MalwareScanner User Manual

Henry Karagory, Jacob Grove, Ian Yang

Table of Contents

1. Introduction

1.1 Purpose

The Linux command line provides access to powerful tools for different aspects of process management. MalwareScanner was created in the interest of making the holistic task of process management, easy, robust, and automated.This document is the authoritative source of information in the use of MalwareScanner program.

1.2 Scope

This manual begins with a breakdown of the abilities of the program. Instructions for first-time users about installation and use of the program are included in Section 3: Installing, Starting and Stopping the Program. Section 4: Advanced Configuration contains some additional instructions for advanced users, such as how to manually edit the configuration files or run a precompiled demo.

1.3 System Organization

1.3.1 Primary Documents

The main driver of the program is contained in malware_finder.c in the root directory. The structure of the main program is divided into user input (get_options.c), node processing (utility.c), and process management(mware_sys_call.c). This file makes calls out to methods specializing in each of these functions.

1.3.2 System Call

As part of the original project resources, some code was provided related to creating the kernel module and loading the system call. As part of our project we altered the content of the system call to kill processes based upon the output of our program. The system call can be found in the project repository at lkm-syscall-v1/lkm/syscall.c. The name of the system call function is mware_scan_syscall.

2. Key Features

2.1 Identifying Malware

At its core, MalwareScanner centers around identifying processes by a regular expression and queuing them to be deleted or ignored. Regardless of the method of input or precondition, MalwareScanner takes search criterion in as a regular expression, flags processes that match that expression, and either kills or differs these processes at the user's discretion.

2.2 Multiple Modes of Usage

The first objective of the program on startup is to identify under what conditions to search for malware. By default, the user will be asked explicitly for what regular expression that they would like to screen running processes. The user may elect to include this information on the command line as an additional argument. Doing so bypasses some regular steps of user input.

2.3 Whitelist Configuration Settings

MalwareScanner provides the ability to maintain a list of search criteria over multiple sessions. During each session, the user will be asked if they would like to use and append these saved criterion. If so, the session information will be appended to the running list. If not, the previously saved information will be discarded and the current session will begin a new record of criterion.

3. Installing, Starting and Stopping the System

3.1 Installing the System

Clone the repository by entering the following in your terminal application:
git clone https://github.com/HenryKaragory/MalwareScanner.git.

3.2 First Time Users

Navigate from the command line to the root directory of the project. If this is your first time using the program, compile the program by running the "make" command from the

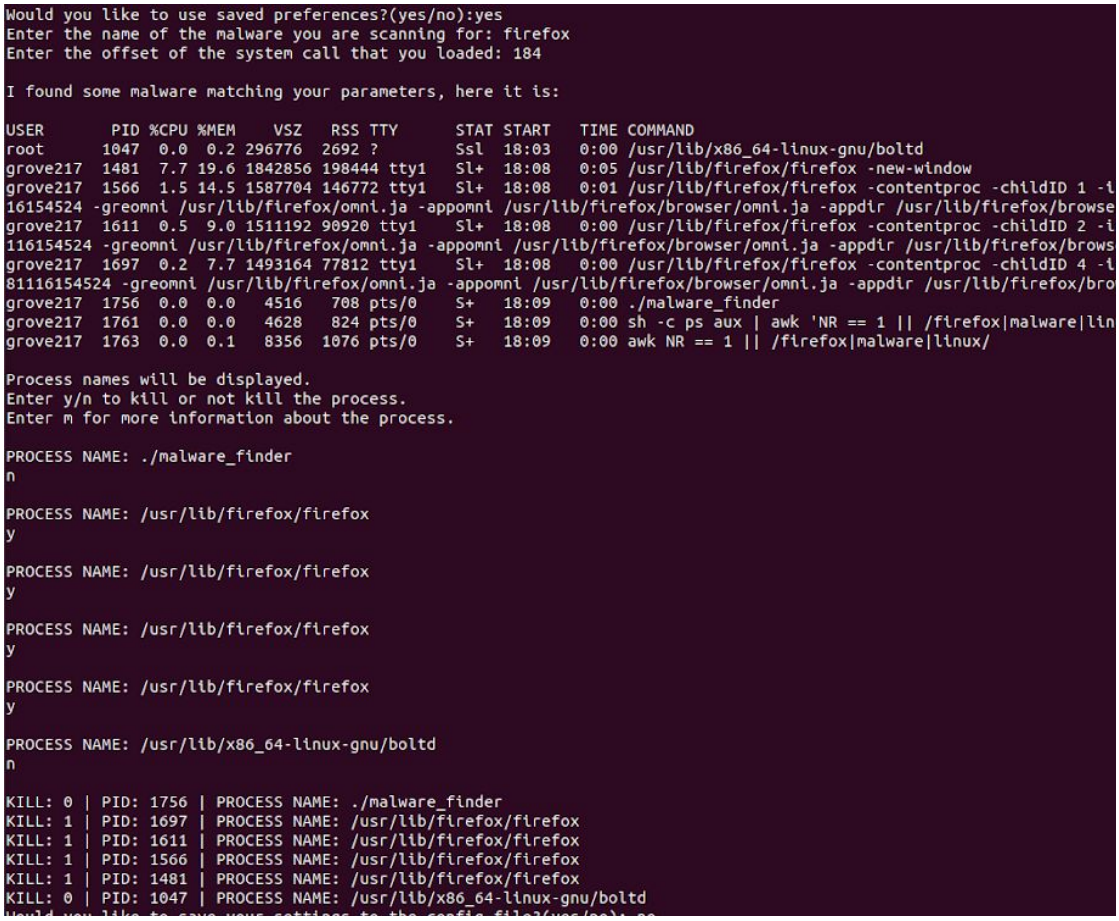
terminal. The program is ready for general use when “malware_finder” exists in the project directory. Your directory should look like the one below.

