

5-2018

Comparison of Intrusion Detection Systems/ Intrusion Prevention Systems – A Selection Criterion

Priyanka Mutyala

St. Cloud State University, pmutyala@stcloudstate.edu

Follow this and additional works at: https://repository.stcloudstate.edu/msia_etds

Recommended Citation

Mutyala, Priyanka, "Comparison of Intrusion Detection Systems/Intrusion Prevention Systems – A Selection Criterion" (2018).
Culminating Projects in Information Assurance. 49.
https://repository.stcloudstate.edu/msia_etds/49

This Starred Paper is brought to you for free and open access by the Department of Information Systems at theRepository at St. Cloud State. It has been accepted for inclusion in Culminating Projects in Information Assurance by an authorized administrator of theRepository at St. Cloud State. For more information, please contact rswexelbaum@stcloudstate.edu.

Comparison of Intrusion Detection Systems/Intrusion Prevention Systems –

A Selection Criterion

by

Priyanka Mutyala

A Starred Paper

Submitted to the Graduate Faculty of

St. Cloud State University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science in

Information Assurance

May, 2018

Committee Members:
Dennis Guster, Chairperson
Lynn Collen
Balasubramanian Kasi

Abstract

Most of the devices and systems nowadays are complex connected devices that perform critical functions. Security in these devices is a critical task and of the highest importance. The protection of the data is mandatory for any organization, so there is a demand for the security mechanism to protect the data. Security is a challenging issue that should be taken into consideration when designing and building business-based web applications, as well as during its maintenance stage. Security can be provided to a system in various ways at different layers. This can be done either by an Intrusion Prevent System (IPS) or Intrusion Detection Systems (IDS). Usually deployed in a network to monitor the traffic, these systems use their own methodology to prevent, mitigate, and arrive at conclusions.

The main objective of the paper is to discuss various kinds of IPS/IDS in detail, and their uniqueness which makes them stand out for various reasons. An additional discussion point will indicate which IDS/IPS can be used according to the security requirement, their functionality, and performances with their effectiveness to stop the malicious activity over a computer network. Reasons to choose a specific IDS/IPS will be listed. One of the high-level objectives of the paper is to create awareness about the availability of IDS/IPS and information on which one to choose for their requirements.

Table of Contents

	Page
List of Tables	5
List of Figures	6
Chapter	
I. Introduction	7
Introduction.....	7
Problem Statement	7
Nature and Significance of the Problem	9
Objective of the Research	10
Research Questions and/or Hypotheses	10
Definition of Terms.....	10
II. Background and Review of Literature	12
Introduction.....	12
Malware Attacks/Threats	12
IDS/IPS	13
III. Methodology	15
Introduction.....	15
Design of the Study.....	15
Tools and Techniques	15
IV. Conclusion and Future Work.....	49
Host-based vs. Network-based IDS	49
Future Work	53

References.....54

Appendix.....56

List of Tables

Table	Page
3.1 Methodology for Research Questions.....	16
4.1 Generic Comparison of Host-based IDS and Network-based IDS.....	50
4.2 IDS Comparison.....	52

List of Figures

Figure	Page
1.1 Victims of Cybercrime.....	8
1.2 Increasing Trend in the Cybercrimes.....	9
3.1 High-level Operational Model of Tripwire.....	17
3.2 BRO – IDS Architecture.....	36
3.3 TCP Wrapper Architecture.....	41

Chapter I: Introduction

Introduction

An intrusion can be defined as any set of actions attempting to compromise the integrity, confidentiality, or availability of a resource [6]. As the technology is growing, the concern for its security is growing as well. There is an enormous number of intrusions caused by both external and internal intruders in various ways. Security has become one of the major topics of concern. Many organizations dealing in e-business should have their security tightly implemented as any downtime caused by intrusion can incur considerable loss of revenue. This situation can also lead to the possible loss of customers, as shoppers may see themselves prone to security attack, which could put a company out of business. Therefore, a reliable security system is vital. There are many companies, both large scale or small scale, that do not see security as an important issue. This is due to the expense of implementing a security system, as well as a general lack of awareness. Subsequently, there is a need to understand and compare different security systems available by summarizing the advantages, disadvantages, and necessary requirements to decide in selecting a security system. Currently, there are many Intrusion Prevention Systems and Intrusion Detection Systems being deployed in the network or host to help understand and mitigate any malicious activities.

In this paper, a selection of open source Intrusion Detection Systems will be compared, and different types will be further discussed to provide an insight and awareness of them and identify the factors affecting such decisions.

Problem Statement

A vast number of people are vulnerable to security breaches but do not know how they can avoid it. Currently, there are various IDS/IPS known, used, and actively being built. For any

big company, common user, or vendor, a problem arises when they want to decide upon which IDS/IPS should be implemented on their network or host as per their requirements and the features of the IDS/IPS. Therefore, an awareness and a contrast on an IDS and its features must be well known to understand and compare different security systems available. Both external and internal cyber-attacks have been growing at an alarming rate. According to a CPI/FBI survey, 59% of companies surveyed had one or more attacks reported [5]. Almost 8% of those companies reported 60 or more internal incidents. The main issues that need to be addressed in preventing and detecting attacks are as follows: what the basic problems of insider attacks are, how IDSs can help solve this problem, and how an internal IDS should be deployed using various IDS technologies. Below are the statistics showing the increase in fraud day-by-day.



Figure 1.1. Victims of Cybercrime

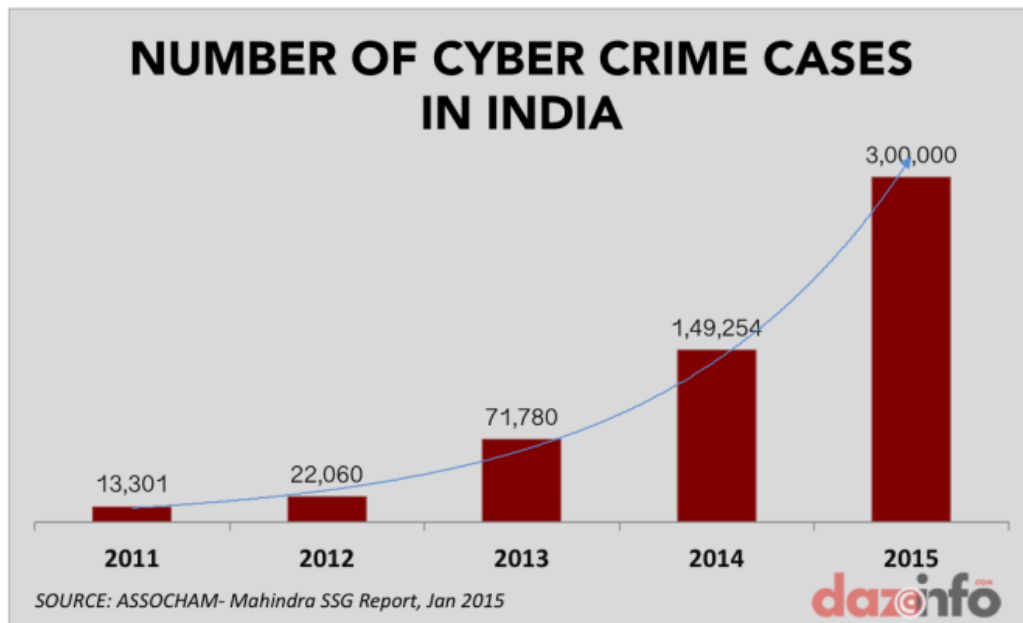


Figure 1.2. Increasing Trend in the Cybercrimes

Nature and Significance of the Problem

It is important to know the growing concerns and vulnerabilities of a security breach, and if anything can be done about it. In a narrower view, implementing any IDS/IPS requires prior knowledge of various IDSs and their methodologies. There should be a criterion for selecting an IDS/IPS. This paper not only helps people become aware of current issues but also educates them on how to be pro-active, and what actions can be taken. This study will be useful mainly for those who would like to implement an IDS/IPS and obtain the knowledge to choose the most suitable system in accordance to their respective requirements. Every IDS/IPS is unique in its own way, though they have many features in common, which makes them preferable for certain requirements. Below are a few statistics.

Current scenario:

- 1 of 3 organizations do not have an information security policy and do not know if a 3rd party can access data/policies already in place.

- Only 20% of organizations have an adequate security system in place that prevents or identifies attacks.
- Less than one-fourth of the companies do network vulnerability scans at least once in a quarter.

Cost of security breaches:

- There is a 15% percent increase in attacks since last year, which accounts to \$4 million.
- Downtime increased to 8 hours for 31% of organizations that are impacted.

Most of these attacks happen due to a lack of awareness.

Objective of the Research

- 1) Compare various IDSs currently trending and provide knowledge of selection criteria according to requirements needed to provide security at various layers.
- 2) Create awareness of IDS/IPS.

Research Questions and/or Hypotheses

- Which IDS/IPS can be implemented according to given requirements?
- To what extent can this help identify or mitigate a security issue?
- What are the criteria to conclude upon for selecting an IDS/IPS?
- How effective can the IDS/IPS be?
- How broad and comprehensive the IDS/IPS detection capabilities are?
- How well can the IDS/IPS incorporate an understanding of context to improve its functioning?

Definition of Terms

Intrusion Detection System (IDS) is a software or hardware component that automates the intrusion detection process. It is designed to monitor the events occurring in a computer

system and network and responds to events with signs of possible incidents of violations of security policies.

Intrusion Prevention System (IPS), on the other hand, is the technology of both detecting of intrusion or threat activities and taking preventive actions to seize them. It combines the knowledge of IDS in an automated manner.

Chapter II: Background and Review of Literature

Introduction

This chapter discusses different attacks on the system, what an Intrusion Detection System is, prevention systems, and a brief overview of these systems.

Malware Attacks/Threats

Malware is an intrusive software whose main purpose is to perform malicious activities on the computing device. For example, computer viruses, worms, Trojan horses, ransomware, spyware, adware, and scareware are all examples of malicious programs. These programs have been a serious problem for past decade. Below is Malware which is basically threats:

Backdoor. Backdoor is a mechanism which allows the access of a computer program by bypassing the security mechanisms. A developer may install Backdoor for troubleshooting a program, among other purposes [2]. The hackers can detect a Backdoor and give themselves access as a part of an exploit. Backdoor can be used to gain root access, i.e. to gain the superuser privilege while remaining undetectable from anti-malware detectors. Few examples of root exploits are rage-against-the-cage [13] and ginger break gain full control of the device.

Botnet. A botnet is a group of computers connected to the internet that has been set up to transmit spam or viruses to other computers connected to the internet while the owner of the computer is unaware of the situation. These compromised computers are called bots, and the person who is controlling these bots is called the Bot-master [12]. Series of commands control a bot, and a group of bots is known as a botnet. Botnets can be controlled by sending commands leading to the launch of Denial of Service to download the malicious applications automatically.

Worm. A worm is a type of malware computer program which makes copies of itself and often uses a network to spread to other computing devices [14].

Spyware. Spyware is a type of malware that is installed on the computing device with the knowledge of the user. Spyware is used to collect the private information of the user and send it to the hacker (remote server) in real time. This also degrades the computing device performance.

Trojan. Trojan performs harmful activities without consent or knowledge of the users. Trojans leak the confidential data or steal the sensitive information like passwords. A trojan can be installed on any system, and data from the system can be taken, modified, damaged, etc.

All the attacks discussed can happen at different levels of a network model. In order to safeguard the system, IDSs at different layers are needed.

IDS/IPS

An Intrusion Detection System (IDS) is a tool or mechanism to detect attacks against a system or a network by analyzing the activity in the network or in the system itself. Once an attack is detected, the system will log information about it and report an alarm. The detection mechanisms in an IDS are either signature-based or anomaly-based.

Signature-based detections match the current behavior of the network against predefined attack patterns. Signatures are pre-configured and stored on the device, with each signature matching a certain attack. In general, signature-based techniques are simpler to use. The signature of each attack must also be stored. This requires special knowledge of each attack, and storage costs grow with the number of attacks. This approach is more static and will not be able to detect new and unknown attacks unless their signature is manually added to the IDS.

Anomaly-based detection tries to detect anomalies in the system by determining the ordinary behavior and using it as a baseline. Any deviations from that baseline are considered an anomaly. On one hand, anomaly-based systems can detect almost any attack and adapt to new environments; on the other hand, these techniques have rather high false positive rates (to raise

an alarm when there is no attack), as deviations from the base might be normal. Also, they have comparatively high false negative rates (no alarm when there is an attack), as attacks might only show a small deviation that is considered within the norm.

The IDS system are categorized as follows depending upon the method used for detection of attack:

Network based IDS: monitors network traffic for particular network segments or devices and analyzes the network. IDSs verify unsuccessful attacks by being placed outside of a firewall, which can see any rejected attacks that can never hit its host

Host-based IDS: monitors the characteristics of a single host and the events occurring within that host for suspicious activity. IDSs verify success/failure of an attack using logs containing the events that actually happened.

Choosing an IDS/IPS according to the requirements necessitates a bit of research and is sometimes complicated. To help with this, there are several papers comparing different IDS/IPSs. One such work published compares two open source network intrusion detection systems, Snort being one and the other being BRO IDS [10]. Also, the article Open Source Intrusion Detection Tools: A Quick Overview [11] compares open source IDSs.

Chapter III: Methodology

Introduction

This chapter describes the methodology in which IDSs are implemented, and a brief comparison has been conducted.

Design of the Study

Data collection. A qualitative approach is being followed as this paper provides insight into IDS/IPS and also makes the average end user aware of pre-requisites to consider when selecting a particular IDS. This paper does not deal with the statistical analysis of any IDS/IPS, though few of the open source IDSs are implemented to compare. This is because there are many IDS/IPS available, and it is not feasible to test them all.

Tools and Techniques

Tools used in this paper are VMWare Workstation, Wireshark to analyze the packet capture, Net Scan Tools Pro packet generator, Multiple IDS such as Bro IDS, Tripwire, TCP Wrappers, Snort.

Hardware and software environment. Installation of Ubuntu on VMWare workstation was required to install BRO IDS and implement it. Few other IDS have been implemented in a similar way.

