Intrusion Detection using Tripwire in UNIX Operating System

Muhammad Saqib Javed

saqibjaved@vu.edu.pk

Virtual Univerity of Pakistan

Abstract:

This paper outlines the detection procedure of intrusion as provided by Tripwire tool along-with the enhancement to be made in design and implementation to achieve the optimum performance level of Tripwire. It operates on effective and swift performance mechanism in order to report the system admin about the possible intrusion detected in the system with security assurance in cutting edge computer structures, which is very important in order to provide the integrity and reliability of information. In order to deal with this emerging issue in the historical research, the basic level of security is provided in the system by making enhancement in the existing tripwire tool for UNIX file system. The proposed methodological working of tripwire makes it more effective by adopting reliable and instantaneous mechanism to deal with specifically anomalies and unwanted access in the UNIX file system.

Keywords:

Message Digest, Index Node, Secure Hash Algorithm, Sandboxing.

1. Introduction

Intrusion detection is the main concern for system administrators nowadays, so that to ensure the smooth running of events. The preferred tool for intrusion detection at large scale is tripwire which is essentially used as a reliability checking platform, especially carved for the UNIX system that provides system administrators complete capacity, so to closely screen out the file systems aimed at addition, deletion, modification using certain technique. The Tripwire tool was formally confined and open for use on November 2nd 1992, where it is vigorously used at hundreds of sites around the globe.

Tripwire is basically an instrument that supports the UNIX system admin to plaid for some sort of variations which are through on discerning set of files. So that to monitor the file systems for addition, deletion and modification of files and directories [1], along with their databases. Alerts been generated for the system admin for any kind of changed or tainted files. After this the system admin very much capable to yield counteractive actions in an apt and timely manner. The UNIX file system delivers procedure for complete storage and accessibility of data along with programs in a system. The Evidence resides in the file system has got some key value that should be scrutinized for illegitimate and unforeseen variations, in order to defend the system conflicting to intrusion. The file stored in the UNIX file system comprises the data of users along with data of several applications and system executable file with databases. For attacker to attack it comes to become a normal goal of an attacker to intrude something. An intruder could change system files in order to access it in future for illegal access.

Core obligation of the system admin is to deeply screen out that which archives contains files have been changed or fiddled and where to take obligatory actions. For instance, the UNIX System administrator's faces great difficulty in detecting the damages in infected files and in ensuring the veracity and the signatures of the file system. The standard checklist contains checksum histories, which are obtainable in UNIX systems are not reliable enough and ready to use at times. So, to help System Admins in detecting the abnormal behaviour in the system due to anomaly, Tripwire actually plays a vital role in spotting irregularities in the UNIX file system. It always aimed at the behavioral change which diverges after typical system response.

When talking about the tool used in parallel for same problem which is facing by huge number of business analysis and site owners referred to as intruders in Sand-boxing tool [2], it helps in detecting intrusion in windows as well as Unix/Linux file system. Sandboxing is another tool used for intrusion detection but is smaller in scope as compared to Tripwire in a sense that sandbox focuses on the fact that it should comprise all the archives of file needed to execute the application. The complete list of files needed to be gathered in order to inculcate with the task and it is not an easy task by any mean. The applications to be determined by utilities, such as lpr uses pooled libraries, loadable segments, configuration archived files along with the operating system attached files, which is also used by numerous C library utilities. Fundamentally, the sandboxing system certainly not handles the system program with better rights, so that it works outside the sandbox, in order to prevent the application from accumulating enough evidence about the system files; such as /etc/password (contains configurable password file) and /etc/hosts (contains configurable host file), it is just needed to be replaced with disinfected varieties, so that it exactly contains the required information, in order to accomplish its tasks and which is mostly time consuming as it is not very much reliable as tripwire. A smart home is basically a system of sensors with devices, combined into a universal system, which is capable of performing certain actions for solving everyday problems without human participation. The Internet of Things (IoT) is the mechanism that presently operates smart homes [8].

2. WHY TRIPWIRE?

2.1 Motivation

In the era of advancement, we need a superficial tool which helps us to tackle the challenging needs of today with respect to intrusion detection, by focusing on our scope. We took tripwire as a best intrusion detector as it is not platform dependent also it can work at enterprise level as well on the basis of log entries. The technique proposed to get the optimum level of existing tripwire working is by adding cryptographic checksum in the existing tripwire config, so that the performance will be enhanced and reliability should be achieved.

