Alert Log Reduction

Submitted By

Siddharth Chowatiya

13MCEI06

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2015

Alert Log Reduction

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

(Information and Network Security)

Submitted By

Siddarth Chowatiya

(13MCEI06)

Guided By

Dr. Sharada Valiveti

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2015

Certificate

This is to certify that the major project entitled **"Alert Log Reduction"** submitted by **Siddharth Chowatiya (Roll No: 13MCEI06)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering (Information & Network Security) of Institute of Technology, Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. Sharada Valiveti
Guide & Associate Professor,
Coordinator M.Tech - INS,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Sanjay Garg
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr K Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Siddharth Chowatiya**, Roll. No. **13MCEI06**, give undertaking that the Major Project entitled "**Alert log reduction**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Siddharth Chowatiya

Date:

Place:

Endorsed by

Dr Sharada Valiveti

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Dr. Sharada Valiveti**, Associate Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for her valuable guidance and continual encouragement throughout this work. The appreciation and continual support she has imparted has been a great motivation to me in reaching a higher goal. Her guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr K Kotecha**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

See that you acknowledge each one who have helped you in the project directly or indirectly.

- Siddharth Chowatiya

13MCEI06

Abstract

One of the major issue of Intrusion Detection Systems (IDS) is the high rate of false alerts that the IDS generates. False alerts are the alerts which pretend to be the true alerts. False positives are critical problems of intrusion detection systems that use different techniques to detect network intrusions. The techniques or algorithms which are used in intrusion detection systems are unable to eliminate false alerts with short lifespan. Secondly, Intrusion Detection Systems (IDSs) can easily generate tons of alerts per day and from them up to 99% of alerts are false positives (i.e. alerts that are triggered incorrectly). This makes it extremely difficult for network administrators to analyze and react to attacks. To overcome this problem a new algorithm is used for handling IDS alerts more efficiently.

Abbreviations

IDS	Intrusion Detection System.
IPS	Intrusion Prevention System.
OOP	Object Oriented Programming.
RIPPER	Repeated Incremental Pruning to Produce Error Reduction.
DARPA	Defense Advanced Research Projects Agency.
IBL	Instance Based Learning.
NBA	Network Behavioral Analysis.

Contents

List of Figures

Chapter 1

Introduction

1.1 Alert

Alerts are kind of warning alarm for security analysts or network administrators of what is discovered as presenting security threat to the organization or network environment they are monitoring in Intrusion Detection System[12].

1.1.1 Types of Alerts

- **True Positive:**

  It is the alert which shows a actual attack which triggers an IDS to produce an alarm. True Positive alerts are true alerts which are useful for Intrusion Detection System.

- **False Negative:**

  This kind of alert signals an IDS to produce an alarm when no attack has taken place. False Negative are unwanted alerts and the false negative alert generation ratio is higher in most of the organizations. the alert log size exceeds because of these False alerts.

1.1.2 Working of Alerts

Firewall Analyzer generates alerts automatically and notifies network administrators, when there is a security breach attempt in the network or abnormal traffic. It detects the anomalous traffic behavior. Alerts can be set up to notify administrators/operators by Email. Alerts can be notified using SMS and a program can be run to initiate other actions

(for example: generate an SNMP trap to be directed to Network/Asset management Application). You can also delete the Alert and deleting bulk alert is possible for a particular profile.

1.1.3 Alerting Techniques

1. Anomaly-based Alerting : In anomaly-based alerting technique, alerts are used for detecting irregular traffic behavior of network packets. In Network Behavioral Analysis(NBA), this anomaly based alerts can be used. In anomaly-based alerting the filters can be changed or put by network administrator like some threshold value can be set to trigger anomaly in the network, set the first concern or order of alerts etc. The alerts are notified by E-mails to the network administrators.

2. Threshold-based Alerting : In Threshold-based Alerting, we can set up alerts for each devices in the Network. We can select the exact value/criteria to generate the alert. we also can set the priority of the alert to be analyzed.We also can set the notification type of alerts to network administrators like it can be E-mail notification or any other notification.