Table 3.1. Methodology for Research Questions

Research Question /Objective	Approach / Design
1.What to implement an IDS?	Implement them locally
2.How each IDS is useful?	Test few real time security attacks on the IDS
3.Can we identify any security attack?	Implement
4. How can any new attack be mitigated or detected?	Research with various new viruses and Analyze the existing protection systems and check its effectiveness

IDS that are implemented in this paper are:

- 1) Tripwire – Host-based IDS
- 2) BRO IDS – Network-based IDS
- 3) TCP Wrappers – Host-based IDS
- 4) Snort – Network-based IDS

Tripwire. Tripwire is an integrity checking security system used for UNIX systems. It gives the system admins the ability to monitor file systems for any changes such as addition, deletion, ownership, and any slight modification done to the files. It is considered to be scalable, flexible, and easily configurable to enjoy widespread use.

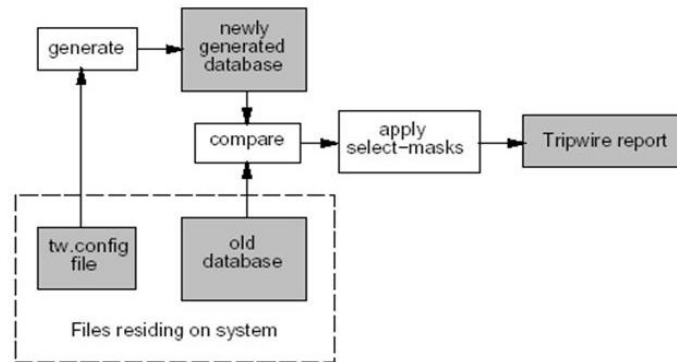


Figure 3.1. High-level Operational Model of Tripwire

A high-level model of Tripwire operation uses two inputs: a configuration describing the file system objects to monitor, and a database of previously generated signatures putatively matching the configuration [9]. The configuration file consists of all the files that the administrator wants to monitor. A database file is generated by tripwire containing the filenames that are selected, inode attribute values, information and the configuration file that generated it.

Tripwire has 4 operation modes:

- 1) Program operation
- 2) Database initialization mode
- 3) Integrity checking mode
- 4) Database update mode and interactive database update mode

In database initialization mode, a state of current file systems is generated for all the files that are mentioned in the configuration file `tw.config`. Every entry in the database contains the filenames, etc as mentioned above.

In integrity checking mode, Tripwire reads the configuration file and generates a new database before comparing with the existing image that was captured earlier to see if the state of

any of the configuration files have been modified. If they have been modified, then a new report is generated.

When there are changes made to files and they no longer match the state of the captured database, updating the database becomes mandatory. This can be done by either database update mode or interactive database update mode, where the administrator is prompted for each file or directory that has been changed and is asked if that change is approved or not. Tripwire implementation is below for the standard file system that is existing on the VM.

➔ Initializing the database:

The basic way to initialize the database is by running:

```
sudo tripwire --init
```

```
priyanka@ubuntu:/etc/tripwire$ sudo tripwire --init
```

Database generated:

```
### No such file or directory
### Continuing...
### Warning: File system error.
### Filename: /root/.ICEauthority
### No such file or directory
### Continuing...
The object: "/dev/pts" is on a different file system...ignoring.
### Warning: File system error.
### Filename: /proc/6953/fd/4
### No such file or directory
### Continuing...
### Warning: File system error.
### Filename: /proc/6953/fdinfo/4
### No such file or directory
### Continuing...
### Warning: File system error.
### Filename: /proc/6953/task/6953/fd/4
### No such file or directory
### Continuing...
### Warning: File system error.
### Filename: /proc/6953/task/6953/fdinfo/4
### No such file or directory
### Continuing...
Write database file: /var/lib/tripwire/ubuntu.twd
The database was successfully generated.
priyanka@ubuntu:/etc/tripwire$
```

➔ Creating a configuration file:

Editing file twpol.txt:

Now we have all the list of the files that are missing and are causing errors, we need to correct them. Perform a search for each of the files that does not exist and comment out all of the lines that match.

```
priyanka@ubuntu:/etc/tripwire$ sudo vim twpol.txt
```

```

rulename = "System boot changes",
severity = $(SIG_HI)
)
{
    /var/lock          -> $(SEC_CONFIG) ;
    /var/run           -> $(SEC_CONFIG) ; # daemon PIDs
    /var/log           -> $(SEC_CONFIG) ;
}

# These files change the behavior of the root account
(
    rulename = "Root config files",
    severity = 100
)
{
    /root              -> $(SEC_CRIT) ; # Catch all additions to /root
    #/root/mail        -> $(SEC_CONFIG) ;
    #/root/Mail        -> $(SEC_CONFIG) ;
    #/root/.xsession-errors -> $(SEC_CONFIG) ;
    #/root/.xauth      -> $(SEC_CONFIG) ;
    #/root/.tcshrc     -> $(SEC_CONFIG) ;
    #/root/.sawfish    -> $(SEC_CONFIG) ;
    #/root/.pinerc     -> $(SEC_CONFIG) ;
    #/root/.mc         -> $(SEC_CONFIG) ;
    #/root/.gnome_private -> $(SEC_CONFIG) ;
    #/root/.gnome-desktop -> $(SEC_CONFIG) ;
    #/root/.gnome      -> $(SEC_CONFIG) ;
    #/root/.esd_auth   -> $(SEC_CONFIG) ;
    #/root/.elm        -> $(SEC_CONFIG) ;
    #/root/.cshrc      -> $(SEC_CONFIG) ;
    /root/.bashrc      -> $(SEC_CONFIG) ;
    #/root/.bash_profile -> $(SEC_CONFIG) ;
    #/root/.bash_logout -> $(SEC_CONFIG) ;
    #/root/.bash_history -> $(SEC_CONFIG) ;
    #/root/.amandahosts -> $(SEC_CONFIG) ;
    #/root/.addressbook.lu -> $(SEC_CONFIG) ;
    #/root/.addressbook -> $(SEC_CONFIG) ;
    #/root/.Xresources -> $(SEC_CONFIG) ;
    #/root/.Xauthority -> $(SEC_CONFIG) -i ; # Changes Inode number on login
    #/root/.ICEauthority -> $(SEC_CONFIG) ;
}

```

```

# /root/.gnome_private          -> $(SEC_CONFIG) ;
# /root/.gnome-desktop         -> $(SEC_CONFIG) ;
# /root/.gnome                 -> $(SEC_CONFIG) ;
# /root/.esd_auth              -> $(SEC_CONFIG) ;
# /root/.elm                   -> $(SEC_CONFIG) ;
# /root/.cshrc                 -> $(SEC_CONFIG) ;
# /root/.bashrc                -> $(SEC_CONFIG) ;
# /root/.bash_profile          -> $(SEC_CONFIG) ;
# /root/.bash_logout           -> $(SEC_CONFIG) ;
# /root/.bash_history          -> $(SEC_CONFIG) ;
# /root/.amandahosts           -> $(SEC_CONFIG) ;
# /root/.addressbook.lu        -> $(SEC_CONFIG) ;
# /root/.addressbook           -> $(SEC_CONFIG) ;
# /root/.Xresources             -> $(SEC_CONFIG) ;
# /root/.Xauthority            -> $(SEC_CONFIG) -i ; # Changes Inode number on login
# /root/.ICEauthority           -> $(SEC_CONFIG) ;
}

#
# Critical devices
#
(
  rulename = "Devices & Kernel information",
  severity = $(SIG_HI),
)
{
  /dev          -> $(Device) ;
  #/proc        -> $(Device) ;
  /proc/devices -> $(Device) ;
  /proc/net     -> $(Device) ;
  /proc/tty     -> $(Device) ;
  /proc/sys     -> $(Device) ;
  /proc/cpuinfo -> $(Device) ;
  /proc/modules -> $(Device) ;
  /proc/mounts  -> $(Device) ;
  /proc/dma     -> $(Device) ;
  /proc/filesystems -> $(Device) ;
  /proc/interrupts -> $(Device) ;
  /proc/ioports -> $(Device) ;
  /proc/scsi    -> $(Device) ;
  /proc/kcore   -> $(Device) ;
  /proc/self    -> $(Device) ;
  /proc/kmsg    -> $(Device) ;
  /proc/stat    -> $(Device) ;
  /proc/loadavg -> $(Device) ;
  /proc/uptime  -> $(Device) ;
  /proc/locks   -> $(Device) ;
  /proc/meminfo -> $(Device) ;
  /proc/misc    -> $(Device) ;
}
-- INSERT --

```

```

{
  /dev          -> $(Device) ;
  /dev/pts      -> $(Device) ;
  #/proc        -> $(Device) ;
  /proc/devices -> $(Device) ;
  /proc/net     -> $(Device) ;
  /proc/tty     -> $(Device) ;
  /proc/sys     -> $(Device) ;
  /proc/cpuinfo -> $(Device) ;
  /proc/modules -> $(Device) ;
  /proc/mounts  -> $(Device) ;
}

```

```
#
# These files change every time the system boots
#
(
  rulename = "System boot changes",
  severity = $(SIG_HI)
)
{
  #/var/lock          -> $(SEC_CONFIG) ;
  #/var/run           -> $(SEC_CONFIG) ; # daemon PIDs
  /var/log            -> $(SEC_CONFIG) ;
}

# These files change the behavior of the root account
(
  rulename = "Root config files",
  severity = 100
)
```

Now that our file is configured, we need to implement it by recreating the encrypted policy file that Tripwire actually reads:

```
sudo twadmin -m P /etc/tripwire/twpol.txt
```

```
priyanka@ubuntu:/etc/tripwire$ sudo twadmin -m P /etc/tripwire/twpol.txt
Please enter your site passphrase:
Incorrect site passphrase.
Please enter your site passphrase:
Wrote policy file: /etc/tripwire/tw.pol
priyanka@ubuntu:/etc/tripwire$ █
```

After this is created, we must reinitialize the database to implement our policy:

```
sudo tripwire --init
```

```
priyanka@ubuntu:/etc/tripwire$ sudo tripwire --init
Please enter your local passphrase:
Parsing policy file: /etc/tripwire/tw.pol
Generating the database...
*** Processing Unix File System ***
Wrote database file: /var/lib/tripwire/ubuntu.twd
The database was successfully generated.
priyanka@ubuntu:/etc/tripwire$ █
```

Verifying configuration to see no errors report:

```

Open Source Tripwire(R) 2.4.2.2 Integrity Check Report

Report generated by:      root
Report created on:       Wed Mar 30 19:34:34 2016
Database last updated on: Never

=====
Report Summary:
=====

Host name:                ubuntu
Host IP address:          127.0.1.1
Host ID:                  None
Policy file used:         /etc/tripwire/tw.pol
Configuration file used:  /etc/tripwire/tw.cfg
Database file used:       /var/lib/tripwire/ubuntu.twd
Command line used:        tripwire --check

=====
Rule Summary:
=====

-----
Section: Unix File System
-----

```

Rule Name	Severity Level	Added	Removed	Modified
Other binaries	66	0	0	0
Tripwire Binaries	100	0	0	0
Other libraries	66	0	0	0
Root file-system executables	100	0	0	0
Tripwire Data Files	100	0	0	0

Rule Name	Severity Level	Added	Removed	Modified
Other binaries	66	0	0	0
Tripwire Binaries	100	0	0	0
Other libraries	66	0	0	0
Root file-system executables	100	0	0	0
Tripwire Data Files	100	0	0	0
System boot changes (/var/log)	100	0	0	0
Root file-system libraries (/lib)	100	0	0	0
Critical system boot files	100	0	0	0
Other configuration files (/etc)	66	0	0	0
Boot Scripts	100	0	0	0
Security Control	66	0	0	0
Root config files	100	0	0	0
Devices & Kernel information	100	0	0	0
Invariant Directories	66	0	0	0

Total objects scanned: 32181
Total violations found: 0

Setting up email notifications: install mailutils. We will use the mail command to mail our notifications to our email address.

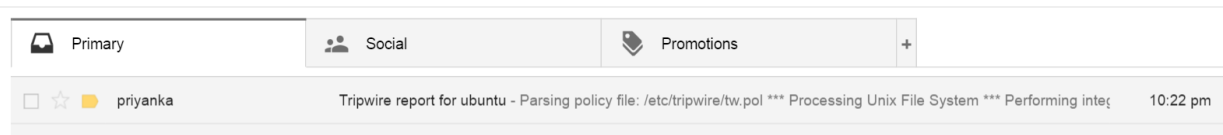
sudo apt-get install mailutils

Configure the e-mail ID for receiving reports:

sudo tripwire --check | mail -s "Tripwire report for `uname -n`" priyanka97562@gmail.com

```
priyanka@ubuntu:/etc/tripwire$ sudo tripwire --check | mail -s "Tripwire report for `uname -n`" priyanka97562@gmail.com
```

Once the configuration is complete, we will receive an email from Ubuntu about the current state of the system.



