UNIVERSITI TEKNOLOGI MARA

A SOURCE CODE PERSPECTIVE C OVERFLOW VULNERABILITIES EXPLOIT TAXONOMY BASED ON WELL-DEFINED CRITERIA

NURUL HASZELI BIN AHMAD

Thesis submitted in fulfilment
of the requirements for the degree of
Master of Science

Faculty of Computer and Mathematical Sciences

August 2015

CONFIRMATION BY PANEL OF EXAMINERS

I certify that a Panel of Examiners has met on 7th April 2015 to conduct the final examination of Nurul Haszeli Bin Ahmad on his Master of Science thesis entitled "A Source Code Perspective C Overflow Vulnerabilities Exploit Taxonomy Based On Well-Defined Criteria" in accordance with Universiti Teknologi MARA Act 1976 (Akta 173). The Panel of Examiners recommends that the student be awarded the relevant degree. The Panel of Examiners was as follows:

Zamalia Mahmud, PhD
Associate Professor
Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA
(Chairman)

Fakariah Hani Mohd Ali, PhD
Senior Lecturer
Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA
(Internal Examiner)

Halabi Hasbullah, PhD
Associate Professor
Director of Intelligent Cities
Universiti Teknologi Petronas
(External Examiner)

SITI HALIJJAH SHARIFF, PhD
Associate Professor
Dean
Institute of Graduate Studies
Universiti Teknologi MARA
Date: 19th August, 2015

ABSTRACT

Despite various works for more than three decades, C overflow vulnerabilities is still a major security issue, as it has contributed to more than 30% of all recorded vulnerabilities and has been the root cause of many successful exploits. One of the main causes lies in the C software developers themselves, who inadvertently introduced these vulnerabilities due to their lack of understanding of vulnerabilities being the security loophole. To educate them, researchers have constructed C overflow vulnerabilities taxonomies. However, most of these taxonomies are memory-based, focuses on symptoms upon vulnerability triggered and did not describe the appearance of vulnerabilities in coding, which subsequently, prevented software developers from understanding the vulnerabilities and writing safe codes. There were also works done previously on source code-based taxonomies but they were too broad with ambiguous classes and failed to describe clearly from software developers point-of-view. Currently, there is no source code-based taxonomy constructed with criteria of well-defined taxonomy resulting in difficulty to apply taxonomy as foundation and references in writing secure codes. Therefore, the objective of this research is to construct a well-defined C overflow vulnerabilities exploit taxonomy from source code perspective. To achieve that, reviews on numerous reports, advisories and publications related to C overflow vulnerabilities, analysis methods and tools, and relevant classifications and taxonomies were meticulously performed. It was followed by reclassification of well-defined criteria, which was used to construct C overflow vulnerabilities exploit taxonomy from source code perspective. The taxonomy was then evaluated for both relevancy against well-defined criteria and as well as the effectiveness of static analysis tools. The results suggested that the taxonomy facilitates the understanding of software developers in classifying and detecting C overflow vulnerabilities and the selected five static analysis tools require further improvement to enable the tools to detect from three to four classes to all available C overflow vulnerabilities classes. The significances of this study are the constructed well-defined taxonomy of C overflow vulnerabilities exploits consisting of 10 classes with three new classified classes; i.e. Memory Functions, Variable Type Conversion and Pointer Scaling/Mixing, and methods to evaluate taxonomy in accordance to well-defined criteria.

TABLE OF CONTENTS

CHAPTER ONE
INTRODUCTION

1.1 INTRODUCTION

The introduction of computer to the world simplifies many of our daily tasks in many ways. Human used to be quite simplistic in nature that they used to regard computers as a genius and flawless technology. The stigma dominated for quite some time, until the emergence of 'Morris worm' which inevitably diminish the thought. Then, human learned how dangerous, unreliable and vulnerable a computer system could be, especially when computer becomes vulnerable to making mistakes. A single error in a computer program could result in catastrophic consequences. Opportunists took advantage of these flaws, and consequently they exploited the flaws and caused many unwanted hardships. In addition, a catastrophe as a result from these errors may demand massive financial resources to mitigate. Subsequently, people started to emphasize more care in computer systems, particularly in analytical methods and tools to prevent these vulnerabilities. However, the battle is still ongoing.

Recently, emphasis has focused more towards preventing vulnerabilities on various fronts. To stymie these vulnerabilities at the root source, researchers have also began focusing on the system and software developers themselves, by making improvements on their knowledge, understanding and attitude of these vulnerabilities. These include possible vulnerabilities of their source codes during development and deployment. Thus, the main purpose of this study is to improve their understanding of software vulnerabilities as the initial step towards preventing future vulnerabilities.

This chapter lays the foundation and baseline of the whole thesis, and is arranged as follows – section 1.2 explains software vulnerabilities in general. It is followed by section 1.3 that addresses the problem statement and the critical research gap. Relevant research questions are listed in section 1.4, followed by research objectives in section 1.5. In section 1.6, the significances of this research is emphasized. Necessary and relevant assumptions, scope and limitations are listed in sections 1.7 and 1.8. Section 1.9 lists the thesis contributions and finally, section 1.9 summarizes the chapter.