3. Signature-based Alerting : In signature-based alerting technique, The network administrator examine the network communications, identify heuristics and patterns of common computer attacks and taking that to alert operators. The historical data of computer attacks are used as signature in this technique. All the network communications are first compared with these signatures and if it is same than it is showing the alert.

1.2 Need of alert log reduction

Increasing use of Internet among different types of people make security problems very important. Nowadays, using powerful tools is vital to keep networks safe and far away from attacker's harms. During the last few years, intrusion detection systems (IDS) have been used widely and their value as security elements have been revealed. Intrusion detection systems are the security devicels which give the protection in high scale.[4]. The objectives of IDS is differentiated among normal behavior of network or system and intrusion actions in the network or systems.

In test results, it is detected that IDS can easily generat thousands of security alerts per day, up to 99% of which are false alerts, i.e. alerts that are generated incorrectly by non-dangerous events. These false positive alerts have made it very tough for network administrator to analyze security position of the network. The response to vulnerable attacks are often slowed up because, the alerts generated for them are failed to notice due to large size of logged information, which is not yet analyzed and is often ignored. So how to reduce or remove false alerts is a major problem.

Now the best solution to these problems is to equip the system with a mechanism whereby, it can automatically identify the genuine alerts and log them. Hence, the use of different machine learning algorithms is foreseen in such situations. These algorithms experience some common limitations as follows:

1. Almost every the algorithms need training dataset or alert signature values to build their alert reduction training dataset.

2. Most of the algorithms are used in offine mode, Hence delay in reaction is experienced.

3. The statistical algorithms cannot adapt in new situations[4].

A new method is proposed to overcome the drawback associated with machine learning techniques. In order to filter IDS alerts in a better way, all types of alerts are examined. Bulky alerts pose many problems for network administrators.Network administrators get confused with huge volume of alerts and it takes too much time for them to decide which alerts are false and which are true. So firstly, this algorithm will count the size of the usual true alerts and based on that, a value will be assigned, Based on these values the alerts will be prioritized. Second step is based on the abnormal behaviour of alerts. It will be neglected. So, the number of alerts will reduce. In the third step, the new alerts signature will be added in training set, so that next time for same alert it will not take too much time[7].

1.3 Applications Of Alert Log Reduction

This alert log is a very general problem for big organisations/Industries. So, alert log reduction can be used in this following applications.

1. This alert log reduction algorithm can be used in any firewall of LAN of Industry/Organization.
2. This algorithm can be used in any Network based IDS.
3. It can be used in any big Academic Institute where the generation of false alerts is very common issue.

1.4 Research Gaps

This false alert generation is a very common and big issue so that to overcome this problem some algorithms are already used. That algorithms are ainly of two types:

1. Anomaly based alert reduction
2. Signature based alert reduction

Anomaly based alert reduction is based on the impact of that attack and this algorithm is efficient than signature based algorithm. but it is time consuming so it is not that much effective.

Signature based alert reduction is based on the source and destination or type of attack and this algorithm is less time consuming. It reduces the size of alert log in less time period. So maximum number of organisations/industries use this algorithm.

These algorithms are having some problems that are still not solved. The first problem is that the bulky alerts. bulky alerts take too much time to get analyzed and so sometimes it becomes a very big problem where the false alert generation rate is too high. The other problem is training dataset. For any algorithm, we need initial training dataset and than it is used. so the creation of training dataset is also a big proble when new alert is found. The third is that maximum number of IDS uses the offline algorithm to reduce the alerts. so for new alert these algorithms do not have any signature. so it is also a very big problem.

Chapter 2

Objectives and Scope Of The Work

2.1 Objectives

The objectives of alert log reduction algorithm are as follows:

- Reduce the log size:

  The first objective of the algorithm is to reduce the log size. To reduce the size of the false alerts first we need to remove the unwanted data from the alerts. so that the analysis of all the alerts become faster than before. so the first part of the algorithm is reduction of the size of the alert.

- Reduce the false alert generation rate:

  The second objective of the algorithm is to reduce the generation of false alert rate. For that we need to analyze all the alert and we need to find the source of the false alert generation. we can directly ignore the alerts of that source machine and we can reduce a big amount of alerts.