Report:

Tripwire report for ubuntu

Inbox x



priyanka <priyanka@ubuntu>

to me

Parsing policy file: /etc/tripwire/tw.pol
 *** Processing Unix File System ***
 Performing integrity check...
 Wrote report file: /var/lib/tripwire/report/ubuntu-20160330-202144.twr

Open Source Tripwire(R) 2.4.2.2 Integrity Check Report

Report generated by: root
 Report created on: Wed Mar 30 20:21:44 2016
 Database last updated on: Never

=====
 Report Summary:
 =====

Host name: ubuntu
 Host IP address: 127.0.1.1
 Host ID: None
 Policy file used: /etc/tripwire/tw.pol
 Configuration file used: /etc/tripwire/tw.cfg
 Database file used: /var/lib/tripwire/ubuntu.twd
 Command line used: tripwire --check

=====
 P... C


```
=====
Rule Summary:
=====
```

```
-----
Section: Unix File System
-----
```

Rule Name	Severity Level	Added	Removed	Modified
* Other binaries	66	18	0	2
Tripwire Binaries	100	0	0	0
* Other libraries	66	47	0	2
Root file-system executables	100	0	0	0
Tripwire Data Files	100	0	0	0
System boot changes (/var/log)	100	0	0	0
Root file-system libraries (/lib)	100	0	0	0
Critical system boot files	100	0	0	0
* Other configuration files (/etc)	66	18	0	5
Boot Scripts	100	0	0	0
Security Control	66	0	0	0
Root config files	100	0	0	0
Devices & Kernel information	100	0	0	0
Invariant Directories	66	0	0	0

Total objects scanned: 32264

Total violations found: 92

```
=====
```

➔ The software changes we made by performing an interactive check to update the database. This was completed by running the following command:

```
sudo tripwire --check --interactive
```

This will run the same tests as normal. However, the report is copied into a text file and opened with the default editor instead of outputting the report to the screen.

```
priyanka@ubuntu:/etc/tripwire$ sudo tripwire --check --interactive
```

```

Open Source Tripwire(R) 2.4.2.2 Integrity Check Report

Report generated by:      root
Report created on:       Wed Mar 30 20:44:12 2016
Database last updated on: Never

=====
Report Summary:
=====

Host name:                ubuntu
Host IP address:          127.0.1.1
Host ID:                  None
Policy file used:         /etc/tripwire/tw.pol
Configuration file used:  /etc/tripwire/tw.cfg
Database file used:       /var/lib/tripwire/ubuntu.twd
Command line used:        tripwire --check --interactive

=====
Rule Summary:
=====

-----
Section: Unix File System
-----

Rule Name                Severity Level   Added   Removed   Modified
-----
* Other binaries          66               18      0          2
  Tripwire Binaries      100              0      0          0
* Other libraries         66               47      0          2
  Root file-system executables 100              0      0          0
  Tripwire Data Files     100              0      0          0
  System boot changes     100              0      0          0
  (/var/log)
  Root file-system libraries 100              0      0          0
  (/lib)
  Critical system boot files 100              0      0          0
* Other configuration files 66               18      0          5
  (/etc)
  Boot Scripts            100              0      0          0
  Security Control        66               0      0          0
  Root config files       100              0      0          0
  Devices & Kernel information 100              0      0          0
  Invariant Directories   66               0      0          0

```

Read 1913 lines

Added:

```

Added:
[x] "/usr/lib/x86_64-linux-gnu/libmysqlclient.so.18"
[x] "/usr/lib/x86_64-linux-gnu/libntlm.so.0.0.19"
[x] "/usr/lib/x86_64-linux-gnu/libmysqlclient_r.so.18"
[x] "/usr/lib/x86_64-linux-gnu/libkyotocabinet.so.16.13.0"
[x] "/usr/lib/x86_64-linux-gnu/libmysqlclient_r.so.18.0.0"
[x] "/usr/lib/x86_64-linux-gnu/libkyotocabinet.so.16"
[x] "/usr/lib/x86_64-linux-gnu/libntlm.so.0"
[x] "/usr/lib/x86_64-linux-gnu/libmysqlclient.so.18.0.0"
[x] "/usr/lib/libmu_mbox.so.4.0.0"
[x] "/usr/lib/libgsasl.so.7"
[x] "/usr/lib/libmu_ldap.so.4.0.0"
[x] "/usr/lib/libmu_compat.so.0.0.0"
[x] "/usr/lib/libmu_dbm.so.4"
[x] "/usr/lib/libmu_auth.so.4"
[x] "/usr/lib/libmu_py.so.4.0.0"
[x] "/usr/lib/mailutils"
[x] "/usr/lib/mailutils/timestamp.so"
[x] "/usr/lib/mailutils/pipe.so"

```

```

Added:
"/etc/mysql"
"/etc/mysql/conf.d"
"/etc/mysql/conf.d/.keepme"
"/etc/mysql/my.cnf"
"/etc/alternatives/mailx"
"/etc/alternatives/mail"
"/etc/alternatives/readmsg"
"/etc/alternatives/messages"
"/etc/alternatives/frm"
"/etc/alternatives/movemail"
"/etc/alternatives/dotlock"
"/etc/alternatives/frm.1.gz"
"/etc/alternatives/mailx.1.gz"
"/etc/alternatives/messages.1.gz"
"/etc/alternatives/movemail.1.gz"
"/etc/alternatives/readmsg.1.gz"
"/etc/alternatives/mail.1.gz"
"/etc/alternatives/dotlock.1.gz"

Modified:
"/etc"
"/etc/alternatives"
"/etc/alternatives/from"
"/etc/alternatives/from.1.gz"
"/etc/ld.so.cache"

=====
Error Report:
=====

No Errors

-----
*** End of report ***

Open Source Tripwire 2.4 Portions copyright 2000 Tripwire, Inc. Tripwire is a registered
trademark of Tripwire, Inc. This software comes with ABSOLUTELY NO WARRANTY;
for details use --version. This is free software which may be redistributed
or modified only under certain conditions; see COPYING for details.
All rights reserved.
Integrity check complete.
Please enter your local passphrase:

```

```

or modified only under certain conditions; see COPYING for details.
All rights reserved.
Integrity check complete.
Please enter your local passphrase:
Wrote database file: /var/lib/tripwire/ubuntu.twd
priyanka@ubuntu:/etc/tripwire$ █

```

Tripwire has been run 8 times for different scenarios to test its running.

Test Runs:

Scenario 1: Installation of tcpdump

apt-get install tcpdump

```
priyanka@ubuntu:/etc/tripwire$ sudo apt-get install tcpdump
```

```
priyanka@ubuntu:/etc/tripwire$ sudo apt-get install tcpdump
[sudo] password for priyanka:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  tcpdump
1 upgraded, 0 newly installed, 0 to remove and 579 not upgraded.
Need to get 355 kB of archives.
After this operation, 3,072 B disk space will be freed.
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main tcpdump amd64 4.5.1-2ubuntu1.2 [355 kB]
Fetched 355 kB in 1s (346 kB/s)
(Reading database ... 165899 files and directories currently installed.)
Preparing to unpack .../tcpdump_4.5.1-2ubuntu1.2_amd64.deb ...
Unpacking tcpdump (4.5.1-2ubuntu1.2) over (4.5.1-2ubuntu1) ...
Processing triggers for man-db (2.6.7.1-1) ...
Setting up tcpdump (4.5.1-2ubuntu1.2) ...
priyanka@ubuntu:/etc/tripwire$ █
```

Report:

```
priyanka@ubuntu:/etc/tripwire$ sudo tripwire --check --interactive █
```

Rule Name	Severity Level	Added	Removed	Modified
Other binaries	66	0	0	2
Tripwire Binaries	100	0	0	0
Other libraries	66	0	0	0
Root file-system executables	100	0	0	0
Tripwire Data Files	100	0	0	0
System boot changes (/var/log)	100	0	0	0
Root file-system libraries (/lib)	100	0	0	0
Critical system boot files	100	0	0	0
Other configuration files (/etc)	66	0	0	3
Boot Scripts	100	0	0	0
Security Control	66	0	0	0
Root config files	100	0	0	0
Devices & Kernel information	100	0	0	0
Invariant Directories	66	0	0	0

```

Modified object name: /etc/apparmor.d/cache/usr.sbin.tcpdump

Property:           Expected           Observed
-----
Object Type         Regular File         Regular File
Device Number       2049                 2049
* Inode Number       656929               658117
Mode                -rw-----          -rw-----
■ Num Links          1                    1
UID                 root (0)              root (0)
GID                 root (0)              root (0)
Size                41713                 41713
* Modify Time        Wed Mar 30 16:21:36 2016 Wed Mar 30 21:25:06 2016
Blocks              88                    88
CRC32                DQ4STc                DQ4STc
MD5                  AAPToM6BAksHE0ChGjqz7i AAPToM6BAksHE0ChGjqz7i
=====
Error Report:
=====

No Errors

-----
*** End of report ***

```

Scenario 2: File addition in /var/log > test.log

sudo vim test.log

Rule Name	Severity Level	Added	Removed	Modified
Other binaries	66	0	0	0
Tripwire Binaries	100	0	0	0
Other libraries	66	0	0	0
Root file-system executables	100	0	0	0
Tripwire Data Files	100	0	0	0
System boot changes (/var/log)	100	1	0	0
Root file-system libraries (/lib)	100	0	0	0
Critical system boot files	100	0	0	0
Other configuration files (/etc)	66	0	0	0
Boot Scripts	100	0	0	0
Security Control	66	0	0	0
Root config files	100	0	0	0
Devices & Kernel information	100	0	0	0
Invariant Directories	66	0	0	0

```
-----
Added Objects: 1
-----
```

```
Added object name: /var/log/test.log
```

Property:	Expected	Observed
* Object Type	---	Regular File
* Device Number	---	2049
* Inode Number	---	146862
* Mode	---	-rw-r--r--
* Num Links	---	1
* UID	---	root (0)
* GID	---	root (0)

```
Integrity check complete.
Please enter your local passphrase:
Wrote database file: /var/lib/tripwire/ubuntu.twd
```

Scenario 3: Modifying test.log in /var/log.

sudo vim test.log

```
priyanka@ubuntu:/var/log$ sudo vim test.log
```

```
test tes
```

```
hello
```

Report:

```
=====
Rule Summary:
=====
-----
Section: Unix File System
-----
Rule Name                Severity Level  Added  Removed  Modified
-----
Other binaries           66             0      0         0
Tripwire Binaries       100            0      0         0
Other libraries          66             0      0         0
Root file-system executables 100            0      0         0
Tripwire Data Files     100            0      0         0
* System boot changes   100            0      0         1
(/var/log)
Root file-system libraries 100            0      0         0
(/lib)
Critical system boot files 100            0      0         0
Other configuration files 66             0      0         0
(/etc)
Boot Scripts             100            0      0         0
Security Control         66             0      0         0
Root config files        100            0      0         0
Devices & Kernel information 100            0      0         0
Invariant Directories    66             0      0         0

^G Get Help          ^O WriteOut
^X Exit              ^J Justify         ^R Read File
^W Where Is
```

```

-----
Modified Objects: 1
-----
Modified object name: /var/log/test.log

Property:      Expected      Observed
-----
Object Type    Regular File    Regular File
Device Number  2049            2049
* Inode Number 146862         146865
Mode           -rw-r--r--     -rw-r--r--
■ Num Links    1               1
UID            root (0)        root (0)
GID            root (0)        root (0)

=====
Error Report:
=====

No Errors

-----
*** End of report ***

```

Scenario 4: Deleting the file test.log

sudo rm -rf test.log

```

-----
Section: Unix File System
-----

```

Rule Name	Severity Level	Added	Removed	Modified
Other binaries	66	0	0	0
Tripwire Binaries	100	0	0	0
Other libraries	66	0	0	0
Root file-system executables	100	0	0	0
Tripwire Data Files	100	0	0	0
* System boot changes (/var/log)	100	0	1	0

```
priyanka@ubuntu:/var/log$ sudo rm -rf test.log
```

Scenario 5: Changing the file location of btmp file to hp/btmp

```
sudo mv btmp hp/btmp
```

```
priyanka@ubuntu:/var/log$ sudo mv btmp hp/btmp
```

```
=====  
Object Summary:  
=====
```

```
-----  
# Section: Unix File System  
-----
```

```
-----  
Rule Name: System boot changes (/var/log)  
Severity Level: 100  
-----
```

Remove the "x" from the adjacent box to prevent updating the database with the new values for this object.

Removed:

```
[x] "/var/log/test.log"
```

```
-----  
Rule Name                Severity Level   Added   Removed  Modified  
-----  
Other binaries           66              0       0         0  
Tripwire Binaries        100             0       0         0  
Other libraries          66              0       0         0  
Root file-system executables 100             0       0         0  
Tripwire Data Files      100             0       0         0  
* System boot changes    100             1       1         0  
  (/var/log)  
Root file-system libraries 100             0       0         0  
  (/lib)
```



```
Added object name: /var/log/hp/btmp
```

Property:	Expected	Observed
* Object Type	---	Regular File
* Device Number	---	2049
* Inode Number	---	145156
* Mode	---	-rw-rw----
* Num Links	---	1
* UID	---	root (0)
* GID	---	utmp (43)

```
-----  
Removed Objects: 1  
-----
```

```
Removed object name: /var/log/btmp
```

Property:	Expected	Observed
* Object Type	Regular File	---
* Device Number	2049	---
* Inode Number	145156	---
* Mode	-rw-rw----	---
* Num Links	1	---

Scenario 6: Modifying the authorizations

```
priyanka@ubuntu:/var/log/hp$ sudo chmod 777 btmp
```

```
-----  
Section: Unix File System  
-----
```

Rule Name	Severity Level	Added	Removed	Modified
Other binaries	66	0	0	0
Tripwire Binaries	100	0	0	0
Other libraries	66	0	0	0
Root file-system executables	100	0	0	0
Tripwire Data Files	100	0	0	0
* System boot changes (/var/log)	100	0	0	1
Root file-system libraries (/lib)	100	0	0	0
Critical system boot files	100	0	0	0
Other configuration files	66	0	0	0

```
-----
Modified Objects: 1
-----
```

```
Modified object name: /var/log/hp/btmp
```

Scenario 7: Creating a soft link

```
priyanka@ubuntu:/var/log$ sudo vim test.log
priyanka@ubuntu:/var/log$ cd ~
priyanka@ubuntu:~$ ln -s /var/log/test.log test
priyanka@ubuntu:~$ ls -l
total 44
drwxr-xr-x 2 priyanka priyanka 4096 Mar 30 16:27 Desktop
drwxr-xr-x 2 priyanka priyanka 4096 Mar 30 16:27 Documents
drwxr-xr-x 2 priyanka priyanka 4096 Mar 30 16:27 Downloads
-rw-r--r-- 1 priyanka priyanka 8980 Mar 30 16:16 examples.desktop
drwxr-xr-x 2 priyanka priyanka 4096 Mar 30 16:27 Music
drwxr-xr-x 2 priyanka priyanka 4096 Mar 30 16:27 Pictures
drwxr-xr-x 2 priyanka priyanka 4096 Mar 30 16:27 Public
drwxr-xr-x 2 priyanka priyanka 4096 Mar 30 16:27 Templates
lrwxrwxrwx 1 priyanka priyanka  17 Mar 30 22:08 test -> /var/log/test.log
drwxr-xr-x 2 priyanka priyanka 4096 Mar 30 16:27 Videos
```

Rule Summary:

```
=====
```

```
Section: Unix File System
```

Rule Name	Severity Level	Added	Removed	Modified
Other binaries	66	0	0	0
Tripwire Binaries	100	0	0	0
Other libraries	66	0	0	0
Root file-system executables	100	0	0	0
Tripwire Data Files	100	0	0	0
* System boot changes (/var/log)	100	1	0	0
Root file-system libraries (/lib)	100	0	0	0
Critical system boot files	100	0	0	0
Other configuration files (/etc)	66	0	0	0
Boot Scripts	100	0	0	0
Security Control	66	0	0	0
Root config files	100	0	0	0
Devices & Kernel information	100	0	0	0
Invariant Directories	66	0	0	0

Scenario 8: Installing ssh**sudo apt-get install ssh openssh-server**

```

priyanka@ubuntu:/var/lib/tripwire/report$ sudo apt-get install ssh openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libck-connector0 ncurses-term openssh-client openssh-sftp-server
  python-requests python-urllib3 ssh-import-id
Suggested packages:
  .....
```

```

-----
Rule Summary:
=====
```

```

-----
Section: Unix File System
-----
```

Rule Name	Severity Level	Added	Removed	Modified
* Other binaries	66	4	0	12
Tripwire Binaries	100	0	0	0
* Other libraries	66	78	0	7
Root file-system executables	100	0	0	0
Tripwire Data Files	100	0	0	0
System boot changes (/var/log)	100	0	0	0
* Root file-system libraries (/lib)	100	3	0	1
Critical system boot files	100	0	0	0
* Other configuration files (/etc)	66	17	0	12
* Boot Scripts	100	1	0	1
* Security Control	66	0	0	2

To read a Tripwire report:

The reports generated for the above 8 scenarios are reported in /var/lib/tripwire/report

```
/usr/sbin/twprint -m r --twrfile /var/lib/tripwire/report/<name>.twr
```

The reports here are encrypted.

