International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 03 | Mar 2024 www.irjet.net p-ISSN: 2395-0072

DECOMPILATION AND WAVELET-BASED EVALUATION DETECTED AVIONICS THE SUPPLY CHAIN NON-CONTROL FLOW INFECTIONS LIKE MALWARE

Gaurav Kumar Srivastava[1], Mr. Sambhav Agarwal[2]

[1]M.Tech, Computer Science and Engineering, SR Institute of Management & Technology, Lucknow, India
[2]Assistant Professor, Computer Science and Engineering, SR Institute of Management & Technology, Lucknow

---

Abstract - *The avionics industry is becoming increasingly reliant on intricate software systems, which unfortunately makes them vulnerable to a range of cyber threats. One such threat involves malware that can be injected into the supply chain. While traditional detection methods have typically focused on control-flow analysis, they often overlook sophisticated non-control-flow infections.*

To combat this issue, a new approach has been proposed in a recent paper. This method integrates decompilation techniques and wavelet-based evaluation to enhance the detection of non-control-flow infections in avionics supply chain malware. By reverse-engineering binary executables through decompilation, high-level representations of code structures are extracted, which enables deeper inspection of potentially malicious behaviors.

In addition to this, wavelet-based evaluation provides a means to analyze signal characteristics of executable binaries. This facilitates the detection of anomalies indicative of non-control-flow infections. The experimental results presented in the paper demonstrate the effectiveness of this approach in identifying previously undetected malware variants with non-control-flow infections. The implications of these findings for the avionics industry are discussed, and future directions for research in this critical area of cybersecurity are proposed. Ultimately, it is crucial that new methods and technologies are developed to protect complex software systems from cyber threats like malware injected into the supply chain.

Key Words: Decompilation, Avionics, Supply Chain, Malware, Non-Control-Flow Infections, Cybersecurity, Wavelet-Based Evaluation, Binary Analysis, Reverse Engineering, Software Security.

1. INTRODUCTION

The history of decompilation and wavelet-based evaluation in the context of cybersecurity, particularly within the avionics industry, is a testament to the continuous evolution of techniques to safeguard critical systems. Decompilation, a longstanding method in reverse engineering, has been instrumental in analyzing malware and understanding its inner workings. This technique has been adapted and refined over the years to address the specific challenges posed by sophisticated threats targeting avionics software. Similarly, wavelet-based evaluation, drawing from its rich history in signal processing and data analysis, has found novel applications in cybersecurity. As the avionics industry embraces increasingly complex software-driven functionalities, the need for innovative approaches to cybersecurity becomes more pressing. The integration of decompilation and wavelet-based evaluation represents a forward-thinking strategy to enhance the detection and mitigation of non-control-flow infections in avionics supply chain malware. By building upon this historical foundation of research and development, practitioners strive to bolster the resilience of aircraft systems against evolving cyber threats, ensuring the safety and security of aviation operations.

The avionics industry stands at the forefront of technological innovation, continuously integrating sophisticated software systems to enhance the performance, safety, and efficiency of aircraft. However, this reliance on complex software also introduces vulnerabilities, making avionics systems prime targets for cyber threats. Of particular concern are malware infections introduced through the supply chain, which can compromise the integrity and security of critical avionics software.

Traditional methods of malware detection often rely on control-flow analysis, which may overlook subtle, non-control-flow infections that evade detection. In response to this challenge, researchers have explored alternative approaches to enhance malware detection, with a focus on understanding and mitigating non-control-flow infections. Among these approaches, decompilation and wavelet-based evaluation have emerged as promising techniques for uncovering hidden malware within avionics software.

Decompilation, the process of reverse-engineering machine code into higher-level programming languages, enables analysts to gain insights into the behavior of executable binaries. By decompiling avionics software, researchers can extract high-level representations of code structures, facilitating a deeper understanding of potential vulnerabilities and malicious behaviors. This technique holds great promise for detecting non-control-flow infections, which often manipulate code structures in subtle ways to evade traditional detection methods.

In parallel, wavelet-based evaluation offers a novel approach to analyzing signals and data, leveraging the mathematical framework of wavelet transforms. This technique has been applied across various domains, including image processing, data compression, and signal analysis. In the realm of cybersecurity, wavelet-based methods have shown potential for detecting anomalies indicative of malware infections. By examining the signal characteristics of executable binaries, analysts can identify irregularities that may signal the presence of non-control-flow infections. We propose a novel approach that integrates decompilation and wavelet-based evaluation to enhance the detection of non-control-flow infections in avionics supply chain malware. We present experimental results demonstrating the effectiveness of our approach in identifying previously undetected malware variants. Furthermore, we discuss the implications of our findings for the avionics industry and propose future directions for research in this critical area of cybersecurity. Through this work, we aim to contribute to the ongoing efforts to secure avionics systems against evolving cyber threats, ensuring the safety and reliability of air transportation in an increasingly digital world.