2.2 Scope of the work

From the generated alerts, maximum number of alerts are useless or false alerts. Many alerts from them are showing that this system is not trusted system. but in real that alerts are not generated from the said system. Using this algorith we show that this alert is fake alert or not. To work on that alert reduction and alert log generation the data/alerts are taken from the firewall or IDS of any organisation. and after analysing these alerts the unwanted alerts are removed from that data. Here we are not developing IDS but after the alert generation of IDS we are using this algorithm to reduce the log generted by IDS. So this algorithm is proposed to reduce alert log size and rate of alerts to enhance the performance of IDS or firewall.

Chapter 3

Literature Survey Synopsis

Research papers on different algorithms for reduction of alert log data are analyzed. For different applications different algorithms are used. The different algorithms which are used are as follows:

3.1 Memory based learning approach:

Memory based approach is similar to signature based detection algorithm. If substantial set of examined data is accessible than memory based supervised learning approach can be effectualy used to upgrade the result of signature based Intrusion Detection System[1]. It is a kind of behavior-based approach which could remove false alerts or reduce the rate of false alarms and also repay the precision of IDS with the memory-based supervised learning algorithm. We use a alternative of Instance Based Learning (IBL) algorithm as a performance explorer of SNORT not only if a SNORT alarm is truly good notifying to the network administrator but also rise an alarm for IBLs own assessment, although, SNORT has not performed action.

IBL reserves a subgroup of actual occurences that are identified as "abnormal" or "normal" by using the training set database. Thus, it has a knowledge of the set of important attack signatures or actions. When a group of events occur in a network, IBL algorithm tries to find the most close occurence from its training dataset to current events and decides whether it is normal or not by following the signature values of the most similar training data[9].

The experimental result show that the false positive alert detection rate is 97.76%[3] and is not satisfactory. This may be due to deficient set of signature values we have in

SNORT. Maximum number of alarms are generateded by the same attack set. In that type of situations, a good alarm filter that gives valuable information could be useful for these systems[1].

3.2 Neuro-Fuzzy Approach:

Neuro-fuzzy approach is a hybrid approach to reduce false alert generation rate. The neuro-fuzzy approach was tested with different training datasets in DARPA 1999 network traffic database. The approach was analyzed and compared with Repeated Incremental Pruning to Produce Error Reduction(RIPPER) algorithm. The results show that the rate of false alert reduction of neuro-fuzzy approach is more than the RIPPER algorithm and it needs less background training datasets compared to these algorithms[2]. The flow diagram of Neuro-Fuzzy approach is shown in figure 3.1.

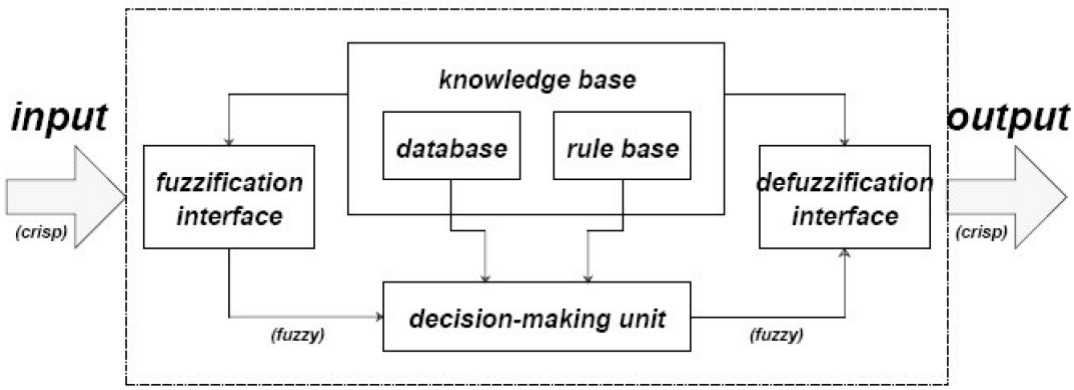


Figure 3.1: Neuro-Fuzzy Approach

To reduce the false positive alerts generation rate of an IDS, we require an approach, which is capable to deal with anonymity in network traffic to predict never seen and noisy data precisely. Moreover, the data provided for alerts through raw dataset and logs do not give enough details on the features of the connections built on the network[2]. A neural network can predict a function from raw data; however, it is not capable to exlicate the outcome in terms of natural language. A fuzzy rule base encloses this drawback, but fuzzy rules require prior training data which is not able to be obtained from raw data.