Advantages:

- 1) Tripwire reduces the administration.

- 2) Open source.
- 3) Can be configured as per the policies.
- 4) Generates reports that are easy to understand.
- 5) Enables the admin to interactively update the database.
- 6) Monitors file accesses, changes to file systems, etc.
- 7) Reduce time between attack recognition and response.

Disadvantages:

- 1) Higher learning curve to install and maintain the software.
- 2) For the Unix version of Tripwire, there should be a familiarity with a command line and vi text editor.

BRO – IDS: BRO – IDS is a network-based intrusion detection system. It is one of the best known open source network-based IDSs available. Its deep packet inspection makes it stand out as a network attack detector. Bro IDS architecture is straight forward. It monitors the network and analyses them with the policy interpreter. If it deviates from normal, it will create an alert or can be configured to respond.

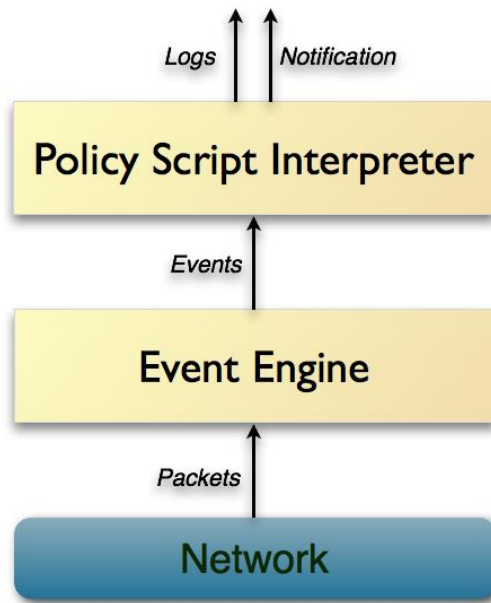


Figure 3.2. BRO – IDS Architecture

BRO – IDS captures logs for the network traffic and makes it easier to read the logs in different conditions. An infected pcap file was analyzed with BRO – IDS in this paper to spoof a bad network traffic. Below are the logs and their results.

```
priyanka@ubuntu:~/pcap$ bro -r infected.pcap
```

```

priyanka@ubuntu:~/pcap$ bro -r infected.pcap
priyanka@ubuntu:~/pcap$ ll
total 248
drwxrwxrwx  4 root    root    4096 Apr 12 13:51 ./
drwxrwxrwx 17 priyanka priyanka 4096 Apr 17 19:07 ../
-rw-rw-r--  1 priyanka priyanka 2512 Apr 18 13:37 conn.log
-rw-rw-r--  1 priyanka priyanka 1935 Apr 18 13:37 dns.log
-rw-rw-r--  1 priyanka priyanka  141 Apr 12 13:50 extract-all.bro
drwxrwxr-x  2 priyanka priyanka 4096 Apr 12 13:51 extract_files/
-rw-rw-r--  1 priyanka priyanka 1977 Apr 18 13:37 files.log
-rw-rw-r--  1 priyanka priyanka 2162 Apr 18 13:37 http.log
-rw-rw-r--  1 priyanka priyanka 193410 Apr 12 12:59 infected.pcap
-rw-rw-r--  1 priyanka priyanka  253 Apr 18 13:37 packet_filter.log
-rw-rw-r--  1 priyanka priyanka  558 Apr 18 13:37 pe.log
-rw-rw-r--  1 priyanka priyanka  864 Apr 18 13:37 ssl.log
drwx----- 3 priyanka priyanka 4096 Apr 18 13:37 .state/
-rw-rw-r--  1 priyanka priyanka  330 Apr 18 13:37 weird.log
-rw-rw-r--  1 priyanka priyanka 1529 Apr 18 13:37 x509.log
priyanka@ubuntu:~/pcap$

```

These are the logs which generate according to the network traffic, and each log file has its own significance.

- 1) Conn.log contains TCP/UDP/ICMP connections.
- 2) Dns.log contains DNS activity.
- 3) http.log contains HTTP requests and replies.
- 4) ssl.log contains SSL/TLS handshake info.
- 5) weirds.log contains Unexpected network-level activity.

We can generate different outputs according to the queries we have:

```
cat conn.log | bro-cut id.orig_h id.orig_p id.resp_h uid duration
```

```
priyanka@ubuntu:~/pcap$ cat conn.log | bro-cut id.orig.h id.orig_p id.resp_h uid duration
64291 192.168.23.2 Cm9FYN3oD4C956nee7 2.930238
1062 65.55.195.250 CdtQ37CmoE1acVe7 0.593467
1061 59.53.91.102 CvbwI53qWt5u71Nkp4 14.323807
59820 192.168.23.2 CTL19yEnHqINyH1vb 0.070762
1063 59.53.91.102 CS8ez53hxCs3PSKeuj 8.803635
1064 59.53.91.102 CqzWvd2Epw1f9IVhsj 10.189838
1065 59.53.91.102 CE3GyyB9vu6GimYsb 14.041870
64292 192.168.23.2 C4ZAcR1gX59Nt4DbRl 0.844963
137 192.168.23.2 CFdXWd2tRFSpBf0Abd 2.998871
137 59.53.91.102 Cln502esmEvJFgDdd 3.002087
1068 213.155.29.144 CXK2aj3uuEjpb3XQ34 0.518227
1067 59.53.91.102 Cq3so2nWs3o7rted6 13.928659
52499 192.168.23.2 CK6HpS3k5Ytdzdx6 0.099538
1069 212.252.32.20 C2vjjj3fwp4QX90e34 1.516784
1066 59.53.91.102 CfvzLQ1VPBPrqsw8TL 19.946688
```

→ Find all connections that are last longest:

```
cat conn.log | bro-cut duration id.orig.h id.orig_p id.resp_h uid duration
```

```
priyanka@ubuntu:~/pcap$ cat conn.log | bro-cut duration id.orig.h id.orig_p id.resp_h uid duration
2.930238 64291 192.168.23.2 Cm9FYN3oD4C956nee7 2.930238
0.593467 1062 65.55.195.250 CdtQ37CmoE1acVe7 0.593467
14.323807 1061 59.53.91.102 CvbwI53qWt5u71Nkp4 14.323807
0.070762 59820 192.168.23.2 CTL19yEnHqINyH1vb 0.070762
8.803635 1063 59.53.91.102 CS8ez53hxCs3PSKeuj 8.803635
10.189838 1064 59.53.91.102 CqzWvd2Epw1f9IVhsj 10.189838
14.041870 1065 59.53.91.102 CE3GyyB9vu6GimYsb 14.041870
0.844963 64292 192.168.23.2 C4ZAcR1gX59Nt4DbRl 0.844963
2.998871 137 192.168.23.2 CFdXWd2tRFSpBf0Abd 2.998871
3.002087 137 59.53.91.102 Cln502esmEvJFgDdd 3.002087
0.518227 1068 213.155.29.144 CXK2aj3uuEjpb3XQ34 0.518227
13.928659 1067 59.53.91.102 Cq3so2nWs3o7rted6 13.928659
0.099538 52499 192.168.23.2 CK6HpS3k5Ytdzdx6 0.099538
1.516784 1069 212.252.32.20 C2vjjj3fwp4QX90e34 1.516784
19.946688 1066 59.53.91.102 CfvzLQ1VPBPrqsw8TL 19.946688
```

→ List the connections by in increasing order of duration, i.e., the longest connections at the end.

```
awk 'NR > 4' < conn.log | sort -t$'\t' -k 9 -n
```

```
priyanka@ubuntu:~/pcap$ awk 'NR > 4' < conn.log | sort -t$'\t' -k 9 -n
#close 2016-04-18-13-39-33
#fields ts uid id.orig_h id.orig_p id.resp_h id.resp_p proto service duration orig_bytes resp_bytes conn_state local_orig local_resp misse
#open 2016-04-18-13-39-33
#path conn
#types time string addr port addr port enum string interval count count string bool bool count string count count count set[string]
1268758234.688275 CTL19yEnHqINyH1vb 192.168.23.129 59820 192.168.23.2 53 udp dns 0.070762 26 120 SF - - 0 Dd 1 54 1 1
48 (empty)
1268758204.998619 CK6HpS3k5Ytdzdx6 192.168.23.129 52499 192.168.23.2 53 udp dns 0.099538 29 193 SF - - 0 Dd 1 57 1 2
21 (empty)
1268758261.726678 CXK2aj3uuEjpb3XQ34 192.168.23.129 1068 213.155.29.144 444 tcp - 0.518227 263 0 SF - - 0 ShAdFaf 5 471 4 1
64 (empty)
1268758221.420447 CdtQ37CmoE1acVe7 192.168.23.129 1062 65.55.195.250 443 tcp ssl 0.593467 1318 5387 SF - - 0 ShAdadff 8 16461
1 5831 (empty)
1268758244.609168 C4ZAcR1gX59Nt4DbRl 192.168.23.129 64292 192.168.23.2 53 udp dns 0.844963 43 101 SF - - 0 Dd 1 71 1 1
29 (empty)
1268758205.114322 C2vjjj3fwp4QX90e34 192.168.23.129 1069 212.252.32.20 80 tcp http 1.516784 251 886 S3 - - 0 ShAdadff 5 459 5 1
976 (empty)
1268758214.788023 Cm9FYN3oD4C956nee7 192.168.23.129 64291 192.168.23.2 53 udp dns 2.930238 78 360 SF - - 0 Dd 3 162 3 4
44 (empty)
1268758239.366023 CFdXWd2tRFSpBf0Abd 192.168.23.129 137 192.168.23.2 137 udp dns 2.998871 204 0 SF - - 0 D 3 288 0 0
1268758245.456726 Cln502esmEvJFgDdd 192.168.23.129 137 59.53.91.102 137 udp dns 3.002087 150 0 SF - - 0 D 3 234 0 0
1268758234.546057 CS8ez53hxCs3PSKeuj 192.168.23.129 1063 59.53.91.102 80 tcp http 8.803635 255 577 RSTO - - 0 ShAdadff 6 503 5
1358 (empty)
1268758234.841573 CqzWvd2Epw1f9IVhsj 192.168.23.129 1064 59.53.91.102 80 tcp - 10.189838 263 7336 SF - - 0 SahAdff 10 679 1
4 8584 (empty)
1268758249.968107 Cq3so2nWs3o7rted6 192.168.23.129 1067 59.53.91.102 80 tcp - 13.928659 199 68370 SF - - 0 SahAdff 30 14156
3 72354 (empty)
1268758234.842707 CE3GyyB9vu6GimYsb 192.168.23.129 1065 59.53.91.102 80 tcp - 14.041870 260 5830 SF - - 0 SahAdff 7 556 8
6154 (empty)
1268758217.701672 CvbwI53qWt5u71Nkp4 192.168.23.129 1061 59.53.91.102 80 tcp http 14.323807 773 1966 RSTO - - 0 ShAdadff 8 11018
2714 (empty)
1268758248.981429 CfvzLQ1VPBPrqsw8TL 192.168.23.129 1066 59.53.91.102 80 tcp http 19.946688 211 68370 SF - - 0 ShAdadff 27 12996
1 74566 (empty)
```

→ Find all connections that last longer than one minute.

awk 'NR > 4 && \$9 > 60' conn.log

```
priyanka@ubuntu:~/pcap$ awk 'NR > 4 && $9 > 60' conn.log
#fields ts uid id.orig_h id.orig_p id.resp_h id.resp_p proto service duration orig_bytes resp_bytes conn_state local_orig local_resp mtss
#d_bytes history orig_pkts orig_ip_bytes resp_pkts resp_ip_bytes tunnel_parents
#types time string addr port addr port enum string interval count count string bool bool count string count count count count set[string]
```

→ Show a breakdown of the number of connections, sorted by service.

bro-cut service < conn.log | sort | uniq -c | sort -n

```
priyanka@ubuntu:~/pcap$ bro-cut service < conn.log | sort | uniq -c | sort -n
 1 ssl
 4 -
 4 http
 6 dns
```

→ Show the top 10 destination ports in descending order.

bro-cut id.resp_p < conn.log | sort | uniq -c | sort -rn | head -n 10

```
priyanka@ubuntu:~/pcap$ bro-cut id.resp_p < conn.log | sort | uniq -c | sort -rn | head -n 10
 7 80
 4 53
 2 137
 1 444
 1 443
```

→ What are the distinct browsers in this trace? What are the distinct MIME types of the downloaded URLs?

bro-cut user_agent < http.log | sort -u

```
priyanka@ubuntu:~/pcap$ bro-cut user_agent < http.log | sort -u
Microsoft Internet Explorer
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Mozilla/4.0 (Windows XP 5.1) Java/1.6.0_05
```

→ What are the three most commonly accessed web sites?

bro-cut host < http.log | sort | uniq -c | sort -n | tail -n 3

```
priyanka@ubuntu:~/pcap$ bro-cut host < http.log | sort | uniq -c | sort -n | tail -n 3
 1 freeways.in
 4 nrtjo.eu
```

We can use various commands to find different methods by which we can use this IDS effectively. This can be integrated with security onion, as well to provide a GUI to analyze the

logs for novice users rather than providing the Linux interface for analyzing.

Advantages:

- 1) Open source.
- 2) Network-based intrusion detection system.
- 3) Bro policy engine can do very powerful tasks.
- 4) Automation can be done in Bro IDS.

