RedShL Intrusion Detection System

Course: 3809ICT Applied Network Security | Date: 03/05/2019

Group:

* Taylor Radke (s5094922)
* Lucas Geurtjens (s5132841)
* Sebastian Perry (s5132483)

Table of Contents

[Introduction 3](#_Toc7805405)

[How to Program Works 3](#_Toc7805406)

[Program Modules 3](#_Toc7805407)

[Results 4](#_Toc7805408)

[Summary 4](#_Toc7805409)

# Introduction

The program was designed to detect changes to a given set of files and directories. It evaluated this by creating a verification file that could be used as a base case comparison to potential changes. In practical use, the program could eventually be used to detect potential intrusions and file tampering within the system, heightening integrity.

# How to Program Works

+ [Requires an update to better discuss the option features. Will improve this after I study the program more.]

Firstly, the program receives a directory. This directory will act as a base case state for the verification to occur. Next, a verification file is created. It includes the following information of each file/directory:

* Inode (Basic info about a file/directory)
* Name
* Basename (Essentially the filename, taken from the location path)
* Absolute Path
* Owner ID
* Group ID
* Access Privileges
* Time Last Modified
* Last Time Accessed
* SHA1 (Exclusively for files, not directories) (Originally MD5 used, but not as strong)

The information from each file /directory is stored on a line of the verification file. From this, the user is prompted to begin verification of files. This is where the user is given the ability to make changes to files and directories. After proceeding, the verification process begins and the lines of the verification file are compared against the current information of the files/directories. Once this has finished, the user will be prompted with the number of failed cases. With this, the user can also display which explicit files did not match their verification file (hence have been altered).

# Program Modules

|  |  |
| --- | --- |
| Options | Description |
| -c name | Create a verification file called “name”. |
| -o name | Display results of the comparison check on screen and saved to an output file. |
| -t directory | Choose a directory to track (skipping user input) |
| -v file | Select a pre-existing verification file to use against the current state |
| --help or -h | Display a help message explaining how to use the program |

Table 1: Command Line Options

# Results

+ [Discuss if everything in the program works correctly]

+ [Adding actual testcases into the table below]

|  |  |  |
| --- | --- | --- |
| Input | Changes Made | Output |
| [Directory containing these files/directories] | * Removed x * Added y * Modified z | 3 Changes were found |
| [Directory containing these files/directories] | * Opened file y | No changes were found |
| ... | ... | ... |

Table 2: Test Case Results

# Summary

+ [A brief conclusion about what was created and if it worked successfully or not]