So, this hybrid neuro-fuzzy technique is used to create fuzzy rules that catagorize alerts as false or true positives using the training dataset. By the experiment results, it is observed that this approach of neuro-fuzzy system can notably reduce the rate of

false positive alerts by 90.92% using less traning data compared to RIPPER algorithm. However, the number the false negative alerts classified by this approach (5.07%) are greater than those algorithms classified by the RIPPER algorithm (0.02%). Thus, the neuro fuzzy approach looks too unpretentious to false negative alerts. The RIPPER algorithm is grouped with this approach for better alert classification so that it is able to capture both false negative and false positive alerts[2, 11, 5].

3.3 Outlier Detection Algorithm:

Outlier detection algorithm is used for reducing false alerts and detecting true alerts. This algorithm uses repeated attribute values extracted from historical alert values as false alerts, and then removes false alerts by the score counted based on these historical alert values[3]. This method contains two-way phase. Firstly, it filters all the alerts. Secondly, it learns the newcoming alerts and automatically adds it into the filtering mechanism. This method does not need any domain knowledge. It needs little human assistance, so it is more empirical than ongoing solutions. Here, The flow diagram of outlier detection algorithm is shown in figure 3.2.

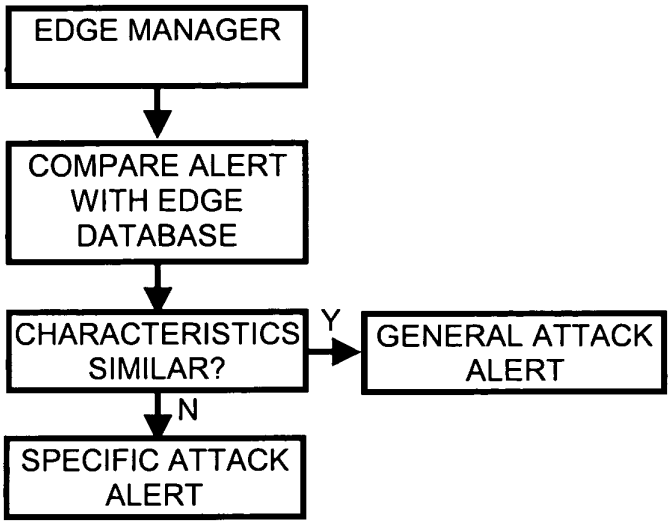


Figure 3.2: OutlierDetection Algorithm

In order to filter IDS alerts better, A special outlier detection algorithm is designed for this field, i.e. an improved frequent pattern-based outlier detection algorithm. It assigns an outlier to each alert VALUE, which indicates how abnormal the alert is. The score is calculated based on how many frequent attribute values the alert contains. Usually,

more the frequent patterns an alert has, higher its score is, and the more likely it is a false alert [3].

To filter real time alerts, a two-phase framework is designed. In the the first learning phase, it creates the training set of false positives alerts and calculates the threshold value of true alerts by using this training set. Then in the next the online filtering phase, it compares the outlier value of each new alerts with this threshold value to determine whether it is true alert or not. furthermore, the training dataset is automatically updated in given time period to keep its accuracy [6].

It is analyzed that when this outlier detection algorithm filters up to 86% of alerts, 100% of true alerts are still detected. And on real-world dataset, this model has greater false alerts reduction rate[3].

Chapter 4

Proposed Algorithm

To overcome the limitations associated with the techniques roposed in literature this framework is proposed. The limitations targeted here include:

1. The algorithms need a training dataset in which the historical alert signatures are stored. So, based on that, it can detect the true alerts.

2. Almost all the algorithms are used in offline mode. so when any newcoming alert is there, will not be detected as true alert or false alert on time. It reacts late.

3. The signature based algorithms (Statistical algorithm) cannot adapt in new situations; hence, they cannot detect the new types of alerts. This may be hazardous as sometimes it leads to data loss [4].