Disadvantages:

- 1) Complicated to set up

TCP Wrappers: TCP Wrapper is a host-based IDS used to filter network access to Internet Protocol servers on Unix-like operating systems. As per Redhat Documentation, “TCP Wrappers add an additional layer of protection by defining which hosts are or are not allowed to connect to "wrapped" network services. One such wrapped network service is the xinetd super server. This service is called a super server because it controls connections to a subset of network services and further refines access control” [15]. The TCP Wrapper architecture is shown in the figure below.

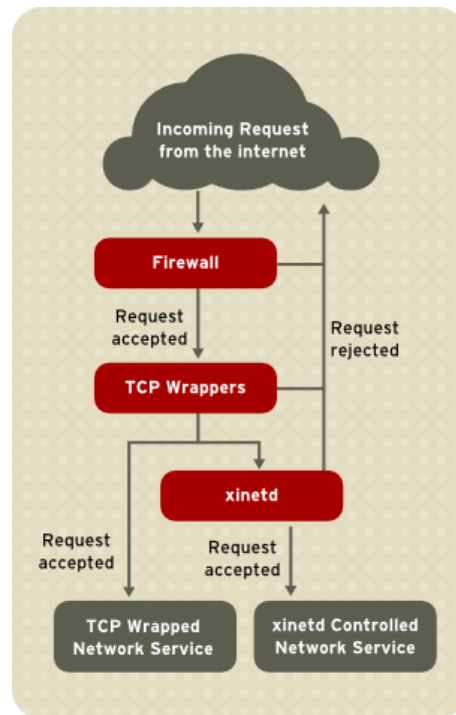


Figure 3.3. TCP Wrapper Architecture

Configuration Files for TCP Wrappers:

1. /etc/hosts.allow
2. /etc/hosts.deny

Each rule uses the following basic format to control access to network services:

<Daemon list> : <client list> : <option>

Daemon: we list different daemon services we connect to through a selected server such as VSFTPD, SSHD, ALL, etc.

Client list: we list the clients who should be given access to or be denied.

Options: we have different options to allow access control such as

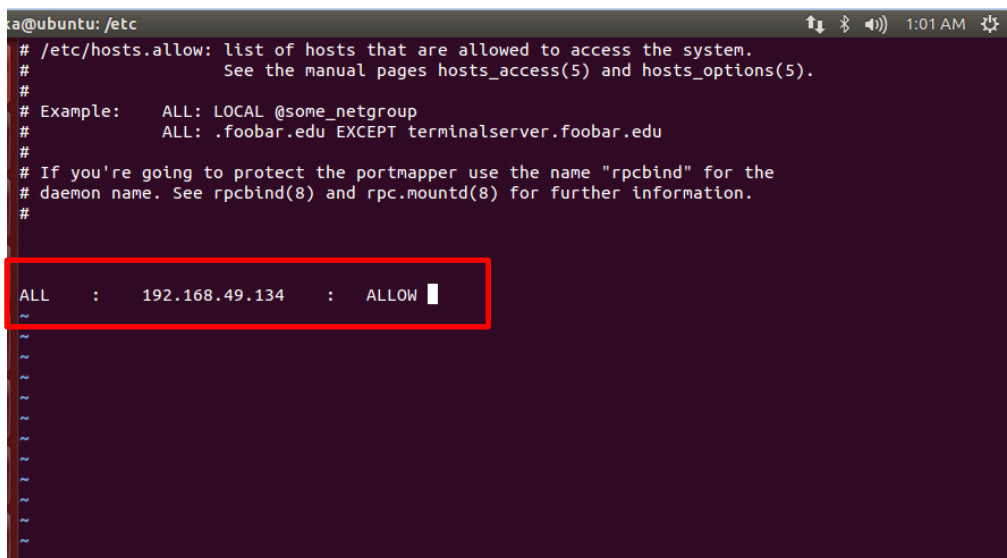
- 1) ALLOW
- 2) DENY
- 3) Spawn

- 4) Twist
- 5) Banner

Implementation of one after one option is shown below.

1) Allow:

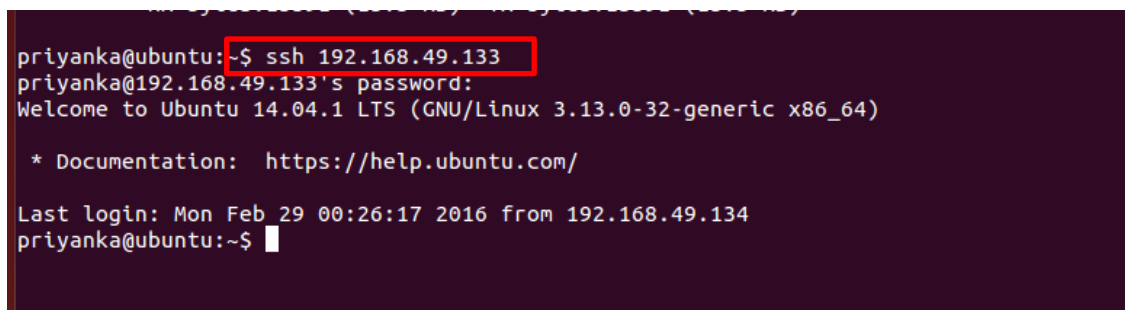
To configure a particular host or list of hosts in a server, the following lines must be added to the hosts.allow file in /etc to allow them to access the server:



```
priyanka@ubuntu: /etc
# /etc/hosts.allow: list of hosts that are allowed to access the system.
# See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:  ALL: LOCAL @some_netgroup
#          ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#
ALL : 192.168.49.134 : ALLOW
```

The above line here displays all services to the client's host 192.168.49.134, which must be allowed to connect to the server where these lines are placed.

Test:



```
priyanka@ubuntu: ~$ ssh 192.168.49.133
priyanka@192.168.49.133's password:
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-32-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Last login: Mon Feb 29 00:26:17 2016 from 192.168.49.134
priyanka@ubuntu: ~$
```

The above option allowed is tested by trying to connect to the server from client using ssh. The test proved to be right.

2) Deny:

To configure a particular host or list of hosts in a server, following lines have to be added to hosts.allow file in /etc to deny them to access the server:

```
## /etc/hosts.allow: list of hosts that are allowed to access the system.
# See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:  ALL: LOCAL @some_netgroup
#          ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#
ALL      :      192.168.49.134      : DENY
~
~
~
~
~
~
~
~
~
~
```

The above line here means all services to the client's host 192.168.49.134 must **NOT** be allowed to connect to the server where these lines are placed.

Test:

```
priyanka@ubuntu:~$ ssh 192.168.49.133
ssh_exchange_identification: read: Connection reset by peer
priyanka@ubuntu:~$
```

As we have denied the access for the client to server, the ssh doesn't allow it to connect throwing an error message as shown above.

3) Spawn:

To configure a particular host or list of hosts in a server, the following lines must be added to hosts.allow file in /etc to deny or allow access to the server.

Uses: 1) Spawn is used to print messages on the terminal of the server or to create a folder in server, which lets the admin of the server know who is accessing the server, at what time, from where, and so on.

```
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#
ALL      :      192.168.49.134      : spawn /bin/echo "%s%d%c" >> /Login
```

The above line here means all services to the client's host 192.168.49.134 must be allowed, as they are written in the hosts.allow DENY option exclusively to connect to the server where these lines are placed. When a client tries to access the server, a folder /Login is created in root, which will have info about:

%s → supplies various types of server information, such as the daemon process and the host or IP address of the server.

%d → Supplies the daemon process name.

%c → Supplies a variety of client information, such as the username and hostname, or the username and IP address.

There are various other parameters that can be used to print details according to requirements.

This following line can be added to print a message “Alert” on the terminal:

```
ALL : 192.168.49.134 : spawn /bin/echo "Alert!!" > /dev/tty1
```

Test:

```
priyanka@ubuntu:~$ ftp 192.168.49.135
Connected to 192.168.49.135.
220 (vsFTPd 3.0.2)
Name (192.168.49.135:priyanka): priyanka
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

As the service is used by client to connect to server a folder Login is created in root directory by itself.

```
priyanka@ubuntu:~/honey$ cd ..
priyanka@ubuntu:/$ ls
bin  cdrom  etc  initrd.img  lib64  lost+found  mnt  proc  run  srv  tmp  var
boot  dev  home  lib  Login  media  opt  root  sbin  sys  usr  vmlinuz
```

Looking at the details of the file, here is the information below: as FTP is performed on the server, we have information related to that in the file

```
priyanka@ubuntu:/$ cat Login
sshd@192.168.49.135sshd192.168.49.134
sshd@192.168.49.135sshd192.168.49.134
sshd@192.168.49.135sshd192.168.49.134
vsftpd@192.168.49.135vsftpd192.168.49.134
priyanka@ubuntu:/$
```

4) Twist:

To configure a particular host or list of hosts in a server, following lines have to be added to hosts.allow file in /etc to deny access to the server with a meaningful message.

Uses: 1) **Twist** is used to print message on terminal of client while trying to access the server which is denied by the usage of twist by default

```
# /etc/hosts.allow: list of hosts that are allowed to access the system.
# See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:  ALL: LOCAL @some_netgroup
#          ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#
ALL      : 192.168.49.134      : twist  /bin/echo "YOU ARE NOT AUTHORIZED TO CONNECT TO THIS
SERVER"
```

Test:

```
priyanka@ubuntu: ~
priyanka@ubuntu:~$ ftp 192.168.49.135
Connected to 192.168.49.135.
YOU ARE NOT AUTHORIZED TO CONNECT TO THIS SERVER
ftp> █
```

Here the access to the client is denied with a meaningful message as mentioned in hosts.allow file, i.e. “You are not authorized to connect to this server”.

5) Banners:

To configure a particular host or list of hosts in a server, following lines have to be added to hosts.allow file in /etc to allow access to the server with a meaningful message.

Uses: 1) **Banner** is used to print message on terminal of client while trying to access the server by using a particular daemon service. There are different steps to implement banners.

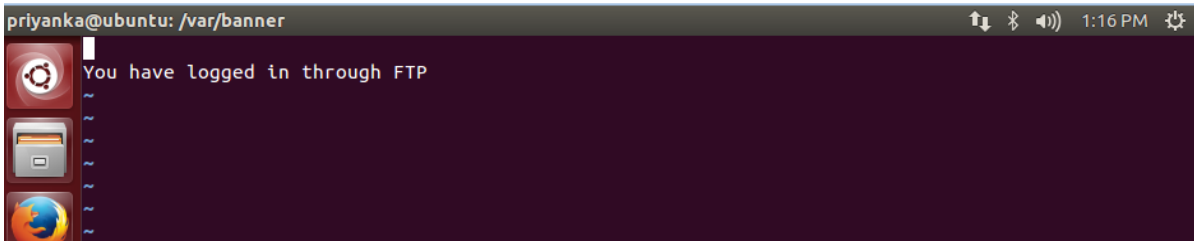
Step 1: creating a folder banner in /var directory.

```
priyanka@ubuntu:/var$ ll
total 56
drwxr-xr-x 14 root root  4096 Feb 29 01:49 ./
drwxr-xr-x 23 root root  4096 Feb 29 13:01 ../
drwxr-xr-x  2 root root  4096 Feb 29 01:39 backups/
drwxr-xr-x  2 root root  4096 Feb 29 01:51 banner/
drwxr-xr-x 17 root root  4096 Jul 22 2014 cache/
drwxrwsrwt  2 root whoopstie 4096 Jul 22 2014 crash/
drwxr-xr-x 64 root root  4096 Feb 28 22:55 lib/
drwxrwsr-x  2 root staff  4096 Apr 10 2014 local/
lwxrwxrwx  1 root root      9 Feb 28 15:26 lock -> /run/lock/
drwxrwxr-x 14 root syslog  4096 Feb 29 12:07 log/
drwxrwsr-x  2 root mail   4096 Jul 22 2014 mail/
drwxrwsrwt  2 root whoopstie 4096 Jul 22 2014 metrics/
drwxr-xr-x  2 root root  4096 Jul 22 2014 opt/
lwxrwxrwx  1 root root      4 Feb 28 15:26 run -> /run/
drwxr-xr-x  9 root root  4096 Jul 22 2014 spool/
drwxrwsrwt  2 root root  4096 Feb 29 12:11 tmp/
priyanka@ubuntu:/var$
```

Step 2: Create a different file with a daemon services name, such as vsftpd or sshd, with a message you want to print on terminal accessing those services.