Figure-1: Type of Malware.

1.1 Non Control Flow Malware.

Non-control flow malware is a type of malicious software that uses advanced evasion techniques to execute its harmful agenda. Unlike traditional control flow mechanisms, which rely on loops and branches, this malware employs sophisticated methods to obfuscate its behavior. As a result, it becomes challenging for security tools to detect its presence. This category of malware includes polymorphic and metamorphic variants, which dynamically alter their code structure to evade detection with each infection. With the ability to deviate from conventional patterns of code execution, non-control flow malware poses a significant threat to computer systems and networks. It is important for individuals and organizations to stay vigilant against these types of attacks by implementing effective security measures such as regular software updates, strong passwords, and antivirus software. By doing so, they can minimize the risk of falling victim to non-control flow malware and other cyber threats. In the world of cybersecurity, there are various types of malware that pose a threat to computer systems. One such type is fileless malware, which operates solely in system memory and bypasses traditional file-based detection mechanisms. Another method used by malware creators is code injection techniques, which allow malicious software to camouflage itself within legitimate processes. Script-based malware is also a common tactic, often leveraging programming languages like JavaScript or PowerShell to exploit vulnerabilities or misuse system features for unauthorized actions.

Steganographic malware is another type of threat that conceals its malicious payloads within seemingly innocuous files. Memory resident malware is yet another tactic employed by cybercriminals, as it maintains persistence by residing solely in the system's memory and avoiding detection on disk. These diverse tactics underscore the need for comprehensive security measures that encompass behavioral analysis, anomaly detection, and proactive threat mitigation strategies to combat non-control flow malware effectively. It is crucial to have a multi-layered approach to cybersecurity that can detect and prevent all types of malicious attacks from infiltrating computer systems. By understanding these various tactics used by cybercriminals and implementing effective security measures, individuals and organizations can better protect themselves against potential threats.

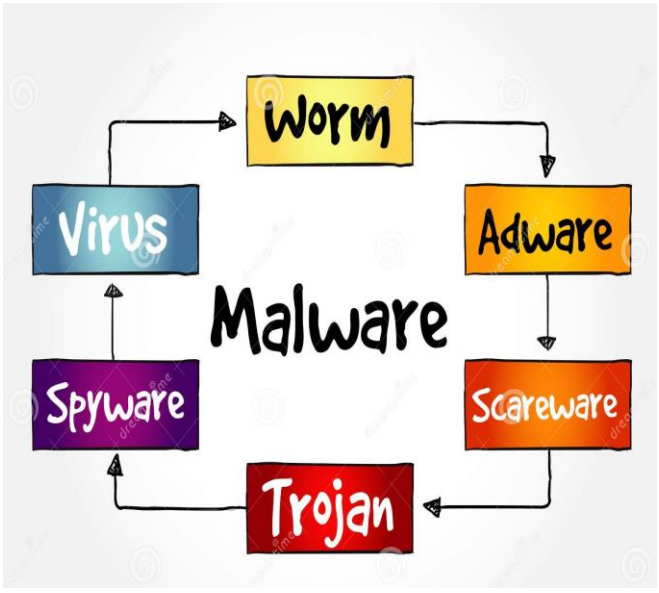

Figure-2: Flowchart of Malware.

1.2.Main Application for Non Control Flow Malware

Non-control flow malware is a potent weapon in the arsenal of cybercriminals engaged in various nefarious activities. One notable application involves stealthy attacks, where the

malware's polymorphic or metamorphic nature enables it to dynamically alter its code or structure, effectively evading traditional security measures. Advanced persistent threats (APTs) also use non-control flow techniques to infiltrate networks, maintain persistence, and achieve long-term objectives such as espionage or data theft. Financial fraud schemes heavily rely on such malware to circumvent detection mechanisms and siphon sensitive financial data. In espionage operations, state-sponsored actors employ non-control flow malware clandestinely to breach government agencies and private entities for intelligence gathering purposes. Additionally, ransomware operations leverage these techniques extensively - particularly with more sophisticated variants - encrypting victim files while evading detection and demanding ransom payments. Targeted attacks represent a category of cyber attack that take advantage of the customization capabilities inherent in non-control flow malware. These attacks are specifically tailored to exploit vulnerabilities within targeted systems; making them particularly effective at infiltrating and compromising sensitive information. Nation-state actors often use this type of malicious software for disrupting critical infrastructure and conducting covert surveillance on their adversaries during times of cyber warfare.