3.3 Access Control

Several core features of this program require access to the “sudo” command. It is required that whatever account be using this program have the ability to run “sudo” commands.

3.4 Starting the System

Run the program by typing “./malware_scanner” followed by an appropriate regular expression. This field may be left empty, in which case, the user will be specifically prompted for a regular expression. The program will then initiate the process of identifying and classifying malware based on the search criteria. Note that it is safest to enter the value 184 when prompted for a system call table offset. This value must be changed inside the module files if a different number would be used. In the release version of the program, user input is required to say whether flagged candidates should be terminated or not. An example of the flow of the program is shown below.



3.5 Stopping the System

The program will terminate normally after iterating through all of the candidate malware. At any time, if the user would like to stop the program while it is running. They may input CTRL+C from the command line, and the program will exit safely.

4. Advanced Configuration

4.1 Demo Instructions

The the root directory for the project, in a folder labeled “demo”, there are some additional files that can be used to automatically test features of the program. The release version of the demo program creates two processes labeled “malware” and “not_malware” and then provides prompts for identifying and removing these processes. The demo code itself is well documented such that a competent programmer could create a different sequence of calls and inputs with a little effort. An example output from the demo is shown below.

```
======= LOADING THE KERNEL MODULE ======


======= COMPILING THE KERNEL MALWARE ======

make: Nothing to be done for 'all'.

======= RUNNING THE MALWARE ======


======= RUNNING THE MALWARE SCANNER ======

======= MALWARE SCANNER DEMO =======

Would you like to use saved preferences?(yes/no):no
Enter the name of the malware you are scanning for: malware
Enter the offset of the system call that you loaded: 184

I found some malware matching your parameters, here it is:

USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      2452  0.0  0.0   4376   716 pts/0    S+   18:18   0:00 ./malware
root      2453  0.0  0.0   4376   708 pts/0    S+   18:18   0:00 ./not_malware
root      2454  0.0  0.0   4516   704 pts/0    S+   18:18   0:00 ./malware_finder
root      2459  0.0  0.0   4628   828 pts/0    S+   18:18   0:00 sh -c ps aux | awk 'NR == 1 || /malware/'
root      2461  0.0  0.1   8356  1104 pts/0    S+   18:18   0:00 awk NR == 1 || /malware/

Process names will be displayed.
Enter y/n to kill or not kill the process.
Enter m for more information about the process.

PROCESS NAME: ./malware_finder
n

PROCESS NAME: ./not_malware
y

PROCESS NAME: ./malware
y

KILL: 0 | PID: 2454 | PROCESS NAME: ./malware_finder
KILL: 1 | PID: 2453 | PROCESS NAME: ./not_malware
KILL: 1 | PID: 2452 | PROCESS NAME: ./malware
```

4.2 Modifying the Configuration File

There are two vital pieces of information in the “config” file. The first is a PRINT_MWARE_INFO option which is used for debugging purposes and should not be modified. The second is a MALWARE_NAME field which contains a singular regular expression. This field can contain any valid regular expression. At the user's discretion, this field can be changed to modify the search criterion automatically loaded into the program on startup.