2.2 The Consequential Challenges

The practices used to monitor the file systems, so to monitor the potential changes which includes, maintaining the checklists of files, the evaluation copies and extensive history of backup files used for exigency. A comparative study is performed for three Machine Learning algorithms that were implemented on the NSL- KDD dataset for the IDS system. To obtain the optimal accuracy, it is required to select the appropriate set of features in a large dataset. Feature extraction calculated on the basis of ANOVA F-test and Recursive Feature Elimination (RFE).

2.3 Cryptographic Techniques in Tripwire

The tripwire supports, RSA Data Security, MD5, MD4 and MD2 the Snefru (which is actually the Secure Hash Function in Xerox) and at last the SHA (the Secure Hashing Algorithm).

2.4 Performance Measurement Comparison with Security

In normal Scenario only one checksum to be calculated for each file would be sufficient to detect the variation. Aimed at the purpose of attaining speed, it is highly recommended to use an easy way to calculate the checksum. Nevertheless, the easiest way to compute signatures is to conquest a resolute attacker who coveted to do some intrusion in existing stable system. Tripwire embraces with six identical cryptographic algorithms along with twofold frequently used (Cyclic redundancy check) CRC routines.

Through implementing the evasion setup of recording twofold signatures together, such as the MD5 and the Snefru. Hence, for each database entry, this gives very robust commitment and assurance that a file used has not been hampered through. If let's say that the intruder is somehow successful in tampering the system file by adding suitable stuffing characters, in order to reconstruct the already created both checksums lacking to alter the size of the existing file. Both MD5 and snefru are considered as strong message authenticated codes.

2.5 Features of Tripwire

Tripwire tracks on peak of the disparities of UNIX accentuating platform manageability in the sense of accessing it anywhere. In order to highlight the database compactness, the database archives are encrypted in standard format. The files which are generated by tripwire can run and used to read on any platform. Tripwire is independent because it is likely to make run program without dependent on any external and hypothetically susceptible platform. For example, if we see that an integrity assessor depends upon a utility, it must respond uneven if it is disrupted.

2.6 Proposed enhancement in existing Tripwire Procedure

Mainly the focus is on UNIX file system as according to the theme of the paper the working of tripwire in UNIX file system works in a complete sequence and pattern. In a complete system, files are constantly scanned for intrusion using detection tool that reports each changed file formerly to system for timely cure.

The Tripwire engenders yield that is easy to test by letting discerning archives to be scrutinized. Generally, the archives which are nominated aimed for checking are the files which are not easily updated at whichever alteration to the files already checked actually alarms for us. For example, modifications made in the log files are predictable, with variation made in the inode number, In meekest terms, Tripwire generates a protected catalogue of file and directory features plus their signatures, where focus is on which signature to use for formerly comparison against the newly created database to realize if a file or directory has changed some way, on the basis of which any differences can be reported in the proposed system. The system administrator manages Servers operated in UNIX file system. Tripwire practices numerous checksum Algorithm's, where the proposed one is cryptographic checksum to achieve better functionality.

3. Current Working of Tripwire:

The objective for intrusion detection is to recognize the unlicensed use, which is characterized as a misuse and false use of systems via in-house computer system users and outside invaders. Tripwire aims to sense and alert system admin for altered, added, and deleted files using versatile method. The high level diagram of tripwire flow is shown below:

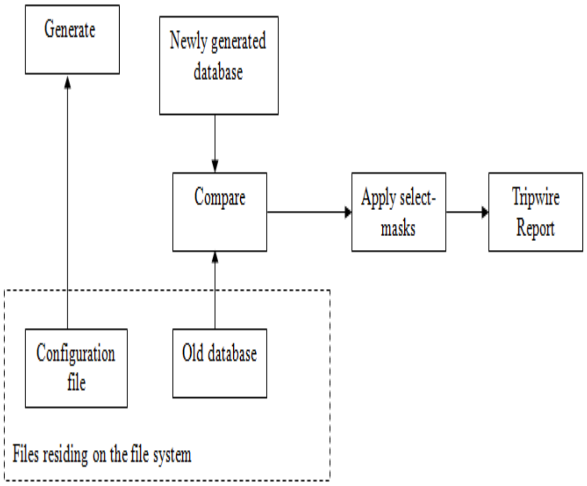


Fig: 3 Tripwire flow in UNIX file system

The Tripwire tool was specifically designed to monitor files and directories on a UNIX system for keeping track of changes that could come from unauthorized modifications such as, software failures, malware, or intrusions. Over the time, a number of other uses were also identified that includes verifying updates and ensuring consistency with a baseline model [9].

