, A. Raghunathan and N. Jha, "A framework for defending embedded systems against software attacks," ACM Transactions on Embedded Computing Systems (TECS), vol. 20, no. 3, pp. 1-23, 2011.

[15] OWASP, "OWASP Go Secure Coding Practices Guide," OWASP, 7 March 2022. [Online]. Available: https://owasp.org/www-project-go-secure-coding-practices-guide/. [Accessed 8 June 2023].

[16] SAFECode, "Fundamental Practices for Secure Software," SAFECode, March 2018. [Online]. Available: https://safecode.org/wp-content/uploads/2018/03/SAFECode_Fundamental_Practices_for_Secure_Software_Development_March_2018.pdf. [Accessed 8 June 2023].

[17] L. Yanlin, J. M. McCune and A. Perrig, "VIPER: Verifying the integrity of peripherals' firmware," in Proceedings of the 18th ACM conference on Computer and communications security, 2011.

[18] M. B. Salunke, P. N. Mahalle and G. R. Shinde, "Importance of Lightweight Algorithm for Embedded Security in Machine-to-Machine Communication towards Internet of Things," in Proceedings of the 4th International Conference on Information Management & Machine Intelligence, 2022, December.

[19] H. Qiang, Z. Zhun, X. Dongdong, W. Jiqing, L. Jiakang, Z. Jinlei, M. Jinhui and W. Xiang, "A hardware security-monitoring architecture based on data integrity and control flow integrity for embedded systems," Applied Sciences, vol. 12, no. 15, p. 7750, 2022.

[20] M. Hasan, S. Mohan, R. B. Bobba and R. Pellizzoni, "Beyond just safety: Delay-aware security monitoring for real-time control systems," ACM Transactions on Cyber-Physical Systems (TCPS), vol. 6, no. 3, pp. 1-25, 2022.

[21] M. A. Al Faruque, "Cross-Layer Security of Embedded and Cyber-Physical Systems," in Proceedings of the 2022 ACM CCS Workshop on Additive Manufacturing (3D Printing) Security, 2022, November.

[22] T. Dey, A. Karnauch and A. Mockus, "Representation of developer expertise in open source software," in 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 2021, May.

[23] D. Papp, M. Zhengdong and L. Buttyan, "Embedded systems security: Threats, vulnerabilities, and attack taxonomy," in 2015 13th Annual Conference on Privacy, Security and Trust (PST), 2015, July.

[24] A. B. Jeddi, A. Shafieezadeh and R. Nateghi, "PDP-CNN: A Deep Learning Model for Post-Hurricane Reconnaissance of Electricity Infrastructure on Resource-Constrained Embedded Systems at the Edge," IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1-9, 2023.

[25] A. D. Williams, S. N. Abbott, N. Shoman and W. S. Charlton, "Results from invoking artificial neural networks to measure insider threat detection & mitigation," Digital Threats: Research and Practice (DTRAP), vol. 3, no. 1, pp. 1-20, 2021.

[26] J. Tørresen and A. Nakazawa, "Ethical Considerations in User Modeling and Personalization (ECUMAP) ACM UMAP 2022 Tutorial," in Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization, 2022, July.