```
priyanka@ubuntu:/var$ cd banner
priyanka@ubuntu:/var/banner$ ll
total 12
drwxr-xr-x  2 root root 4096 Feb 29 01:51 ./
drwxr-xr-x 14 root root 4096 Feb 29 01:49 ../
-rw-r--r--  1 root root  18 Feb 29 01:51 vsftpd
priyanka@ubuntu:/var/banner$
```

Message written in vsftpd file:



```
priyanka@ubuntu: /var/banner
You have logged in through FTP
~
~
~
```

Step 3: Edit the host's.allow file


```
ALL : 192.168.49.134 : banners /var/banner
```

Test:

```
priyanka@ubuntu: ~  
priyanka@ubuntu:~$ ftp 192.168.49.135  
Connected to 192.168.49.135.  
You have logged in through FTP  
ftp>
```

When the FTP service is used the message in the VSFTPD file is printed accordingly.

Advantages:

1. Client is unaware of the TCP Wrappers in use.
2. Comparatively easy to set up.
3. Open source.

Disadvantages:

1. Does not check for file integrity or internal attack that have been caused by inside intruders.

Chapter IV: Conclusion and Future Work

There are numerous selection criteria based on which an IDS/IPS can be chosen. No IDS is “the best” because every IDS has its own features and different technologies. Selection of an IDS depends on requirements solely.

Host-based vs. Network-based IDS

Host-based IDSs utilize log files to determine an attack by pattern matching using key system files like checksums, hash, etc. Conversely, a network-based IDS will use network traffic to determine attacks by having the network adapter running on promiscuous mode. Network-based IDS are capable of even detecting the rejected attacks as they monitor the network traffic, and will be able to trace any suspicious activity that took place or was attempted. Unlike host-based IDS, an attack on the host will not generate any log files.

That being said, evidence can be manipulated or erased in the case of host-based IDSs; this is very difficult in the case of network-based IDSs. As a host-based IDS is installed on the host, it depends its respective OS to ensure the it will function effectively on that system. Network-based IDSs are installed on the network, are OS independent, and monitor all nodes and devices on that network. Any notification from the IDS can be configured as per requirement, such as console alarms or emails. Upon detection, a host-based IDS can delete the user account, disable access, and terminate user login. Alternatively, network-based IDS can reset the connection, reconfigure the firewall, and are comparatively low in cost as they go into the network. Host-based IDSs are deployed in each host and would cost comparatively more.

Table 4.1. Generic Comparison of Host-based IDS and Network-based IDS

	Host based	Network based
Attack signature	Log files	Network traffic
Process	Pattern matching with key system files via checksums, etc	Network adapter running in promiscuous mode
Rejected attacked	Cannot detect	Detects
Removing evidence	Easier	Not very easy
OS independence	Depends on OS	Not dependent on OS in most IDS
Functionality	Specific type of systems	Can protect a server running multiple services
System activities	Logs very well	Very difficult to provide this level of details
Verification	Verifies an attack with few false positives	Provides early warning
Notification	Alarm to console, email, etc	Alarm, email, view active session
Active response	Terminate user login, disable account	Connection reset, reconfigure firewall
Cost of ownership	Higher for an enterprise as it requires software to be	Comparatively lower as it runs on a dedicated server

	managed on various hosts	
Disabling the IDS	Attacker can disable the IDS	Very responsive even before attacker attempts to do so

According to the implementation done in the paper below is the comparison at each IDS level:

TCP Wrappers was most easy to implement followed by tripwire and then Bro IDS. TCP Wrappers and tripwire are host based IDS whereas BRO IDS is a network based IDS. Though BRO was complicated to set up, that is one of the most powerful IDS known. Customization is quite high for Bro IDS. TCP wrappers is one of the most basic IDS that can be implemented by anyone who have basic UNIX knowledge and need security from unwanted sources. All the IDS I tested on are open source and has lot of documentation on the internet which helps in setting them up. Logging level is quite granular and detailed in case of Bro followed by Tripwire and hardly there are any logs for TCP Wrappers.

Table 4.2. IDS Comparison

	Tripwire	TCP wrapper	BRO
Installation/configuration	Medium	Easy	Difficult
OS	Unix	Any	Unix
Type of IDS	Host based	Host based	Network based
Customization	Medium	Low	Very High
Open source	Yes	Yes	yes
Intelligent	Very Intelligent	Medium	Very Intelligent
Logs	Very Detailed	Not many logs	Very detailed

Below are a few features listed that can be taken into consideration while choosing an IDS. Features of host-based IDS:

- 1) Verifies success/failure of an attack using logs containing events that actually happened.
- 2) Monitors specific system activity by recording file accesses, changes to permissions, etc
- 3) A host-based IDS can detect an attack that a network based IDS cannot such as attacks from keyboards or key logger of a critical server.
- 4) This is well suited for encrypted environment by residing on hosts
- 5) This has prompt response when there is an interruption from the OS whenever there is a log file entry. This reduces the time between attack recognition and response
- 6) Host based IDS does not require any additional hardware
- 7) This can be configured on existing web servers

Features of a network-based IDS:

- 1) A network-based IDS makes it more difficult for an attacker to remove any evidence of

the attack because it uses live network for detection.

- 2) It has real time detection response and faster notifications.
- 3) It can detect unsuccessful attacks placed outside of firewall as well which never hit the host
- 4) This is OS independent

Future Work

An easier tool can be developed where an end user can input his requirements and the tool suggests few IDS that can be used. And also script files to set up the basic working IDS on the system that can be customized later on.

References

- [1] Allen, J., Christie, A., Fithen, W., McHugh, J., & Pickel, J. (2000). *State of the practice of intrusion detection technologies* (No. CMU/SEI-99-TR-028). Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst.
- [2] Alminshid, K., & Omar, M. N. (2013, September). Detecting backdoor using stepping stone detection approach. In *Informatics and Applications (ICIA), 2013 Second International Conference on* (pp. 87-92). IEEE.
- [3] Amoroso, E., & Kwapniewski, R. (1998, December). A selection criteria for intrusion detection systems. In *Computer Security Applications Conference, 1998. Proceedings. 14th Annual* (pp. 280-288). IEEE.
- [4] Debar, H. (2000). An introduction to intrusion-detection systems. *Proceedings of Connect, 2002*, 1-18.
- [5] Einwechter, N. (2002). Preventing and detecting insider attacks using IDS. *SecurityFocus, March*.
- [6] Heady, R., Luger, G. F., Maccabe, A., & Servilla, M. (1990). *The architecture of a network level intrusion detection system* (pp. 3-6). University of New Mexico. Department of Computer Science. College of Engineering.
- [7] Minella, J. J. (2010, November). IDS vs. IPS: How to know when you need the technology. Retrieved from <http://searchsecurity.techtarget.com/tip/IDS-vs-IPS-How-to-know-when-you-need-the-technology>.
- [8] Kent, K., & Warnock, M. (2004). *Intrusion Detection Tools Report, 4th Edition*. Herndon, VA: Information Assurance Technology Analysis Center (IATAC).

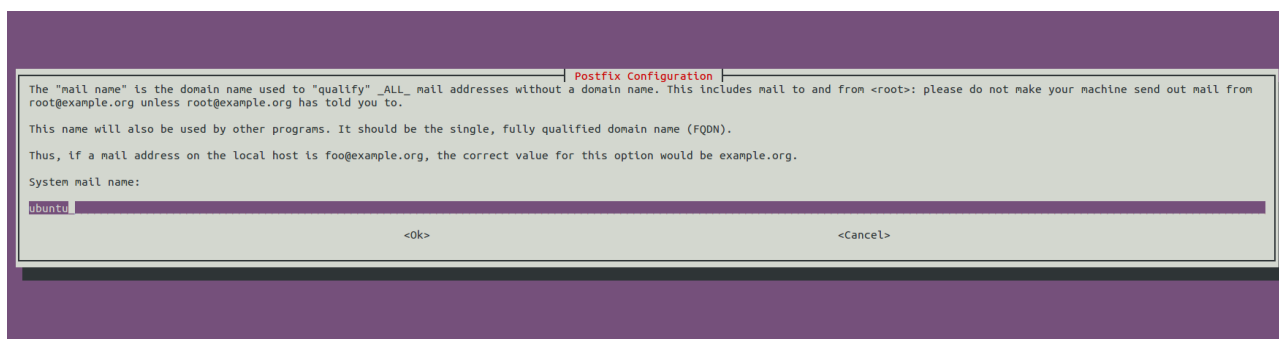
- [9] Kim, G. H., & Spafford, E. H. (1994, November). The design and implementation of tripwire: A file system integrity checker. In *Proceedings of the 2nd ACM Conference on Computer and Communications Security* (pp. 18-29). ACM.
- [10] Mehra, P. (2012). A brief study and comparison of snort and bro open source network intrusion detection systems. *International Journal of Advanced Research in Computer and Communication Engineering*, 1(6), 383-386.
- [11] Schreiber, J. (2014, January 13). Open Source Intrusion Detection Tools: A Quick Overview. Retrieved from <https://www.alienvault.com/blogs/security-essentials/open-source-intrusion-detection-tools-a-quick-overview>.
- [12] Shahrestani, A., Feily, M., Masood, M., & Muniandy, B. (2012, November). Visualization of invariant bot behavior for effective botnet traffic detection. In *Telecommunication Technologies (ISTT), 2012 International Symposium on* (pp. 325-330). IEEE.
- [13] Sylve, J., Case, A., Marziale, L., & Richard, G. G. (2012). Acquisition and analysis of volatile memory from android devices. *Digital Investigation*, 8(3-4), 175-184.
- [14] Zou, C. C., Towsley, D., Gong, W., & Cai, S. (2005, June). Routing worm: A fast, selective attack worm based on ip address information. In *Proceedings of the 19th Workshop on Principles of Advanced and Distributed Simulation* (pp. 199-206). IEEE Computer Society.
- [15] 48.5. TCP Wrappers and XINETD. (n.d.). Retrieved from https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/deployment_guide/ch-tcpwrappers

Appendix

Tripwire:

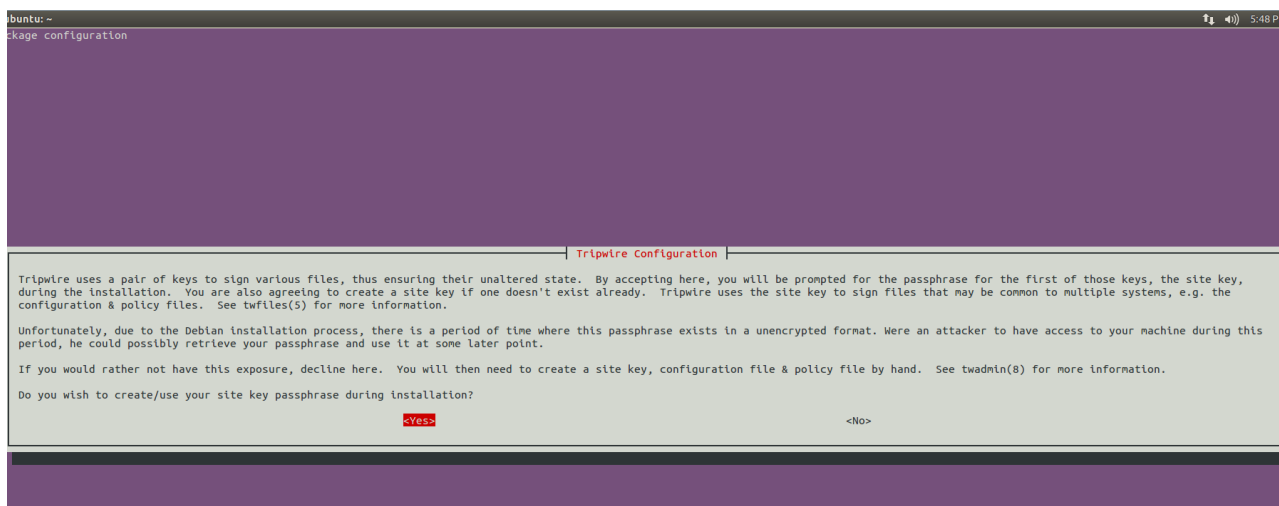
Install tripwire

sudo apt-get install tripwire - It is used to install tripwire.



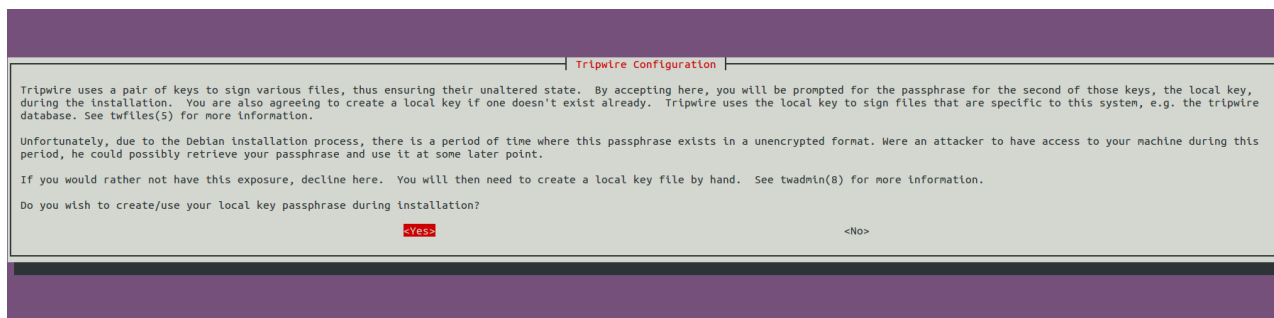
Use site key phrase:

This key is used to secure the configuration files.



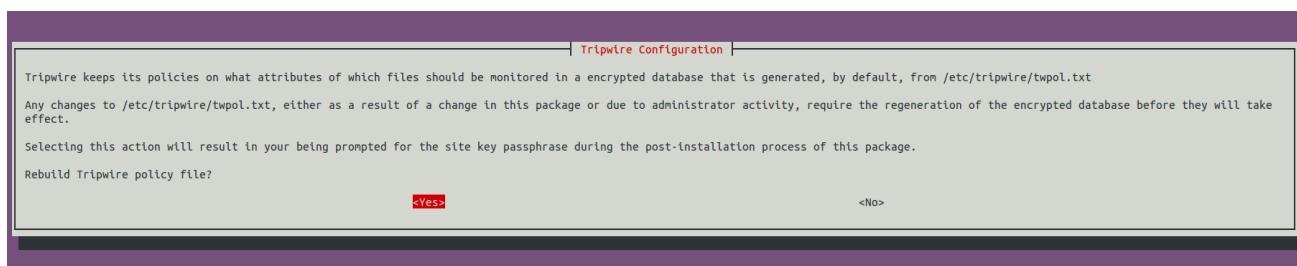
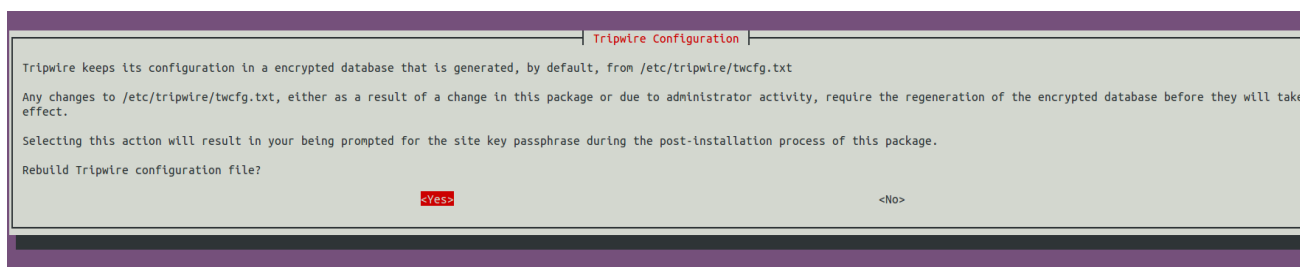
Local key phrase:

This key is used on each machine to run the binaries. This is necessary to ensure that our binaries are not run without our consent.



Select **yes** to create a local key passphrase.

Tripwire Configuration:



Select **yes** to rebuild the configuration file, in addition to the policy file.

Console installation:

The figure below displays the installation happening in console background.