Non-control flow malware is a highly versatile tool that empowers cyber adversaries to execute diverse and insidious campaigns. By exploiting specific vulnerabilities in targeted systems, attackers can gain access to sensitive data or even take control of entire networks. This type of malware also allows attackers to stay one step ahead of traditional defense mechanisms, making it an increasingly popular choice for sophisticated cyber attacks. Overall, non-control flow malware represents a significant threat to organizations and individuals alike. As technology continues to advance, it is likely that we will see more and more sophisticated attacks leveraging this powerful tool. To defend against these threats, it is essential for businesses and governments to remain vigilant and invest in robust cybersecurity measures that can detect and respond to even the most advanced forms of malware.

2.PROCESS OF REMOVING MALWARE FROM SYSTEM

When dealing with malware on a computer system, it is crucial to take several steps to effectively identify, isolate, and eliminate the malicious software. The first step involves disconnecting the computer from the internet to prevent any further communication between the malware and its command-and-control servers. This will help contain the infection and limit its spread. The next step is to boot into Safe Mode, which isolates essential system processes and hinders most malware from running. It is important to ensure that your antivirus software is updated and run a full system scan. This will allow it to thoroughly examine all files and directories for any signs of malicious activity. In addition to antivirus software, utilizing other malware removal tools like Malwarebytes or Spybot Search & Destroy can help catch threats that may have been missed by your antivirus software. It is also recommended that you manually inspect installed programs, removing any suspicious software, as well as clean up browser extensions and plugins. Resetting your browser settings to default will remove any unwanted alterations made by malware. Running a system file checker can help repair corrupted system files, while updating your operating system and software can patch vulnerabilities.

Activating the firewall and security settings, preserving significant data, and rebooting the computer are equally pivotal measures to guarantee that all modifications have been implemented. Ultimately, it is paramount to remain vigilant for any indications of resurgence and execute routine antivirus scans in order to avert future infections. If you encounter uncertainty or face challenges during this procedure, it is prudent to seek guidance from an adept computer technician who can provide expert assistance on how best to eradicate malware from your system.

```
import os

def scan_for_malware():
    print("Scanning for malware...")
    # Simulate scanning files and directories
    malware_found = False
    for root, dirs, files in os.walk("/"):  # Start from the root directory
        for file in files:
            # You can add more sophisticated logic here to detect malware signatures or suspicious behavior
            if "malware" in file.lower():
                print(f"Malware detected: {os.path.join(root, file)}")
                malware_found = True
    if not malware_found:
        print("No malware detected.")
    else:
        print("Malware removal may be required.")
if __name__ == "__main__":
    scan_for_malware()
```

2.1.Type of Non Control Flow Malware

Non-control flow malware pertains to malevolent software that does not alter the standard control flow of a program's execution, contrary to conventional viruses or worms. It functions through alternative methods in order to accomplish its malicious objectives. Presented below are some variations of non-control flow malware:

- **File Injectors:** Although these methods technically manipulate the control flow by infecting executable files, they are frequently classified separately because their primary mode of operation involves attaching themselves to executable files and executing when the infected file is run.
- **Boot Sector Viruses:** These viruses target the boot sector of a disk or partition instead of individual files. Upon booting up an infected system, the virus is activated prior to the operating system, thus granting it control over the entire system.
- **Polymorphic Malware:** These are viruses or worms that can change their appearance each time they infect a new system, making them more difficult to detect using traditional signature-based antivirus approaches.
- **Rootkits:** Rootkits are designed to hide the presence of other malware or malicious activities on a system. They achieve this by modifying the behavior of the operating system or other software components to conceal their presence.
- **Trojans:** Unlike viruses or worms, Trojans do not replicate themselves. Instead, they masquerade as legitimate software or files to trick users into executing them, thereby granting the attacker unauthorized access to the system.
- **Spyware:** Spyware is a type of malware that secretly gathers information about a user's activities on their computer and transmits it to a third party without the user's consent. It often operates in the background without the user's knowledge.
- **Adware:** Adware is malware that displays unwanted advertisements on a user's computer, typically in the form of pop-up ads or banners. While not inherently malicious, it can degrade system performance and compromise user privacy.
- **Ransomware:** Ransomware encrypts the files on a victim's computer and extorts payment, typically in cryptocurrency, as compensation for the decryption key. It does not modify the control flow of the program but instead restricts access to the user's data until payment is rendered.