This approach is based on occurence of unusual attacks. It attempts to overcome some of these limitations. It is Hybrid approach which uses signatures and also detects the alerts by their behavior. The flow chart of the proposed algorithm is shown in Figure Figure 4.1. Raw data is collected from an Intrusion Detection System (IDS) for analysis purpose. The data is collected from the Firewall running at the institute. Data is available in chunk and needs manual analysis.

The proposed approach can be defined as follows:

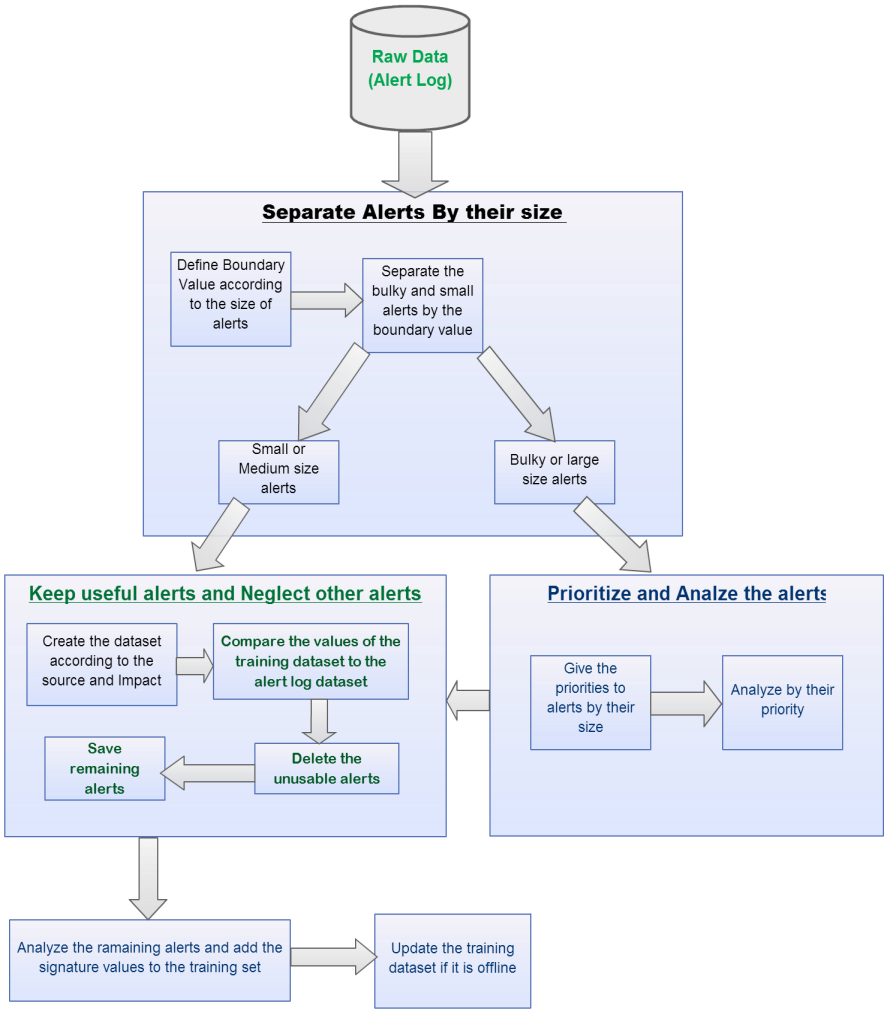


Figure 4.1: Proposed Algorithm

4.1 Separate Alerts By their size:

First of all we need to separate the bulky alerts and small size alerts. so for that we are separating them using following steps:

- First of all we assign the value to all the alerts by their size. than we define the boundary values.
- We separate the buky and small size alerts by this boundary value. if the value exceeds it is defined as bulky and if not than it is small size alerts. the boundary value is working as the separator.
- Now all the bulky alerts and small size alerts are stored in different datasets.

