Design and Development of an Agent Based Centralized Tool for Analyzing and Managing Security Enhanced Linux Policies

Ishara Madushanka Kularatna

MS18904710

A THESIS

SUBMITTED TO

SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY

IN PARTIAL FULLFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

AT FACULTY OF GRADUATE STUDİES AND RESEARCH

(January 2018 Batch)

December 2021

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

_____

Mr. Samantha Rajapaksha

Approved for MSc. Research Project:

_____

MSc. Research Project Co-Ordinator, SLIIT

Approved for MSc:

_____

MSc. Programme Co-Ordinator, SLIIT

Declaration of Originality

This is to certify that the work is entirely my own and not of any other person, unless explicitly acknowledged (including citation of published and unpublished sources). The work has not previously been submitted in any form to the Sri Lanka Institute of Information Technology or to any other institution for assessment for any other purpose.

Signature _____

Ishara Madushanka Kularatna

Date _____

Abstract

Design and Development of an Agent Based Centralized Tool for Analyzing and Managing Security Enhanced Linux Policies

I.M Kularatna

MSc. in Information Technology

Supervisor: Mr. Samantha Rajapaksha

December 2021

Security Enhanced Linux also known as the SELinux, facilitates and includes an extensible Mandatory-Access-Control which is called, “MAC” structure/system built within the Linux kernel. An application or a process life cycle which runs as a user (UID or SUID) has the specific authority to access objects such as files, sockets and other processes with Linux’s default Discretionary-Access-Control (DAC). SELinux prescribes the access and the progress privileges of each user, application, process, and the files on the system and administers the communications of these elements utilizing a security strategy that determines how severe or indulgent a given Red Hat EnterpriSELinux establishment ought to be.

However, due to its constraints such as, not being user friendly, having too complicated policies and convoluted policy description language, are limiting the implementation of SELinux policies in the IT industry. As a result, there is only few research available on the subject of UI based policy management tools and even those research have limitations such as, inability to remotely manage a host/server, manual documentation and inability to monitor the systems automatically from a dashboard.

In order to overcome the said research gap and problems, this research will implement a system, using a web-socket technology that facilitates ability to conversation in full duplex through a just one TCP connection. This system is included with a web socket-agent, which can be installed in server endpoints and has the ability to change SELinux policies, a web-socket server: which can do live communication with the agent to perform policy changes, UI component: to manage policies using user interface and a database component to store policy details.

Acknowledgements

First and foremost, I would like to express my gratitude to my M.Sc. Research project supervisor, Mr. Samantha Rajapaksha, who is a senior lecturer and the program coordinator of the M.Sc. in IT at the Faculty of Graduate Studies and Research at SLIIT, for his unwavering support and assistance throughout the research project.

Also, I'd want to express my heartfelt appreciation to my ever-loving wife, mother, and father for their unwavering support during my study endeavor.

Furthermore, everyone who lent a hand and assisted in the completion of this res earch study was recognized with thanks for everything they had done to help pro gress the project.

Table of contents

Declaration of Originality iii
Abstract iv
Acknowledgements vi
Table of contents vii
List of tables viii
List of figures viii
Chapter 1 1
Introduction 1
1.1: Context and Background 1
1.2: Thesis Structure 6
1.3: Aim and Objectives 7
1.3.1 Problem Statement (Definition) 7
1.3.2 Research Questions & Research Objectives 9
Chapter 2 11
Literature Review 11
2.1: Introduction 11
2.2 Literature review of previous solutions 11
2.3: Literature review of technologies that might be applied in this research project 35
Chapter 3 41
Methodology 41
3.1: Introduction 41
3.2: Methods and techniques 41
Chapter 4 75
Testing and Evaluation 75
4.1: Introduction 75
4.2: Testing and Evaluation 75
Chapter 5 80
Discussion 80

5.1: Introduction 80
5.2: Discussion 80
Chapter 6 81
Conclusions 81
6.1: Conclusions 81
6.2: Recommendations and Future Development 81
Appendices 85

List of tables

Table 1 : Capability comparison of the research work that has been analyzed during the literature review 8
Table 2 : SELinux security policy analysis tools comparison [6] 30
Table 3 : Comparison of WebSocket push technology vs Traditional Web push technologies 36
Table 4 : Features included in the proposed tool 42
Table 5 : Features included in the proposed tool cont. 43
Table 6 : Technologies used in the research project 47
Table 7 : Test Case 01 – Change SELinux Mode to Enforcing or Permissive 76
Table 8 : Test Case 02 – Check SELinux mode (disabled, permissive, or enforcing) 77
Table 9 : Test Case 03 – Check all SELinux boolean values 78
Table 10 : Test Case 04 – Check a specific SELinux boolean value 78
Table 11 : Test Case 05 – Change a specific SELinux boolean value 79

List of figures

Figure 1 : SELinux Decision Process[2] 1
Figure 2 : Editing selinux configuration file using Nano 2
Figure 3 : SELinux Man page 3
Figure 4 : Fedora OS - SELinux Management software 4
Figure 5 : SELinux Management - Fedora OS 4
Figure 6: Suggested SELinux policy analysis tool framework [3] 13
Figure 7: Data structures that reflect information that has been extracted [3] 14
Figure 8 : Flow of constructing access vectors [3] 15
Figure 9 : Basic Policy Management Architecture 22
Figure 10 : An example namespace hierarchy 26

Figure 11 : The SELinux System's decision-making core architecture [6]. 28
Figure 12 : Typical SELinux analysis tool structure 3.5[6] 28
Figure 13 : SELinux policies update – Manual process [7]. 32
Figure 14 : Proposed architecture of distributed SELinux management tool 33
Figure 15 : UI of the implemented tool[9]. 34
Figure 16 : Proposed system architecture diagram 49
Figure 17 : Flow chart diagram for policy management client algorithm 52
Figure 18 : Request JSON formatted text 55
Figure 19 : Response JSON formatted text 55
Figure 20 : Development phase of Polict management client 56
Figure 21 : Establishing the WS server connection 57
Figure 22 : Set Enforcing function in policy management python client 57
Figure 23 : Flow chart diagram for WebSocket server algorithm 59
Figure 24 : Making the connected clients online/ offline 60
Figure 25 : Project Structure - web socket server 62
Figure 26 : Functions which requests SELinux configuration items from policy management client 62
Figure 27 : Functions which sends responses coming from the policy management client which will direct to UI. 63
Figure 28 : pre-defined functions. 64
Figure 29 : Application configuration items 64
Figure 30 : ER Diagram. 65
Figure 31 : Designing of the ER 66
Figure 32 : Forward engineering the ER 66
Figure 33 : Flow chart diagram for UI component 70
Figure 34 : Project Structure 72
Figure 35 : Sample UI 73
Figure 36 : Password hashing / encryption 74