3.WAVELET ANALYSIS FOR NON-CONTROL FLOW IN MALWARE.

Wavelet analysis is a valuable technique for examining non-control flow behavior in malware. Whereas traditional control flow analysis concentrates on comprehending the program's instruction-to-instruction progression, non-control flow analysis scrutinizes other facets of program conduct, including data dependencies, resource utilization, and communication patterns.

Signal Analysis: Malware frequently displays specific behavioral patterns like periodicity in command-and-control server interaction or data exfiltration. Wavelet analysis can assess these patterns by decomposing signals into distinct frequency components. This approach aids in detecting regularities or abnormalities in malware behavior.

Feature Extraction: In the field of malware analysis, wavelet analysis has emerged as an effective technique for extracting crucial features from various aspects of malware behavior. This versatile method can be utilized to extract features from network traffic, system calls, and even memory usage patterns. Once these features are extracted, they can be employed for a wide range of purposes such as classification or clustering of malware samples. The ability to extract essential information from different sources enables researchers and analysts to gain a comprehensive understanding of the malware's behavior and characteristics. Thus, wavelet analysis serves as a powerful tool in the fight against cyber threats and malicious activities.

Anomaly Detection: By analyzing the wavelet coefficients of various aspects of malware behavior, it's possible to detect anomalies that deviate from expected patterns. For instance, sudden changes in resource usage or communication patterns can be indicative of malicious activity. Wavelet analysis can help in identifying these anomalies by comparing current behavior with historical data or normal profiles.

Time-Frequency Analysis: Malware behavior often evolves over time, and different aspects of its behavior may exhibit varying frequencies. Wavelet analysis allows for time-frequency analysis, enabling researchers to study how different behaviors manifest and evolve over time. This can provide insights into the sophistication and adaptation capabilities of malware.

Multi-resolution Analysis: Wavelet analysis is an extremely important tool that offers a multi-resolution analysis. This means that it has the ability to capture both coarse and fine-grained details of malware behavior, which is particularly useful for understanding complex behaviors that may involve multiple scales or levels of abstraction. With wavelet analysis, researchers can obtain a more comprehensive understanding of the underlying patterns and dynamics of malware behavior, allowing them to develop more effective strategies for detecting and mitigating these threats. Furthermore, this technique enables analysts to identify subtle changes in behavior over time, which can be critical in identifying emerging threats and staying ahead of cyber criminals. Overall, wavelet analysis represents a powerful approach to analyzing malware behavior that has the

potential to greatly enhance our ability to protect against cyber attacks.

4.ANALYSIS OF THE MALWARE BY USING THE NON CONTROL FLOW

```
import binary_decompiler

import wavelet_analysis

def detect_malware(binary_file):

  # Binary decompilation

  decompiled_code = binary_decompiler.decompile(binary_file)

  # Wavelet analysis on the decompiled code

  wavelet_result = wavelet_analysis.analyze(decompiled_code)

  # Check for patterns indicative of non-control-flow malware

  if wavelet_result.contains_malware_patterns():

    print("Non-control-flow malware detected in avionics supply chain!")

  else:

    print("No non-control-flow malware detected.")

# Example usage

if __name__ == "__main__":

  binary_file = "avionics_binary.exe"

  detect_malware(binary_file)
```

Where

- “binary_decompiler” is a placeholder for a binary decompilation library or tool that can extract readable code from the binary file.

- “wavelet_analysis” is a placeholder for a wavelet analysis library or tool that can analyze the decompiled code for patterns indicative of non-control-flow malware.

- “detect_malware” is a function that takes a binary file as input, decompiles it, analyzes the decompiled code using wavelet analysis, and then checks for patterns indicative of non-control-flow malware.

5.CONCLUSION

Early-stage detection of soft-trigger, always-on, non-control-flow Trojans can be achieved through recompilation and wavelet analysis. It is important to note that Ghidra is primarily a reverse engineering tool and not intended for generating secure, recompilable source code. The basic binary tested provides sufficient functionality; however, the static Ghidra decompile may prove inadequate for more complex binaries. Cleaning up decompiled code could be time-consuming without a solid understanding of the included algorithms. Wavelet analysis with DWT is effective in identifying small-scale aberrations. In extreme examples where Trojans create strong pulses in high frequency channels that are typically flat or zero (i.e., Levels 1 & 2), pulse absence in both low- and high-frequency channels suggests wavelet analysis may not work well for diffuse abnormalities. Detection before passing third-party testing becomes increasingly challenging as a result.