Algorithm 1 Alert Log Data Separation

INPUT: Alert log Dataset
OUTPUT: Separated Dataset

```
Input(l) ← Select the log file (Raw data)
Sheet1 ← Sheet No // Raw data Sheet No
Sheet2 ← New Sheet // Created a New Sheet
for Total j number of rows in Sheet 1 do
    x ← Cell value of Sheet 1
    words ← splited values of x by ″,″
    for Total n range length of words do
        Write words(n) in Sheet 2
        n ← n + 1
    end for
end for
```

4.2 Prioritize and Analyze the alerts:

For bulky alerts we need to give them priority according to their size to make the algorithm more effective. So for that we are following these steps:

- First we assign the numbers to the alerts according to their size. small size alerts in bulky alert database is prioritized first and the large size alert is prioritized by last.
- After giving the priority we analyze them by the priority value. so that we can analyze maximum number of alerts.

4.3 Keep useful alerts and neglect other alerts:

- Organize the available raw data data in proper format. Data is seggregated into various fields. For the same, Python language is used. The information available is filtered using the approach shown in Figure 3.1.
- The training dataset is created by using the historical signature data values. The training dataset is filled with the values according to the source and impact of that data.
- Now the filtered data is compared with the values of the training dataset, the unwanted data is removed from the filtered dataset and save the ramining useful data. This will reduce the huge amount of unwanted data.

Algorithm 2 Alert Prioritization

INPUT: Alert log Dataset
OUTPUT: Prioritized Alert log Dataset

Input(l) ← Select the log file (Raw data)
$Sheet1$ ← Sheet No // Raw data Sheet No
$Sheet2$ ← New Sheet // Created a New Sheet
$Sheet3$ ← New Sheet // Created a New Sheet
$limit \leftarrow 50$
for Total n number of rows in Sheet 1 **do**
    $x \leftarrow$ Row length of Sheet 1
    if $x(n) \leq limit$ **then**
        Write $x(n)$ in Sheet 2
    else
        Write $x(n)$ in Sheet 3
    end if
end forn=0
for Total j number of rows in Sheet 2 **do**
    Write $x(n)$ in sheet 2 of row $(j + 1)$
    $n \leftarrow n + 1$
    $j \leftarrow j + 1$
end for

- Now this remaining data is having too many blank rows. so, this blank rows are removed. so, the data size of this unfiltered log data is reduced by 80% in final log data.

Algorithm 3 Alert data Reduction

INPUT: Alert log Dataset
OUTPUT: Reduced Alert log Dataset

```
Input(l) ← Select the log file
Sheet1 ← Sheet No // Raw data Sheet No
Sheet2 ← New Sheet // Created a New Sheet
Rules ← Keywords // Key words of False alert data
for Total m number of rows in Sheet 1 do
    for Total n number of columns in Sheet 1 do
        p(m, n) ← cell value of Sheet 1
        Write p in Sheet 2
    end for
end for
for Total j number of rows in Sheet 1 do
    x(j) ← cell value
    if x(j) is Rules then
        Delete j row
    end if
end for
```

Algorithm 4 Alert data Merging

INPUT: Final Reduced Alert Log Dataset

```
Input(l) ← Select the log file
Sheet1 ← Sheet No // Raw data Sheet No
Sheet2 ← New Sheet // Created a New Sheet
s ← 0
s1 ← " "
for Total m number of rows in Sheet 1 do
    p(m, 0) ← cell value of Sheet 1
    if p ≠ s1 then
        s ← s + 1
        for Total d number of columns in Sheet 1 do
            x ← cell value of Sheet 1
            Write x in Sheet 2
        end for
    end if
end for
```

4.4 Analyze the remaining alerts and add the signature values to the training set:

- Now analyze the ramaining final data log and find that if the data is showing any major impact or not. based on that the signature values will be added in the training dataset.

Algorithm 5 Alert data Analysis

INPUT: Alert log Dataset
OUTPUT: Signatures of remaining alerts

```
Input(l) ← Select the log file
Sheet1 ← Sheet No // Raw data Sheet No
Sheet2 ← New Sheet // Created a New Sheet
for Total m number of rows in Sheet 1 do
    for Total n number of columns in Sheet 1 do
        p(m, n) ← cell value of Sheet 1
        Write p in Sheet 2
        x ← p(m, n)
    end for
end for
for Total j number of rows in Sheet 1 do
    x(j) ← cell value
    if p(j) is x then
        Delete j row
    end if
    Write x in Sheet 2
end for
```

4.5 Update the training dataset if it is offline:

- Most of the organisations use the offline training dataset. so for new alerts it will be useless or time consuming and so the training dataset is updated in prescribed time period.

