AOSP Java Secure Code Audit

DISSERTATION

Submitted by

RAHUL.B CB.EN.P2CYS14010

in partial fulfillment for the award of the degree of

MASTER OF TECHNOLOGY
IN
CYBER SECURITY

TIFAC-CORE IN CYBER SECURITY
AMRITA SCHOOL OF ENGINEERING
AMRITA VISHWA VIDYAPEETHAM
COIMBATORE - 641 112

May 2015

AOSP Java Secure Code Audit

DISSERTATION

Submitted by

RAHUL.B CB.EN.P2CYS14010

in partial fulfillment for the award of the degree of

MASTER OF TECHNOLOGY
IN
CYBER SECURITY

Under the guidance of

Prof. Prabhaker Mateti
Associate Professor
Computer Science and Engineering
Wright State University
USA

TIFAC-CORE IN CYBER SECURITY
AMRITA SCHOOL OF ENGINEERING
AMRITA VISHWA VIDYAPEETHAM
COIMBATORE - 641 112

May 2015

AMRITA VISHWA VIDYAPEETHAM
AMRITA SCHOOL OF ENGINEERING,COIMBATORE -641 112

BONAFIDE CERTIFICATE

This is to certify that this mini project report entitled **"AOSP Java Secure Code Audit"** submitted by **RAHUL B. (Reg.No : CB.EN.P2.CYS14010)** in partial fulfillment of the requirements for the award of the **Degree of Master of Technology** in **CYBER SECURITY** is a bonafide record of the work carried out under my guidance and supervision at Amrita School of Engineering.

Dr. Prabhaker Mateti
(Supervisor)

Dr. M. Sethumadhavan
(Professor and Head)

This mini project report was evaluated by us on.....

INTERNAL EXAMINER

EXTERNAL EXAMINERS

AMRITA VISHWA VIDYAPEETHAM
AMRITA SCHOOL OF ENGINEERING,COIMBATORE
TIFAC-CORE IN CYBER SECURITY

DECLARATION

I,RAHUL.B. (Reg.No: CB.EN.P2.CYS14010) hereby declare that this mini project report entitled **"AOSP Java Secure Code Audit"** is a record of the original work done by me under the guidance of **Prof. Prabhaker Mateti**, Assistant professor, Amrita Vishwa Vidyapeetham and this work has not formed the basis for the award of any degree / diploma / associateship / fellowship or a similar award, to any candidate in any University, to the best of my knowledge.

Place : Coimbatore

Date : Signature of the Student

COUNTERSIGNED

Dr. M. Sethumadhavan
Professor and Head, TIFAC-CORE in Cyber Security

Acknowledgement

At the very outset, I would like to give the first honors to the **Almighty** who gave me the wisdom and knowledge to complete this mini project.

I express my gratitude to my guide, **Prof. Prabhaker Mateti**, Associate professor, Wright State University, USA, and my co-guide, **Mr. Praveen. K**, Assistant professor, TIFAC-CORE in Cyber Security, for her valuable suggestion and timely feedback during the course of this mini project. It was indeed a great support from her that helped me successfully fulfill this work.

I would like to thank **Dr. M. Sethumadhavan**, Professor and Head of Department, TIFAC-CORE in Cyber Security, for his constant encouragement and guidance throughout the progress of this mini project.

I convey special thanks to my friends for listening to my ideas and contributing their thoughts concerning the project. All those simple doubts from their part has also made me think deeper and understand about this work.

In particular, I would like to thank all the other faculties of TIFAC-CORE in Cyber Security and all those people who have helped me in many ways for the successful completion of this mini project.

Contents

List of Tables

List of Figures

ABSTRACT

Keyboards: Android, Secure Coding, Conformance to secure coding

1

Introduction

1.1 Secure Software

1.2 Problem Statement

1.3 Motivation

1.4 Aim

1.5 Organization

2

Background

2.1 Abstract Syntax Tree

2.1.1 Visitor pattern

2.1.1.1 Advantages

2.1.1.2 Applications of Visitor Pattern

2.2 Secure Coding Guidelines

2.2.1 MISRA

2.2.2 CERT Secure Coding

2.2.3 OWASP

2.3 Static Analysis of Source Code

2.3.1 Type checking

2.3.2 Style checking

2.3.3 Program understanding

2.3.4 Bug finding

2.3.5 Security review

3

Proposed System

3.1 Source code to AST

3.2 Verify rules with AST

3.3 Audit AOSP

3.3.1 Applicability of the rule

3.3.2 Semi-algorithmically explanation

3.3.3 Semi-automatic quick fixes

3.4 Rewrite Code

4

Related Works

4.1 The CERT C Secure Coding Standard

4.2 The CERT Java Secure Coding Standard

4.3 All Your Droid Are Belong To Us A Survey of Current Android Attacks

4.4 Tool support for automated CERT C Secure Coding Standard certification

4.5 A Language for Examining Abstract Syntax Trees

4.6 Secure Programming with Static Analysis

4.7 Graph Matching Algorithm on Java Source Code

4.8 Java plus Pattern Matching

4.9 Eclipse Java development tools

4.10 JavaParser- A case study

4.11 JTransformer- A case study

5

Details of A Solution

5.1 About CERT C Secure Coding Standard

5.1.1 Issues Not Addressed

5.1.2 Identifiers

5.1.3 Priority and Levels

5.1.4 Rules and Recommendations

5.2 Solution for CERT rules

5.2.1 Rule 1

5.2.1.1 Description

5.2.1.2 Solution

5.2.1.3 Applicability of the rule

5.2.1.4 Semi-algorithmically explanation

5.2.1.5 Semi-automatic quick fixes

5.2.2 Rule 2

5.2.2.1 Description

5.2.2.2 Solution

5.2.2.3 Applicability of the rule

5.2.2.4 Semi-algorithmically explanation

5.2.2.5 Semi-automatic quick fixes

5.2.3 Rule 3

5.2.3.1 Description

5.2.3.2 Solution

5.2.3.3 Applicability of the rule

5.2.3.4 Semi-algorithmically explanation

5.2.3.5 Semi-automatic quick fixes

6

Implementation and Result

6.1 AOSP Audit Report

6.2 Rule 1

6.2.1 Implementation Details

6.2.2 Challenges

6.3 Rule 2

6.3.1 Implementation Details

6.3.2 Challenges

6.4 Rule 3

6.4.1 Implementation Details

6.4.2 Challenges

7

Conclusion and Future Work