Ransomware attacks and countermeasures by secure programming

Mohamed Madi [1], Dil Hakro [2], Amani Al Hilali[3], Mohamed Kari[4]

PG25S2335@mec.edu.om[1], dhakro@mec.edu.om[2], PG24F2317@mec.edu.om[3]
PG25S2336@mec.edu.om[4]

Department of Computing and Electronics Engineering, Middle East College, Muscat, Oman [1, 2, 3,4]

Abstract. Ransomware has become one of the most damaging forms of cybercrime, encrypting data and demanding payment often in cryptocurrency to restore access. These attacks cause financial loss, data breaches, reputational harm, and operational disruption. This research highlights how secure programming practices can prevent such incidents by eliminating vulnerabilities arising from weak input validation, insecure authentication, and poor error handling. It emphasizes the role of secure coding frameworks like OWASP Secure Coding Practices, memory-safe programming languages, and cryptographic best practices in developing resilient software. The study also analyzes real-world ransomware cases, identifying key attack vectors such as phishing, malicious attachments, and drive-by downloads. Furthermore, it explores how adopting a secure software development lifecycle (SDLC) including threat modeling, code review, penetration testing, and continuous monitoring can proactively detect and mitigate flaws. Overall, secure programming forms a crucial defense layer against ransomware and strengthens cybersecurity resilience.

Keywords: Ransomware attacks, Security, Phishing attacks, misconfiguration.

1 Introduction

In recent years, the digital landscape has become increasingly perilous due to the alarming rise of ransomware attacks. As highlighted in the abstract, these attacks are not just isolated events but widespread campaigns targeting everything from personal devices to national infrastructure. Their success is often rooted in poor software practices and unpatched vulnerabilities flaws that could have been avoided with better development standards. As cybercriminals become more organized and resourceful, the stakes for software developers and cybersecurity professionals continue to grow.

To effectively combat this evolving threat, a proactive approach is needed one that embeds security at the core of software development. Secure programming offers a structured set of methodologies and practices that help developers write code that resists exploitation. From secure coding standards to rigorous testing and threat modelling, these techniques are essential for building robust applications capable of withstanding ransomware tactics.

MECSRC 2025, May 01, Muscat, Oman

DOI 10.4108/eai.1-5-2025.2361392

This paper delves deeper into the anatomy of ransomware, explores the vulnerabilities it exploits, and highlights how secure programming can act as a frontline defence. It presents case studies of major incidents, discusses secure software development lifecycle (SDLC) principles, and examines emerging trends in cybersecurity. The goal is to demonstrate how secure programming is not just a preventative measure, but a critical strategy in the broader fight against ransomware.

2 Types of Ransomware Attacks

Ransomware attacks come in various forms, each with distinct tactics and objectives. Understanding the different types of ransomwares is crucial for developing targeted defence strategies and secure programming measures. The following are the most common types of ransomware attacks:

2.1 Crypto Ransomware

Crypto ransomware is the most prevalent form. It encrypts the victim's files, rendering them inaccessible without a decryption key. The attacker demands a ransom in exchange for the key. This type does not typically affect system functionality but causes massive data loss if backups are unavailable. Examples include WannaCry and CryptoLocker, shown in figure 1.

Fig. 1. Crypto Ransomware Process.

2.2 Locker Ransomware

Locker ransomware locks the user out of their device entirely, preventing access to applications, files, and even system functions. Unlike crypto ransomware, it does not encrypt files but denies the user access to the system until a ransom is paid. A notable example is Reveton, which impersonated law enforcement agencies to scare victims into paying fines, shown in figure 2.

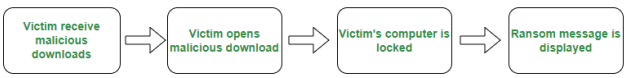

Fig. 2. Locker Ransomware Process.

2.3 Scareware

Scareware uses fear tactics and fake warnings to trick users into paying for unnecessary software or services. It may appear as a pop-up claiming that the system is infected with malware and encourages users to purchase fake antivirus software. While less technically damaging, it is effective due to psychological manipulation.

2.4 Doxware (Leakware)

Doxware threatens to publish or leak sensitive personal or corporate data unless the ransom is paid. This type of ransomware targets not only system integrity but also the victim's reputation or privacy. The Maze ransomware group is infamous for using this tactic.

2.5 Ransomware-as-a-service (RaaS)

RaaS allows cybercriminals to "rent" ransomware kits from developers and carry out attacks without needing deep technical skills. The service providers take a cut of the ransom profits. RaaS significantly lowers the barrier to entry for cybercriminals, increasing the frequency and reach of attacks. Examples include DarkSide and REvil.

2.6 Mobile Ransomware

Targeting smartphones and tablets, mobile ransomware often spreads via malicious apps or phishing links. It may lock the screen or encrypt files and typically affects Android devices. Attackers use similar ransom demands, often paid in cryptocurrency.

Each type of ransomware attack exploits different technical and human vulnerabilities. Secure programming practices tailored to these threat models can drastically reduce the attack surface and protect users and organizations alike [1].

3 Countermeasures against ransomware attacks

The increasing sophistication of ransomware has made countermeasures an essential part of any cybersecurity strategy. Successfully defending against ransomware requires a combination of preventive measures, timely detection, and well-practiced incident response. While there is no one-size-fits-all solution, organizations can significantly reduce their risk by combining secure programming practices with robust cybersecurity policies and tools [2].