```

@ubuntu:~
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main postfix amd64 2.11.0-1ubuntu1 [1,084 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu/ trusty/universe tripwire amd64 2.4.2.2-3 [1,407 kB]
Fetched 2,492 kB in 2s (1,085 kB/s)
Preconfiguring packages ...
Selecting previously unselected package postfix.
(Reading database ... 163879 files and directories currently installed.)
Preparing to unpack .../postfix_2.11.0-1ubuntu1_amd64.deb ...
Unpacking postfix (2.11.0-1ubuntu1) ...
Selecting previously unselected package tripwire.
Preparing to unpack .../tripwire_2.4.2.2-3_amd64.deb ...
Unpacking tripwire (2.4.2.2-3) ...
Processing triggers for ufw (0.34-rc-0ubuntu2) ...
Processing triggers for ureadahead (0.100.0-16) ...
ureadahead will be reprofiled on next reboot
Processing triggers for man-db (2.6.7.1-1) ...
Setting up postfix (2.11.0-1ubuntu1) ...
Adding group 'postfix' (GID 125) ...
Done.
Adding system user 'postfix' (UID 116) ...
Adding new user 'postfix' (UID 116) with group 'postfix' ...
Not creating home directory '/var/spool/postfix'.
Creating /etc/postfix/dynamicmaps.cf
Adding tcp map entry to /etc/postfix/dynamicmaps.cf
Adding sqlite map entry to /etc/postfix/dynamicmaps.cf
Adding group 'postdrop' (GID 126) ...
Done.
setting myhostname: ubuntu
setting alias maps
setting alias database
mailname is not a fully qualified domain name. Not changing /etc/mailname.
setting destinations: ubuntu, localhost.localdomain, , localhost
setting relayhost:
setting mynetworks: 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
setting mailbox_size_limit: 0
setting recipient_delimiter: +
setting inet_interfaces: all
setting inet_protocols: all
/etc/aliases does not exist, creating it.
WARNING: /etc/aliases exists, but does not have a root alias.

Postfix is now set up with a default configuration. If you need to make
changes, edit
/etc/postfix/main.cf (and others) as needed. To view Postfix configuration
values, see postconf(1).

After modifying main.cf, be sure to run '/etc/init.d/postfix reload'.

Running newaliases

```

Local key-phrase set-up:

```

Get local passphrase
Tripwire uses two different keys for authentication and encryption of files. The local key is used to protect files specific to the local machine, such as the Tripwire database. The local key may
also be used for signing integrity check reports.

You are being prompted for this passphrase because no local key file currently exists.

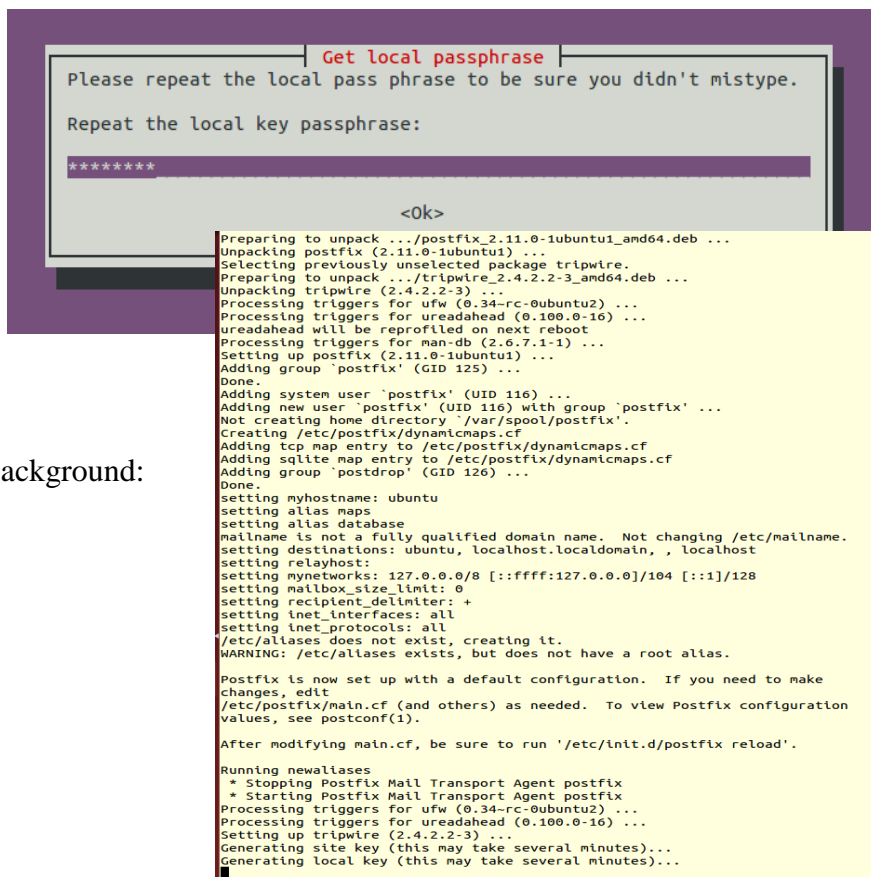
Remember this passphrase; it is not stored anywhere!

Enter local key passphrase:
*****

<ok>

```

Repeat password screen:



```

Get local passphrase
Please repeat the local pass phrase to be sure you didn't mistype.
Repeat the local key passphrase:
*****
<Ok>

Preparing to unpack .../postfix_2.11.0-1ubuntu1_amd64.deb ...
Unpacking postfix (2.11.0-1ubuntu1) ...
Selecting previously unselected package tripwire.
Preparing to unpack .../tripwire_2.4.2.2-3_amd64.deb ...
Unpacking tripwire (2.4.2.2-3) ...
Processing triggers for ufw (0.34-rc-0ubuntu2) ...
Processing triggers for ureadahead (0.100.0-16) ...
ureadahead will be reprofiled on next reboot
Processing triggers for man-db (2.6.7.1-1) ...
Setting up postfix (2.11.0-1ubuntu1) ...
Adding group 'postfix' (GID 125) ...
Done.
Adding system user 'postfix' (UID 116) ...
Adding new user 'postfix' (UID 116) with group 'postfix' ...
Not creating home directory '/var/spool/postfix'.
Creating /etc/postfix/dynamicmaps.cf
Adding tcp map entry to /etc/postfix/dynamicmaps.cf
Adding sqlite map entry to /etc/postfix/dynamicmaps.cf
Adding group 'postdrop' (GID 126) ...
Done.
setting myhostname: ubuntu
setting alias maps
setting alias database
mailname is not a fully qualified domain name. Not changing /etc/mailname.
setting destinations: ubuntu, localhost.localdomain, , localhost
setting relayhost:
setting mynetworks: 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
setting mailbox_size_limit: 0
setting recipient_delimiter: +
setting inet_interfaces: all
setting inet_protocols: all
/etc/aliases does not exist, creating it.
WARNING: /etc/aliases exists, but does not have a root alias.

Postfix is now set up with a default configuration. If you need to make
changes, edit
/etc/postfix/main.cf (and others) as needed. To view Postfix configuration
values, see postconf(1).

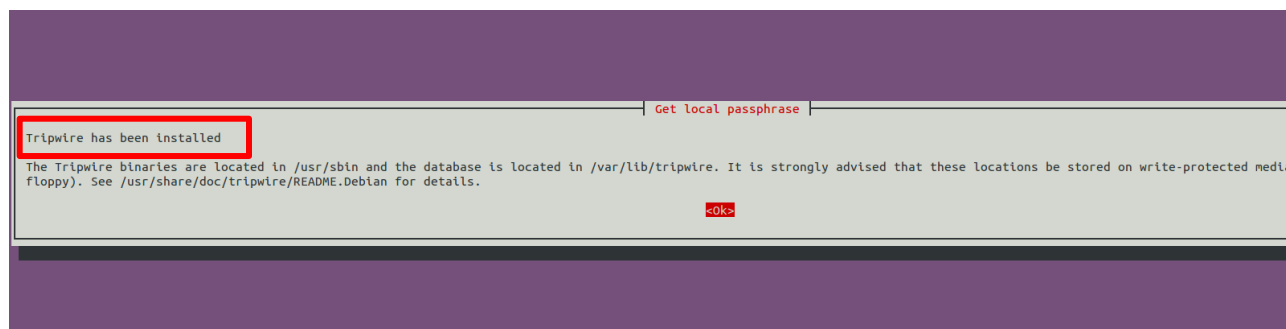
After modifying main.cf, be sure to run '/etc/init.d/postfix reload'.

Running newaliases
* Stopping Postfix Mail Transport Agent postfix
* Starting Postfix Mail Transport Agent postfix
Processing triggers for ufw (0.34-rc-0ubuntu2) ...
Processing triggers for ureadahead (0.100.0-16) ...
Setting up tripwire (2.4.2.2-3) ...
Generating site key (this may take several minutes)...
Generating local key (this may take several minutes)...

```

Console installation on the background:

Tripwire installation completed:



```

Get local passphrase
Tripwire has been installed
The Tripwire binaries are located in /usr/sbin and the database is located in /var/lib/tripwire. It is strongly advised that these locations be stored on write-protected media
floppy). See /usr/share/doc/tripwire/README.Debian for details.
<ok>

```

Once the installation of Tripwire is done, we check for the following files in `/etc/tripwire`:

cd /etc/tripwire; ll

```
priyanka@ubuntu:/etc$ cd tripwire
priyanka@ubuntu:/etc/tripwire$ ll
total 52
drwxr-xr-x  2 root root  4096 Mar 30 17:55 ./
drwxr-xr-x 131 root root 12288 Mar 30 17:56 ../
-rw-----  1 root root   931 Mar 30 17:54 site.key
-rw-r--r--  1 root root  4586 Mar 30 17:55 tw.cfg
-rw-r--r--  1 root root   510 Jan 29  2014 twcfg.txt
-rw-r--r--  1 root root  4159 Mar 30 17:55 tw.pol
-rw-r--r--  1 root root  6057 Jan 29  2014 twpol.txt
-rw-----  1 root root   931 Mar 30 17:55 ubuntu-local.key
priyanka@ubuntu:/etc/tripwire$ █
```

Executable Script for Tripwire

1. Installing Tripwire in Ubuntu 12.04, run following commands:

sudo apt-get update

sudo apt-get install tripwire

While installing Tripwire, it will ask you to setup the site-key passphrase and local-key passphrase.

2. Check that the Tripwire is installed in the directory path */etc/tripwire/*.
3. Initialize the database:

sudo tripwire --init

4. Next, run the *check* command to see what errors we are getting so we may accordingly update the policy file.

sudo tripwire --check

5. Now we will remove all the above error messages from our Tripwire configuration. To do that, we must open a file with the name *twpol.txt* and edit that file using *vi* command.

sudo vi /etc/tripwire/twpol.txt

Now we want to change and modify this file. You must comment out the files/rules that you don't need with a # sign. You put the # sign in front of each line you don't need.

6. Now save all the changes made to *twpol.txt* file using following command:

```
:wq
```

7. Now you must tell Tripwire that all these changes have been made, and that you have tailored *twpol.txt* to match your Ubuntu system. Write the following code:

```
sudo twadmin -m P /etc/tripwire/twpol.txt
```

8. Recreate the database to notice the changes:

```
sudo tripwire -init
```

9. Now you need to verify the configuration:

```
sudo tripwire --check
```

10. Basic integrity check:

```
sudo tripwire --check
```

11. Setup an email notification.

```
sudo apt-get install mailutils
```

```
sudo tripwire --check | mail -s "Tripwire report for 'user@ubuntu -n' "  
priyanka97562@gmail.com
```

12. Perform an interactive check:

```
sudo tripwire --check --interactive
```

13. Automate tripwire with *cron*:

```
sudo crontab -e
```

You will be able to update the file to automate the Tripwire. To have Tripwire run at 3:30am every day and send that report to email address of the user, we can place a line like this in our file:

```
30 3 * * * /usr/sbin/tripwire --check | mail -s "Tripwire report for `uname -n`" priyanka97562@gmail.com
```

14. Automated integrity check by editing root's *crontab* file:

```
sudo gedit /etc/crontab
```

If you want a daily integrity check at 3 am, add the following code in the *crontab* file:

```
0 3 * * * /usr/sbin/tripwire --check
```

15. Printing Tripwire reports:

```
sudo twprint -m r --twrfile /var/lib/tripwire/report/Ubuntu-20160325-001412.twr
```

16. To update database after integrity check

```
tripwire --update --twrfile /var/lib/tripwire/report/<name>.twr
```

After running the above command, Tripwire will show you the particular report using the default text editor. All proposed updates to the Tripwire database start with a [x] before the file name. If you want to specifically exclude a valid violation from being added to the Tripwire database, remove the "x" from the box and then save the file.

BRO – IDS

1) Updating the OS

Once you are logged into your VPS, you should ensure your OS is up to date by executing the following command as root:

apt-get update && apt-get upgrade

```
priyanka@ubuntu:~$ sudo apt-get update && apt-get upgrade
Ign http://us.archive.ubuntu.com trusty InRelease
Get:1 http://us.archive.ubuntu.com trusty-updates InRelease [65.9 kB]
Ign http://extras.ubuntu.com trusty InRelease
Get:2 http://security.ubuntu.com trusty-security InRelease [65.9 kB]
Get:3 http://extras.ubuntu.com trusty Release.gpg [72 B]
Hit http://us.archive.ubuntu.com trusty-backports InRelease
Hit http://extras.ubuntu.com trusty Release
Hit http://us.archive.ubuntu.com trusty Release.gpg
Get:4 http://us.archive.ubuntu.com trusty-updates/main Sources [272 kB]
Hit http://extras.ubuntu.com trusty/main Sources
Hit http://extras.ubuntu.com trusty/main amd64 Packages
Get:5 http://security.ubuntu.com trusty-security/main Sources [110 kB]
Hit http://extras.ubuntu.com trusty/main i386 Packages
Get:6 http://us.archive.ubuntu.com trusty-updates/restricted Sources [5,352 B]
Get:7 http://us.archive.ubuntu.com trusty-updates/universe Sources [153 kB]
Get:8 http://us.archive.ubuntu.com trusty-updates/multiverse Sources [5,928 B]
Get:9 http://us.archive.ubuntu.com trusty-updates/main amd64 Packages [752 kB]
56% [9 Packages 201 kB/752 kB 27%] [5 Sources 39.1 kB/110 kB 35%] [Waiting for
```

If the kernel was updated during this process, you should reboot your instance prior to proceeding.

- 2) We need to install the required dependencies:

```
apt-get install cmake make gcc g++ flex bison libpcap-dev libgeoip-dev libssl-dev python-
dev zlib1g-dev libmagic-dev swig2.0
```

- 3) we need to install the GeoLite database before starting Bro;

➔ `wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz`

