Assignment 2

Host-based IDS in Shell script

3413ICT: Network Security

1 Introduction

The main objective of this assignment is to apply the theoretical knowledge learned in the class on intrusion detection system towards developing a practical system. Your task is to create a host-based intrusion detection application in shell script. The purpose of the application is to help an administrator in monitoring the file systems on a computer to detect changes to files, i.e., to detect possible intrusions. The application has two parts: i) creation of a list (usually a text file) of file names and their attributes, and ii) stepwise testing of all the files included in the list.

One of the widely used commercial host-based IDS is Tripwire. And an open source version of Tripwire is also available. A number of similar applications also exist, for example GNU-licensed software AIDE (Advanced Intrusion Detection Environment). Other file integrity tools can also be found on the Internet, e.g., FCheck and sXid.

2 Terminologies

A verification file is a text file containing a list of names of files and directories and their properties. This file is the output generated by your application. This verification file has to be generated before checking for possible intrusions. During verification, the entries of the verification file are compared to the actual file system. If an entry matches the current properties of a file or directory in the file system, verification of that file/directory succeeded. The properties describe about different kinds of files and links, for example regular files, symbolic link files, or directory files.

3 Requirements

The application must be written in shell script that can run in Bourne shell (sh). Code, or parts of code, that can only be executed by other shells (e.g. bash or csh) is not accepted. Other script languages (Perl, Ruby, etc.) or programming languages (C, C++, Java, etc.) are also not accepted.

3.1 Data collection

You must create a directory including a number of files and directories. Several pieces of information about files and directories (e.g., file type, access control, word count, owner, last date of modification etc.) are to be collected by your script. The collected information should be stored in a text file which is later used by the script to verify the files, directories, etc., included in the output file.

MD5 checksums should be calculated for all regular files. The following information should be collected about all regular files, directory files and symbolic links:

- ✓ full path and file name
- ✓ file type, one of the appropriate strings: regular file, directory, symlink
- ✓ access mode, in text format (e.g. -rwxr--r--)
- ✓ owner id and group id
- ✓ time of last modification and last file status change

Your script may change the modification time of files and directories during execution. This is not acceptable and is one of the challenges that your script will have to manage.

3.2 Command line options

The application must support at least the following command line options:

-c name	Create a verification file called 'name' also display a message "File created".
-o name	Display the results on the screen also save the outputs to an output file

3.3 Allowed tools

It is allowed to use the Bourne shell and the standard tools included in the Ubuntu. The following commands/programs are examples, and recommendations, of such programs:

- ✓ access, awk, chflags, echo, file, less, ls, md5, more, printf, sed, sort, touch, wc
- ✓ Manual pages are available for all of these programs. Use the man program to access them (read man's manual if you have not used it before: type man man).
- ✓ Recommended reading are the manual pages of sh and builtin.

4 Report

The report should include a cover page (with submission details, name, id, date, course code etc.), an introduction, explanation of different modules of the program, findings, and a summary (length of the report should be between 2 and 4 pages, not counting the cover page and appendix).

5 Submission Guidelines

You can complete the assignment individually or in a group of 2 (maximum). For group submission, the report must contain a table specifying the level of contributions and efforts of each group member (see template for details). The report submission deadline is 4.00pm, **Friday, 20th May, 2016**.

6 Marking Scheme

The following marking scheme would be applied to Assignment 2.

Implementation & Testing	Report	Demonstration	Total
60%	20%	20%	100%

7 References

Tripwire, (open source version)

http://www.tripwire.org

http://sourceforge.net/projects/tripwire

AIDE (Advanced Intrusion Detection Environment)

http://www.cs.tut.fi/_rammer/aide.html

Bourne Shell Programming, Steve Parker

http://steve-parker.org/sh/sh.shtml

Bourne Shell Programming, Andrew Arensburger

http://www.ooblick.com/text/sh