3.1 Tripwire Procedural working

The basic working of existing tripwire in UNIX file system is that it mainly focuses on the added and deleted files rather than changed file as shown in Figure 1. The configuration file contains the working of trip wire on the basis of logs generated against every iteration. The database created after every iteration matches the one previously generated in order to find the added and deleted files in UNIX system [3]. The figure 1 gives the foremost components of the whole procedure. It actually demonstrates that in what way the tripwire practices the dual inputs by matching the configuration file along with the previously taken database, so to generate the report showing any positive or negative intent in the Operating system configuration files.

A Configuration file referred to as tw.config file consists mainly of entire objects that needed to remain in check. It covers complete list of files attached with the directories and also the associated attributes involved with the event in order to safely ignore them though doing the evaluations. For example, aiming at just few files, the time stamp can be overlooked, in order to make contrast. The complete list containing attributes which possibly will be ignored is referred to as an assortment mask for that particular entity. For example, the accesses made in the config files as shown in below stream.

Table: 3.1 UNIX file system Configuration

File / Dir	Selection Mask (Label)	Comments
/var/tmp	R	//only directory
etc/utmp	L	//dynamic files
/etc	R	//all files under /etc

The selection mask looks like: +pinugsm12-a, [4] it refers to the fact that the Tripwire must report any changes made in the authorization modes, such as the inode number with the number of links used for authorization, the id of user, the id of group and the file size, modification done along with timestamp and digital signatures used.

The core component of tripwire working is basically the database of config files that consists primarily of signatures which are formerly engendered, aimed at the anonymous accesses made in the config file. This file containing database of whole event as already created by Tripwire must contain complete list of log entries along with its complete filenames, the inode number, the

doi:10.20944/preprints202302.0220.v1

attribute values, and comprehensive signature information, the assorted masks used with associated file entry.

Tripwire is efficiently designed to be run as a standalone program deprived of privileges. Though, this tool would not deliver any sort of ways to make easy alteration in the system. This actually helps the users running Tripwire using their own secluded file fixture. Machines must work in a single user mode, so that it will be easy to manage also when installing the database makes it work even better. Likewise, if the system previously has some wiretaps before constructing tripwire then we must assume that the tripwire we installed is actually too late to report those interfering [5]. So somehow or the other it is essential to reinstall the operating system along with all system files, so to cater the problem and make it move towards it's resolution. These surely cause inconvenience. Particularly, the inadequacies of the distinctive checklist patterns, which are the stream of files linked with checksums that may be very hard to uphold as due to space and time issue which may cause the system to perform the operation slow. Intruder simple can make some changes to the contents of the files without simply altering the already generated checksum aimed for these files.

4. Proposed Tripwire Framework

The tripwire suggested model consists of different modes of task. The internal specific configuration file contents initiate every operational mode. The proposed working of Tripwire tool is explained below:

