UMEÅ UNIVERSITY

Talos+ Leveraging Structured Error Handling For Increased Security

Broad software vulnerability mitigation through enabling Security Workarounds for Rapid Response in Java

Melker Henriksson

Melker Henriksson
Spring 2025
Degree Project in Interaction Technology and Design, 30 credits
Supervisor: Prof.Alexandre Bartél
External Supervisor: Anders Sigfridsson, Evelina Malmqvist, Christoph Reichenbach
Examiner: Ola Ringdahl
Master of Science Programme in Interaction Technology and Design, 300 credits

Abstract

The pre-patch window of Vulnerability remains a critical challenge in cybersecurity, leaving systems exposed to exploits before patches are deployed.

The paper *'Talos: Neutralizing Vulnerabilities with Security Workarounds for Rapid Response'* by Zhen Huang et al. proposes an algorithm that detects error-handling code and instruments it with security workarounds, allowing developers to disable vulnerable segments temporarily.

This thesis aims to extend, compare, and verify the claims made in the Talos paper. Since the algorithm's heuristics focus on languages without structured exception handling and experiments are only performed on programs written in c/c++, there is as mentioned in future works potential to expand Talos functionality to cover these languages.

By extending Talos to Java, this research strengthens its viability as a mitigation strategy and provides a proof of concept for its applicability in languages with structured exception handling.

The findings aim to improve the pre-patch vulnerability window for Java and explore potential Vulnerabilities resulting from Talos and so contribute to the broader field of cybersecurity.

Acknowledgements

List of Acronyms and Abbreviations

SWRR	'Security Workaround for Rapid Response'
CIA	'Confidentiality, Integrity and Availability'

Contents

1 Introduction

Software security is the pursuit of engineering software which under attack continues expected function [4]. This pursuit is still of great import as large industries, infrastructure and commercial networks often expect and assume peak performance from software, downtime caused by security breaches can then cause significant financial impact [1].

The nature of some software requires the management of users personal information, for example banking websites or online therapy. A security breach of such systems may lead to disclosure, blackmail or in some cases escalation using spear-phishing techniques [6, 5].

The often sensitive nature of software highlights why security vulnerabilities are a significant problem which is interesting to solve or mitigate.

If a piece of software becomes unable to offer it's users either availability, confidentiality or integrity for services rendered then that software requires a *Patch*. Patching Modifying the code responsible for exposing some vulnerability such that this is no longer possible, usually without a loss in functionality or performance.

It seems reasonable to assume that there would exist a correlation between the time-to-patch and number of exploits performed on a vulnerable piece of software. In fact, in writing about reusable cyberweapons i.e weaponized software code that exploits flaws in software, Carissa G. Hall refers to the pre-patch window of vulnerability as the window of opportunity. In the life cycle of an vulnerability, the implementation of a patch represents death. [3]

If we are interested in contributing to a greater security then reducing the time-to-patch would likely have a significant effect.

The paper *'Talos: Neutralizing Vulnerabilities with Security Workarounds for Rapid Response'* suggests that software could be instrumented allowing for disabling functions through a configuration file or patching. This would allow developers to quickly disallow vulnerable code from running on user end while they're working on patching. Effectively reducing the time-to-patch at the cost of some availability.

This is then in contrast to traditional configuration workarounds exemplified by CVE2021-44228

2 Background

Background

3 Overview

Overview

4 Talos, Talos+

Talos

Talos+

5 Implementation

Implementation

6 Evaluation

Evaluation

7 Conclusion

Conclusion

8 Discussion

Diskussion

References

[1] Scott Coleman. What if you can't patch? - cyber security review, n.d. [Accessed 10-02-2025].

[2] Nesara Dissanayake, Asangi Jayatilaka, Mansooreh Zahedi, and M. Ali Babar. Software security patch management - a systematic literature review of challenges, approaches, tools and practices. *Information and Software Technology*, 144:106771, 2022.

[3] Carissa G Hall. *Time sensitivity in cyberweapon reusability*. PhD thesis, Monterey, California: Naval Postgraduate School, 2017.

[4] G. McGraw. Software security. *IEEE Security & Privacy*, 2(2):80–83, March 2004.

[5] Anthony Minnaar. Cybercriminals, cyber-extortion, online blackmailers and the growth of ransomware. *Acta Criminologica : African Journal of Criminology & Victimology*, 32(2):105, 2019.

[6] Tianhao Xu, Kuldeep Singh, and Prashanth Rajivan. Personalized persuasion: Quantifying susceptibility to information exploitation in spear-phishing attacks. *Applied Ergonomics*, 108:103908, 2023.