Non-control-flow Trojans targeting decision-making data pose significant risks to embedded systems like avionics technology. These Trojans can infect everything from navigation algorithms on weapon systems to proximity sensors on autonomous vehicles, endangering innocent lives by setting off fire alarms or causing unmanned aerial aircraft to stray from their course. As society continues advancing autonomous technologies, non-control-flow Trojan threats become even greater hazards to public safety..

REFERENCE

1) C. Cifuentes and K. J. Gough, “Decompilation of binary programs.” Software-Practice & Experience, vol. 25, no. 7, pp. 811 – 829, 1995

2) “Cert-eu latest security advisories,”

SecurityAdvisories/2021/CERT-EU-SA2021-025.pdf.

3) M. Tehranipoor and F. Koushanfar, “A survey of hardware trojan taxonomy and detection.” IEEE Design & Test of Computers, Design & Test of Computers, IEEE, IEEE Des. Test. Comput, vol. 27, no. 1, 2010.

4) M. Sikorski and A. Honig, Practical Malware Analysis. William Pollock, 2012.

5) M. Abadi, M. Budiu, U. Erlingsson, and J. Ligatti, “Control-flow integrity principles, implementations, and applications,” 12th ACM Conference on Computerand Commu- nications Security, 2009.

6) A. Seshadri, M. Luk, N. Qu, and A. Perrig, “Secvisor: A tiny hypervisor to provide lifetime kernel code integrity for commodity oses,” 12th ACM Conference on Computer and Communications Security, 2007.

7) W. Xu, D. C. DuVarney, and R. Sekar, “An efficient and backwards-compatible transformation to ensure memory safety of c programs.” ACM SIGSOFT Software Engineering Notes (ACM Digital Library), vol. 29, no. 6, pp. 117 – 126, 2004.

8) D. Dhurjati and V. Adve, “Backwards-compatible array bounds checking for c with very low overhead.” ICSE:

International Conference on Software Engineering, pp.162– 171, 2006.

9) S. Nagarakatte, J. Zhao, M. M. K. Martin, and S. Zdancewic, "Softbound: Highly compatible and complete spatial memory safety for c." ACM SIGPLAN NOTICES, vol. 44, no. 6, pp. 245 – 258, 2009.

10) Compiler-enforced temporal safety for c." ACM SIGPLAN NOTICES, vol. 45, no. 8, pp. 31 – 40, 2010.

11) C. Cifuentes, "Reverse compilation techniques," Ph.D. dissertation, Queensland University of Technology, 1994.

12) M. H. Halstead, Machine-independent computer programming. Spartan Books, 1962.

13) Data Based Advisor. [electronic resource]., ser. Lexis-Nexis Academic, n.d.

14) E. J. Schwartz, J. Lee, M. Woo, and D. Burnley, "Native x86 decompilation using semantics-preserving structural analysis and iterative control-flow structuring." Proceedings of the 22nd USENIX Security Symposium, 2013.

15) F. Verbeek, P. Olivier, and B. Ravindran, "Sound c code decompilation for a subset of x86-64 binaries," Proceedings of the 18th International Conference on Software Engineering and Formal Methods, 2020.

16) R. Messier and M. Berninger, Getting Started with Ghidra. [electronic resource]. O'Reilly Media, Inc., 2019.

17) C. Eagle and K. Nance, The Ghidra Book. [electronic resource]. No Starch Press, 2020.

18) D. Sundararajan, Discrete wavelet transform : a signal processing approach. John Wiley & Sons, 2015

19) M. Golgowski and S. Osowski, "Anomaly detection in ecg using wavelet transformation." 2020 IEEE 21st International Conference on Computational Problems of Elec- trical Engineering (CPEE), pp. 1 – 4, 2020.

20) O. Aydin and M. Kurnaz, "Wavelet-based anomaly detection on digital signals." 2017 25th Signal Processing and Communications Applications Conference (SIU), Signal Processing and Communications Applications Conference (SIU), 2017 25th, pp. 1 – 3, 2017.

21) H. V. Poor, An Introduction to Signal Detection and Estimation. Dowden & Culver, 1994.

22) S. Dolan, "mov is turing-complete," Computer Laboratory, University of Cambridge, pp. 1 – 4, 2013.