3.1 General Countermeasures

Secure Programming Practices. Eliminating software vulnerabilities through secure coding is a foundational defence. Practices like input validation, memory safety, and proper error handling help ensure that ransomware cannot exploit basic flaws in applications. Developers should follow frameworks such as OWASP Secure Coding Practices and conduct frequent code reviews and security testing.
Regular Data Backups. Maintaining encrypted, offline, and regularly updated backups ensures that critical data can be restored in the event of an attack. These backups should not be connected to the main network and should be tested periodically.
Patch Management and Software Updates. Many ransomware attacks exploit known vulnerabilities in outdated software. Promptly applying security patches and updates across all systems helps close these gaps before attackers can take advantage.
Email and Endpoint Protection. Because phishing is a common delivery method, implementing email filters and user awareness training helps reduce the risk. Additionally, endpoint detection and response (EDR) tools can identify suspicious behaviour and isolate affected machines.
Network Segmentation and Access Controls. Limiting user privileges and segmenting networks restricts lateral movement, reducing the impact of a potential ransomware breach. Implementing Zero Trust principles can further strengthen this defence.
Incident Response Plan. Having a rehearsed incident response plan that includes steps for detection, containment, communication, and recovery is critical to minimize damage and downtime during an actual attack [3].

3.2 Countermeasures for Crypto Ransomware

Crypto ransomware is designed to encrypt files and demand a ransom for their decryption. Countering it requires pre-emptive data protection and strong detection capabilities.

Encryption-Aware Backup Strategy. Implement frequent, versioned backups of critical files that are stored offline or in secure cloud storage. Backup solutions should be immune to encryption by the same system ransomware might infect.
Cryptographic Monitoring Tools. Use tools that detect abnormal encryption behaviour such as sudden, bulk file modifications to halt processes that resemble ransomware activity.
Application Whitelisting. Only allow approved software to run on systems. This limits the execution of unauthorized encryption tools commonly used in crypto ransomware.
File Integrity Monitoring. Monitor and log file changes to detect potential tampering or mass encryption early [4].

3.3 Countermeasures for locker Ransomware

Locker ransomware restricts user access to a device or system rather than encrypting files. Preventing and mitigating this type involves preserving system accessibility and ensuring alternative access.
Secondary Access Options. Configure systems with alternate administrative accounts or remote management tools to regain control if the primary interface is locked.

System Restore and Boot Media. Maintain system restore points and create bootable recovery drives that allow users to access and repair locked systems without engaging with the ransomware.
Strong Authentication Practices. Locker ransomware often exploits weak credentials or open RDP (Remote Desktop Protocol) ports. Enforce multi-factor authentication and disable unnecessary remote access services.
User Behaviour Restrictions. Limit administrative rights for end-users. Locker ransomware typically requires elevated privileges to disable functionality, which restricted user permissions can prevent.

By implementing these layered countermeasures, individuals and organizations can dramatically reduce the chances of falling victim to ransomware. Most importantly, a culture of proactive security starting at the code level can fortify systems against even the most aggressive attacks [5].

4 Conclusion

Ransomware continues to pose one of the most formidable threats in the digital age, with attacks becoming more frequent, targeted, and costly. From paralyzing healthcare systems to disrupting global supply chains, the consequences of these attacks extend beyond financial losses, often affecting lives and critical infrastructure. As ransomware evolves, so too must our methods of defence.

At the core of effective ransomware prevention lies secure programming a proactive approach that addresses the root causes of many successful attacks. By eliminating common software vulnerabilities and incorporating security throughout the software development lifecycle, developers can build applications that are not only functional but resilient to exploitation. Techniques such as input validation, secure authentication, memory-safe coding, and rigorous code review processes reduce the attack surface significantly [6].

In addition to secure coding practices, this paper highlighted the importance of countermeasures tailored to specific ransomware types, such as crypto and locker variants. From maintaining secure and immutable backups to enforcing strict access controls and endpoint protection, a layered security strategy is essential [7].

Ultimately, defending against ransomware is not the sole responsibility of IT or security teams it requires collaboration across development, operations, and management. By fostering a culture of secure development and investing in preventative practices, organizations can transform from vulnerable targets into hardened environments capable of withstanding modern cyber threats. In this evolving battle, prevention through secure programming is not just an option it is a necessity [8].

References

[1] Ransomware Attacks in Cyber-Physical Systems: Countermeasure of attack vectors through automated web defenses. (2024). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/10713311

[2] Analysis of Ransomware Attack and their Countermeasures: a review. (2022, March 16). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/9751949

[3] Madani, H., Ouerdi, N., Boumesaoud, A., & Azizi, A. (2022). Classification of ransomware using different types of neural networks. Scientific Reports, 12(1). https://doi.org/10.1038/s41598-022-08504-6

[4] GeeksforGeeks. (2024b, October 1). Ransomware explained: How it works and how to prevent it. GeeksforGeeks. https://www.geeksforgeeks.org/ransomware-explained-how-it-works-and-how-to-prevent-it/

[5] GeeksforGeeks. (2022, March 6). Types of ransomwares. GeeksforGeeks. https://www.geeksforgeeks.org/types-of-ransomware/

[6] GeeksforGeeks. (2025, April 7). Types of cyber-attacks. GeeksforGeeks. https://www.geeksforgeeks.org/types-of-cyber-attacks/

[7] Bae, S. I., Lee, G. B., & Im, E. G. (2019). Ransomware detection using machine learning algorithms. Concurrency and Computation Practice and Experience, 32(18). https://doi.org/10.1002/cpe.5422

[8] Ransomware attacks: risks, protection and prevention measures. (2021, September 15). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/9548507

[9] A Systematic study on ransomware attack: Types, phases and recent variants. (2024, March 11). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/10511218