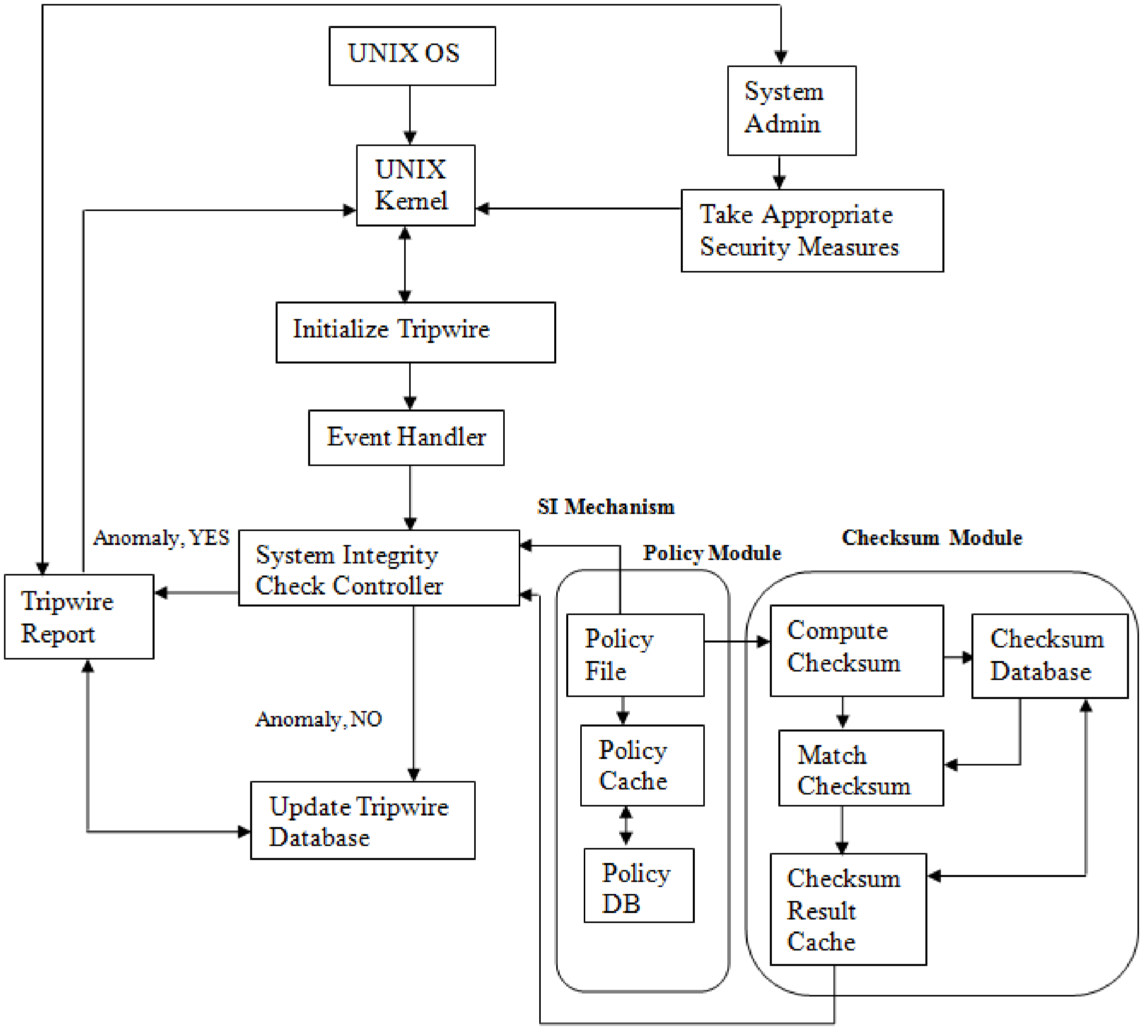


Fig: 4 The Enhanced Tripwire Framework Ensuring Swift Performance at Optimum Level.

4.1 Tripwire Initialization

Before the initialization of tripwire there are certain activities which is performed starting from the start of UNIX operating system the kernel is loaded. After this kernel initiates different libraries and directories needed to perform normal operating system functionality then after loading the kernel of UNIX operating system along with different link libraries and directories initialize tripwire, now if tripwire creates problem in initialization kernel provides support to make it start working, where event handler works to carry the configured and initialized tripwire to its routine working, which is to initialize the tripwire database so to cross match the last updated tripwire database with the recent report taken from system admin.

4.2 Integrity Checking Approach

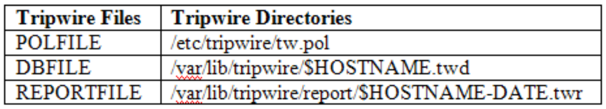
The Integrity checking is basically composed of two level operations as handled by system integrity check controller, where event handler of UNIX file system handles that integrity check controller functionality. The integrity check controller works in two way mode. First it computes the intrusion detection functionality based on policy module and checksum module, and then it takes result from both modules to report about anomaly found or not to system admin via tripwire reporting module with updating in tripwire database.