Chapter 5

Implementation

5.1 Raw Data

The raw data is collected from the Firewall of Nirma University (Watchguard Firewall). The raw data contais too many different alerts based on different actions by users. As shown in 5.1, Each and every alerts include all the details like the reason for alert, the protocol used, the source id, the destination id, the priority of the alert, from which section this alert is generated, the unique id of that user, the request id allowed or denied, the source and destination port no etc. These all details are stored in one single cell. The date and time is also included in these alerts.

Now we are analysing these alerts. Most of the alerts are false and useless alerts. For network administrators it is difficult to find actual True alerts from these alerts. Here, This alert log database is of one day log of alerts. The size of this alert is around 547 MB as shown in 5.2. So, our first step is to saparate the data so that we can analyze this data.

A	B	C	D	E	F
ster	sn	log	xml_log	raw_id	update_time
	A0BB0034	tr	ProxyDNSReq, DNS request, pri=6, disp=Allow, policy=DNS-00, protocol=dns/udp, src_ip=10.1.19.28, src_port=60315, dst_ip=4.2.2.2, dst_port=53,	2355004	3/4/2015 13:55
	A0BB0034	tr	ProxyDNSReq, DNS request, pri=6, disp=Allow, policy=DNS-00, protocol=dns/udp, src_ip=10.1.19.28, src_port=59256, dst_ip=4.2.2.2, dst_port=53,	2354996	3/4/2015 13:55
	A0BB0034	tr	ProxyHTTPReq, HTTP request, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, src_port=63155, d	2354994	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyAllow: HTTP Body Content Type match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=	2354986	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyStrip: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, sr	2354976	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyStrip: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, sr	2354973	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyStrip: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, sr	2354970	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyStrip: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, sr	2354967	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyStrip: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, sr	2354964	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyAllow: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, s	2354962	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyAllow: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, s	2354960	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyAllow: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, s	2354958	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyAllow: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, s	2354956	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyStrip: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, sr	2354953	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyStrip: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, sr	2354950	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyAllow: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, s	2354946	3/4/2015 13:55
	A0BB0034	tr	ProxyDNSReq, DNS request, pri=6, disp=Allow, policy=DNS-00, protocol=dns/udp, src_ip=10.1.19.28, src_port=59148, dst_ip=4.2.2.2, dst_port=53,	2354944	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyAllow: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, s	2354943	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyStrip: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, sr	2354940	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyAllow: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, s	2354938	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyAllow: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, s	2354936	3/4/2015 13:55
	A0BB0034	tr	ProxyMatch, ProxyAllow: HTTP header match, pri=6, disp=Allow, policy=HTTP-proxy-Lib-database-day-00, protocol=http/tcp, src_ip=10.1.17.53, s	2354934	3/4/2015 13:55