```
priyanka@ubuntu:~$ sudo wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
--2016-04-17 18:59:23-- http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
Resolving geolite.maxmind.com (geolite.maxmind.com)... 141.101.115.190, 141.101.114.190, 2400:cb00:2048:1::8d65:72be, ...
Connecting to geolite.maxmind.com (geolite.maxmind.com)|141.101.115.190|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12751589 (12M) [application/octet-stream]
Saving to: 'GeoLiteCity.dat.gz'

57% [=====
```

➔ `wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCityv6-beta/GeoLiteCityv6.dat.gz`


```

priyanka@ubuntu:~$ sudo wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
--2016-04-17 18:59:23-- http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
Resolving geolite.maxmind.com (geolite.maxmind.com)... 141.101.115.190, 141.101.114.190, 2400:cb00:2048:1::8d65:72be, ...
Connecting to geolite.maxmind.com (geolite.maxmind.com)|141.101.115.190|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12751589 (12M) [application/octet-stream]
Saving to: 'GeoLiteCity.dat.gz'

100%[=====] 12,751,589  2.42MB/s   in 5.6s
2016-04-17 18:59:29 (2.18 MB/s) - 'GeoLiteCity.dat.gz' saved [12751589/12751589]

priyanka@ubuntu:~$ sudo wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCityv6-beta/GeoLiteCityv6.dat.gz
--2016-04-17 19:00:28-- http://geolite.maxmind.com/download/geoip/database/GeoLiteCityv6-beta/GeoLiteCityv6.dat.gz
Resolving geolite.maxmind.com (geolite.maxmind.com)... 141.101.114.190, 141.101.115.190, 2400:cb00:2048:1::8d65:73be, ...
Connecting to geolite.maxmind.com (geolite.maxmind.com)|141.101.114.190|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13061025 (12M) [application/octet-stream]
Saving to: 'GeoLiteCityv6.dat.gz'

100%[=====] 13,061,025  3.38MB/s   in 3.8s
2016-04-17 19:00:32 (3.31 MB/s) - 'GeoLiteCityv6.dat.gz' saved [13061025/13061025]

priyanka@ubuntu:~$

```

➔ gunzip GeoLiteCity.dat.gz

```
priyanka@ubuntu:~$ sudo gunzip GeoLiteCity.dat.gz
```

➔ gunzip GeoLiteCityv6.dat.gz

```
priyanka@ubuntu:~$ sudo gunzip GeoLiteCityv6.dat.gz
```

- 4) we need to move the database files to the /usr/share/GeoIP/ directory by executing the following commands:

➔ mv GeoLiteCity.dat /usr/share/GeoIP/GeoLiteCity.dat

```
priyanka@ubuntu:~$ sudo mv GeoLiteCity.dat /usr/share/GeoIP/GeoLiteCity.dat
```

➔ mv GeoLiteCityv6.dat /usr/share/GeoIP/GeoLiteCityv6.dat

```
priyanka@ubuntu:~$ sudo mv GeoLiteCityv6.dat /usr/share/GeoIP/GeoLiteCityv6.dat
```

- 5) Now we need to create a link for the GeoLiteCity.dat and GeoLiteCityv6.dat files to GeoIPCity.dat and GeoIPCityv6.dat respectively.

➔ ln -s /usr/share/GeoIP/GeoLiteCity.dat /usr/share/GeoIP/GeoIPCity.dat

```
priyanka@ubuntu:~$ sudo ln -s /usr/share/GeoIP/GeoLiteCity.dat /usr/share/GeoIP/GeoIPCity.dat
```

➔ ln -s /usr/share/GeoIP/GeoLiteCityv6.dat /usr/share/GeoIP/GeoIPCityv6.dat

- 6) Installing Bro-IDS

➔ wget http://www.bro.org/downloads/release/bro-2.4.1.tar.gz

```
priyanka@ubuntu:/usr/share/GeoIP$ wget http://www.bro.org/downloads/release/bro-2.4.1.tar.gz
```

```
priyanka@ubuntu:/usr/share/GeoIP$ wget http://www.bro.org/downloads/release/bro-2.4.1.tar.gz
--2016-04-17 19:12:35-- http://www.bro.org/downloads/release/bro-2.4.1.tar.gz
Resolving www.bro.org (www.bro.org)... 192.150.187.43
Connecting to www.bro.org (www.bro.org)|192.150.187.43|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.bro.org/downloads/release/bro-2.4.1.tar.gz [following]
--2016-04-17 19:12:36-- https://www.bro.org/downloads/release/bro-2.4.1.tar.gz
Connecting to www.bro.org (www.bro.org)|192.150.187.43|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 15010455 (14M) [application/x-gzip]
Saving to: 'bro-2.4.1.tar.gz'

90% [=====
```

➔ tar -xvzf bro-2.4.1.tar.gz

- 7) To build the application, we change directories with the cd bro-2.4.1 command and set the directory we intend to install the Bro-IDS application by setting --prefix= option.

➔ cd bro-2.2

➔ ./configure --prefix=/nsm/bro

```
priyanka@ubuntu:/usr/share/GeoIP/bro-2.4.1$ ./configure --prefix=/nsm/bro
```

```
-- Not building broccoli-ruby bindings

=====| Broccoli Build Summary |=====
Install prefix: /nsm/bro
Library prefix: /nsm/bro/lib
Debug mode: false
Shared libs: true
Static libs: true
Config file: /nsm/bro/etc/broccoli.conf
Packet support: true
CC: /usr/bin/cc
CFLAGS: -Wall -Wno-unused -O2 -g -DNDEBUG
CPP: /usr/bin/cpp

=====| Bro Build Summary |=====
Install prefix: /nsm/bro
Bro Script Path: /nsm/bro/share/bro
Debug mode: false
CC: /usr/bin/cc
CFLAGS: -Wall -Wno-unused -O2 -g -DNDEBUG
CXX: /usr/bin/c++
CXXFLAGS: -Wall -Wno-unused -O2 -g -DNDEBUG
CPP: /usr/bin/c++

Broker:
Broccoli: true
Broctl: true
Aux. Tools: true

GeoIP: true
gperftools found: false
tcmalloc: false
debugging: false
jemalloc: false

-----
-- Configuring done
-- Generating done
-- Build files have been written to: /usr/share/GeoIP/bro-2.4.1/build
priyanka@ubuntu:/usr/share/GeoIP/bro-2.4.1$
```

➔ Make

```
priyanka@ubuntu: /usr/share/GeoIP/bro-2.4.1$ sudo make
```

➔ make install

```
-- Up-to-date: /nsm/bro/share/broctl/scripts/set-bro-path
-- Up-to-date: /nsm/bro/share/man/man8/broctl.8
-- Installing: /nsm/bro/share/bro
-- Installing: /nsm/bro/share/bro/broctl
-- Up-to-date: /nsm/bro/share/bro/broctl/standalone.bro
-- Up-to-date: /nsm/bro/share/bro/broctl/__load__.bro
-- Up-to-date: /nsm/bro/share/bro/broctl/main.bro
-- Up-to-date: /nsm/bro/share/bro/broctl/auto.bro
-- Up-to-date: /nsm/bro/share/bro/broctl/check.bro
-- Up-to-date: /nsm/bro/share/bro/broctl/process-trace.bro
-- Installing: /nsm/bro/spool
-- Installing: /nsm/bro/spool/tmp
-- Installing: /nsm/bro/logs
-- Skipping: /nsm/bro/etc/broctl.cfg (already exists)
-- Skipping: /nsm/bro/etc/networks.cfg (already exists)
-- Skipping: /nsm/bro/etc/node.cfg (already exists)
-- Installing: /nsm/bro/lib/broctl/SubnetTree.py
-- Installing: /nsm/bro/lib/broctl/_SubnetTree.so
-- Set runtime path of "/nsm/bro/lib/broctl/_SubnetTree.so" to "/nsm/bro/lib"
-- Installing: /nsm/bro/bin/capstats
-- Set runtime path of "/nsm/bro/bin/capstats" to "/nsm/bro/lib"
-- Up-to-date: /nsm/bro/bin/trace-summary
-- Up-to-date: /nsm/bro/share/man/man1/trace-summary.1
-- Installing: /nsm/bro/bin/bro-cut
-- Up-to-date: /nsm/bro/share/man/man1/bro-cut.1
-- Skipping: /nsm/bro/etc/broccoli.conf (already exists)
-- Installing: /nsm/bro/bin/broccoli-config
-- Installing: /nsm/bro/lib/libbroccoli.so.5.1.0
-- Up-to-date: /nsm/bro/lib/libbroccoli.so.5
-- Up-to-date: /nsm/bro/lib/libbroccoli.so
-- Set runtime path of "/nsm/bro/lib/libbroccoli.so.5.1.0" to "/nsm/bro/lib"
-- Installing: /nsm/bro/lib/libbroccoli.a
-- Installing: /nsm/bro/include/broccoli.h
-- Up-to-date: /nsm/bro/lib/broctl/broccoli.py
-- Installing: /nsm/bro/lib/broctl/_broccoli_intern.so
-- Set runtime path of "/nsm/bro/lib/broctl/_broccoli_intern.so" to "/nsm/bro/lib"
-- Installing: /nsm/bro/lib/broctl/broccoli_intern.py
make[1]: Leaving directory '/usr/share/GeoIP/bro-2.4.1/build'
priyanka@ubuntu: /usr/share/GeoIP/bro-2.4.1$
```

8) Add bro to your PATH.

```
export PATH=/nsm/bro/bin:$PATH
```

```
priyanka@ubuntu: /usr/share/GeoIP/bro-2.4.1$ export PATH=/nsm/bro/bin:$PATH
```

9) modify the following 3 files:

➔ \$PREFIX/etc/node.cfg -> Configure the network interface to monitor (i.e.

interface=eth0)

➔ \$PREFIX/etc/networks.cfg -> Configure the local networks (i.e. 10.0.0.0/8 Private IP

space)

➔ \$PREFIX/etc/broctl.cfg -> Change the MailTo address and the log rotation

10) Starting Bro-IDS

We need to launch the broctl shell, from where you can execute bro commands. As root type broctl.

```
priyanka@ubuntu:/usr/share/GeoIP/bro-2.4.1$ broctl
Hint: Run the broctl "deploy" command to get started.

Welcome to BroControl 1.4

Type "help" for help.

[BroControl] > █
```

TCP Wrappers

Configuring the server:

For the VM to act as server it must be containing daemon services such as VSFTPD, SSHD, TELNETD etc use services such a FTP, SSH, TELNET to connect to a server from the client respectively.

Step 1: Installation of VSFTPD

Vsftpd (very secure FTP daemon) is an FTP server for unix-like systems.

Starting the service:

```
priyanka@ubuntu:~$ sudo service vsftpd restart
vsftpd stop/waiting
vsftpd start/running, process 3753
priyanka@ubuntu:~$
```

Similarly, TELNETD and SSHD services have to installed on server side.

Step 2: Installation of XINETD TELNTD

```
priyanka@ubuntu:~$ sudo apt -get install xinetd telnetd
[sudo] password for priyanka:
E: Command line option 'g' [from -get] is not known.
priyanka@ubuntu:~$ sudo apt -get install xinetd telnetd
E: Command line option 'g' [from -get] is not known.
priyanka@ubuntu:~$ sudo apt-get install xinetd telnetd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  telnetd xinetd
0 upgraded, 2 newly installed, 0 to remove and 566 not upgraded.
Need to get 145 kB of archives.
After this operation, 455 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty/main xinetd amd64 1:2.3.15-3ubuntu1 [104 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu/ trusty/universe telnetd amd64 0.17-36build2 [40.1 kB]
Fetched 145 kB in 0s (314 kB/s)
Selecting previously unselected package xinetd.
(Reading database ... 166640 files and directories currently installed.)
Preparing to unpack .../xinetd_1%3a2.3.15-3ubuntu1_amd64.deb ...
Unpacking xinetd (1:2.3.15-3ubuntu1) ...
```

Step 3 : Installation of SSHD service

```
priyanka@ubuntu:~$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 566 not upgraded.
```

Step 4 : Editing the vsftpd.conf file

Tcp_wrappers line have to be added to vsftpd.conf file to enable the tcp_wrappers

```
#
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
# (Warning! chroot'ing can be very dangerous. If using chroot, make sure that
# the user does not have write access to the top level directory within the
# chroot)
#chroot_local_user=YES
#chroot_list_enable=YES
# (default follows)
#chroot_list_file=/etc/vsftpd.chroot_list
#
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "mirror" assume
# the presence of the "-R" option, so there is a strong case for enabling it.
#ls_recurse_enable=YES
#
# Customization
#
# Some of vsftpd's settings don't fit the filesystem layout by
# default.
#
# This option should be the name of a directory which is empty. Also, the
# directory should not be writable by the ftp user. This directory is used
# as a secure chroot() jail at times vsftpd does not require filesystem
# access.
secure_chroot_dir=/var/run/vsftpd/empty
#
# This string is the name of the PAM service vsftpd will use.
pam_service_name=vsftpd
#
# This option specifies the location of the RSA certificate to use for SSL
# encrypted connections.
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
# This option specifies the location of the RSA key to use for SSL
# encrypted connections.
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
tcp_wrappers=yes
```

151,1 Bot

Step 5 : Restarting the services :

```
priyanka@ubuntu:/etc$ sudo service vsftpd restart
stop: Unknown instance:
vsftpd start/running, process 3146
priyanka@ubuntu:/etc$ sudo service vsftpd status
vsftpd start/running, process 3146
```

IP Configuration details are listed below:

1) Server:

```
priyanka@ubuntu:/etc$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:6f:13:a1
          inet addr:192.168.49.133  Bcast:192.168.49.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe6f:13a1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4827 errors:0 dropped:0 overruns:0 frame:0
          TX packets:930 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2443091 (2.4 MB)  TX bytes:96181 (96.1 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:208 errors:0 dropped:0 overruns:0 frame:0
          TX packets:208 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:18406 (18.4 KB)  TX bytes:18406 (18.4 KB)
```

2) Client:

```
priyanka@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:8f:79:f1
          inet addr:192.168.49.134  Bcast:192.168.49.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe8f:79f1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3240 errors:0 dropped:0 overruns:0 frame:0
          TX packets:300 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:321664 (321.6 KB)  TX bytes:39129 (39.1 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:189 errors:0 dropped:0 overruns:0 frame:0
          TX packets:189 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:13591 (13.5 KB)  TX bytes:13591 (13.5 KB)
```