4.3 Policy Module of Tripwire

The controller points to the policy file, where policy file contains essential rules along with related files, which should be tested by Tripwire using certain directions, which indicates the harshness of a violation. The core procedure of Tripwire is organized with the help of configuration file and policy file respectively; the mentioned file before use needs to be encrypted and properly signed for security reasons. The files typically lie in /etc/tripwire [3] directory. The policy file access policy cache for getting stored instructions from policy DB where ultimately the updated policies are stored.

The plain text versions of the two core files used for integrity checking are called twcfg.txt and twpol.txt with the encoded versions of the two core files used for checking integrity and different versions of sign referred to as tw.cfg (tripwire configuration) and tw.pol (tripwire policy) files respectively. The plain text file configuration files version as described above contains key values [7] including the following required variables as when in table mentioned below:

Table:4.3 Tripwire Files with Description



Tripwire Files	Tripwire Directories
POLFILE	/etc/tripwire/tw.pol
DBFILE	/var/lib/tripwire/$HOSTNAME.twd
REPORTFILE	/var/lib/tripwire/report/$HOSTNAME-DATE.twr

- **Tripwire binary files**

The basic file which is used for the database initialization and [7] integrity checking of the system containing files with updation in the database along with policy. Starting from **Twadmin** (tripwire admin), is actually used for creating configuration files along with replacement and printing of a file, which contains policy along with encryption functions.**Twprint** (tripwire print), is used to print the generated reports along with associated database with tripwire in readable format. **Siggen** (Signature generation), actually generates numerous hashes those are very much supported by tripwire for checking the files integrity.

4.4 Checksum Module

In this all important module, we require to compute checksum by carrying policy file values of the existing state of config files so to cross match with the existing checksum database.

4.5 Compute Checksum

The checksum of configuration specific files are computed according to the instructions from policy file.

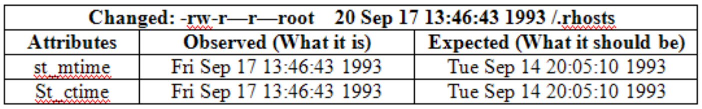
4.6 Database Update Approach

The checksum calculated on the basis of cryptography used is saved in checksum database where the main focus is on config files of UNIX system where the files are used to operate for many system operations and events. When files alter for valid reason, the updation of database is necessary to guarantee uniformity of database. The configuration records for the checked files, referred as specific files are redeveloped and due to this a novel database created, which is used in next iteration as an updated database of tripwire. Again, this database must be linked with checksum, as result of computed checksum first recorded to the cache, where the result of checksum is stored and from cache result it is forwarded to checksum database and also to integrity check controller for decision making about anomaly.

As tripwire practices on an encrypted database because the database holds secure information that attracts the intruder to attempt. Preferred way is SHA (Secure Hash Algorithm) also known as cryptographic checksum to be used to detect intrusion effectively and in procedural manner. The tripwire controller again reads the configuration file, hence redevelops the database which is based on the configuration file innards. The already generated database is cross matched with existing database to generate complete repository of the files which are added and deleted. Those files which are changed are operated under the policy file which is catering and also the likeliness of attack on sensitive files. It also caters group-ID’s to be focused based upon the core functionality of policy cache works with policy file [6].

The printed filenames and the attributes value which is stored along with comparison is shown in the table below:

Table: 4.6 File’s Attribute Integrity Check Example



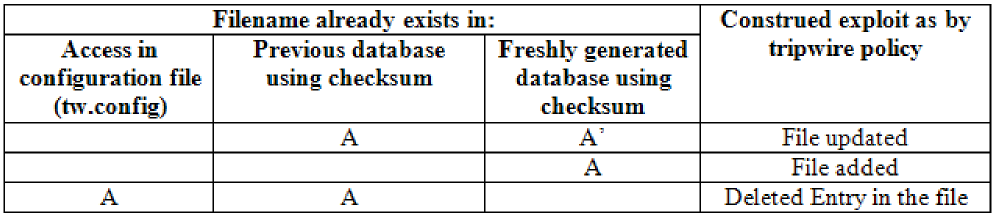
Changed: -rw-r—r—root 20 Sep 17 13:46:43 1993 /.rhosts		
Attributes	**Observed (What it is)**	**Expected (What it should be)**
st_mtime	Fri Sep 17 13:46:43 1993	Tue Sep 14 20:05:10 1993
St_ctime	Fri Sep 17 13:46:43 1993	Tue Sep 14 20:05:10 1993

4.7 Integrity Check Controller

In this mode the complete list of changes required to generate the reporting file used through the integrity checking approach for monitoring and reporting each change. Tripwire inquires the system admin through integrity check controller that whether the file ought to be updated.

Addition of files in the existing computed checksum is much more intricate than apprising files. For the event of file updation and deletion, the permission granted to file is substituted in the database catalogue by a novel access spangled the existing position of the file. File addition is rather more complex task, subsequently there should be no easy entry provided for the configured file. To resolve this issue, the tripwire just picks the closest progenitor entry from the configuration file and start working on it. The table below is an overview of tripwire database:

Table: 4.7 Tripwire Database Overview



Filename already exists in:			Construed exploit as by tripwire policy
Access in configuration file (tw.config)	Previous database using checksum	Freshly generated database using checksum	
	A	A’	File updated
		A	File added
A	A		Deleted Entry in the file

The variation found in the database can be classified into three main cases as shown above. For every case, a suitable action will be taken based on the entry in the configuration file and the integrity check of config file, where the match took place between the old checksum database and newly engendered database and then reported to the controller of integrity checking module.

4.8 Tripwire Report

Tripwire automatically generates its report containing the number of files scanned in UNIX file system along with complete detail that under what policy this scan taken place along with the date and time with complete configuration as taken from tripwire database. This report is generated by tripwire at every iteration whether anomaly found or not, it must report to UNIX kernel first so that if any stored procedure is there to implement itself also report to system admin in parallel for possible update or for anomaly cure referred to as intrusion detection. Then Administrator takes appropriate measures for fulfilling the prevention process with the help of UNIX kernel.

Conclusion:

Integrity of the system is the concern for having smooth running of the system by taking care of any unwanted intrusion either coming from within the system or outside of the system. Therefore, to address the problem the technique used to secure the UNIX file system is tripwire, which is validated to achieve the optimum level. It is therefore concluded that by using cryptographic checksum, we enhance the existing tripwire functionality by introducing the matching procedure of checksum database based on policy file along with effective reporting.

Future Work:

The existing work ensures the reliability by timely detection and reporting of the intrusion. The performance of tripwire is enhanced by the proposed working in this paper. However, the time consumption problem need to be addressed by taking control of the backlog entries in the updater repository as a seed knowledge base resides in tw.config file, which can be fruitful in the future papers if sort out cautiously.

References:

[1] Kim, G. H. and E. H. Spafford (1994). "Experiences with tripwire: Using integrity checkers for intrusion detection."

[2] Prevelakis, V., & Spinellis, D. (2001, June). “Sandboxing Applications”. In*USENIX Annual Technical Conference, FREENIX Track* (pp. 119-126).

[3] Belur, M. (2002). “Tripwire: A File System Integrity Checker For Intrusion Detection”. *CS265: Computer Cryptography and Security*, 1-5.

doi:10.20944/preprints202302.0220.v1

[4]. Kim, G. H. and E. H. Spafford (1994). The design and implementation of tripwire: A file system integrity checker. Proceedings of the 2nd ACM Conference on Computer and Communications Security, ACM.

[5] Kim, G. H., & Spafford, E. H. (1994). “Writing, supporting, and evaluating tripwire: A publically available security tool”.

[6] David A. Curry. *UNIX System Security: A Guide for Users and System Administrators.* Addison-Wesley, Reading, MA, 1992.

[7] Donovan, B. O. (2004, September). Retrieved 09 09, 2013, from http://linuxgazette.net: http://linuxgazette.net/106/odonovan.html

[8] Gorbatiuk, V. O., & Gorbatiuk, S. O. (2023). Method of detection of http attacks on a smart home using the algebraic matching method. PROBLEMS IN PROGRAMMING, (3-4), 396-402.

[9] Eugene H. Spafford. 2022. Tripwire: Pioneering Integrity Scanning for Cybersecurity. In ACSAC 2022, Dec 5–9, 2022, Austin, TX. ACM, New York, NY.

[10] Ibrahim, Z. K., & Thanon, M. Y. (2021, January). Performance comparison of intrusion detection system using three different machine learning algorithms. In 2021 6th international conference on inventive computation technologies (ICICT) (pp. 1116-1124). IEEE.