Figure 5.1: Raw data

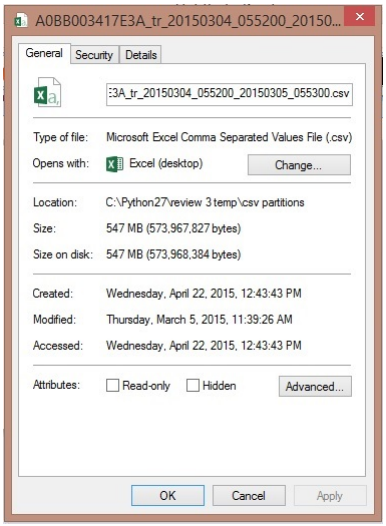
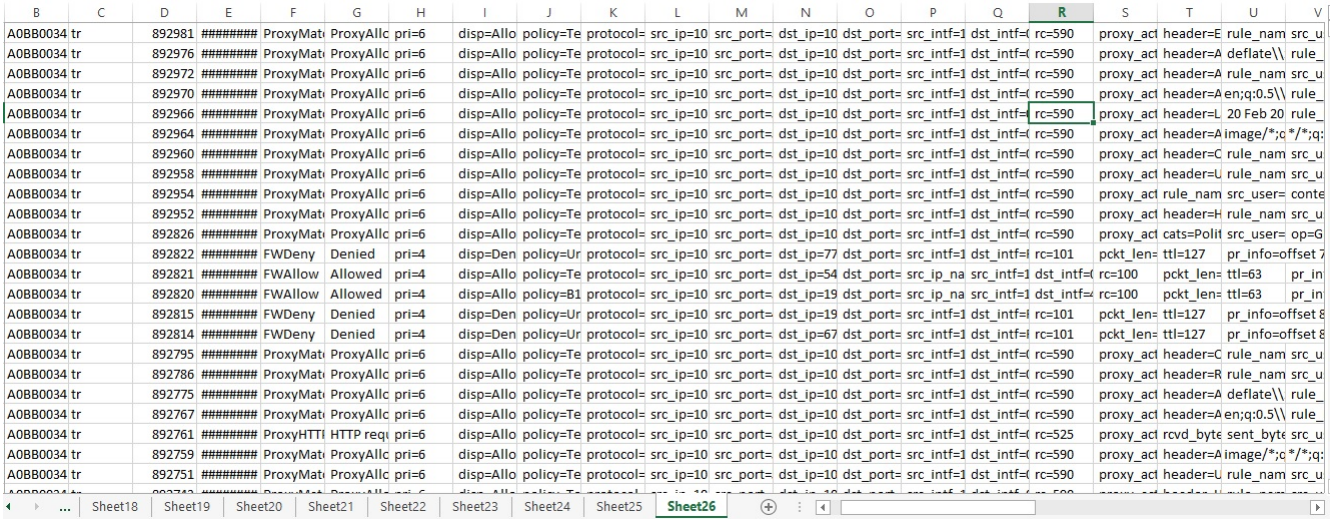


Figure 5.2: Size of the Raw data

5.2 Data Separation

Now we are separating all the fields into different cells so that we can analyze all the alerts and we can find the difference between them. As shown in 5.3, The log data is now separated in different columns. Now we can easily analyze the data and find the signatures of false alerts and using that signature we can remove that data.

Now the data is separated and is prioritized according to the date and time. Now, We are analyzing the types of alerts using this data. The sepaated data is stored in different excel file.



B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
A0BB0034	tr	892981	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	header=E	rule_nam	src_u
A0BB0034	tr	892976	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	header=A	deflate\\	rule_
A0BB0034	tr	892972	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	header=A	rule_nam	src_u
A0BB0034	tr	892970	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	header=A	en;q=0.5\\	rule_
A0BB0034	tr	892966	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	header=L	20 Feb 20	rule_
A0BB0034	tr	892964	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	header=A	image/*;q	*/*;q=
A0BB0034	tr	892960	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	header=C	rule_nam	src_u
A0BB0034	tr	892958	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	header=U	rule_nam	src_u
A0BB0034	tr	892954	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	rule_nam	src_user=	conte
A0BB0034	tr	892952	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	header=H	rule_nam	src_u
A0BB0034	tr	892826	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	cats=Polit	src_user=	op=G
A0BB0034	tr	892822	########	FWDeny	Denied	pri=4	disp=Den	policy=Ur	protocol=	src_ip=10	src_port=	dst_ip=77	dst_port=	src_intf=1	dst_intf=I	rc=101	pckt_len=	ttl=127	pr_info=offset 7	
A0BB0034	tr	892821	########	FWAllow	Allowed	pri=4	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=54	dst_port=	src_ip_na	src_intf=1	dst_intf=(	rc=100	pckt_len=	ttl=63	pr_in
A0BB0034	tr	892820	########	FWAllow	Allowed	pri=4	disp=Allo	policy=B1	protocol=	src_ip=10	src_port=	dst_ip=19	dst_port=	src_ip_na	src_intf=1	dst_intf=/	rc=100	pckt_len=	ttl=63	pr_in
A0BB0034	tr	892815	########	FWDeny	Denied	pri=4	disp=Den	policy=Ur	protocol=	src_ip=10	src_port=	dst_ip=19	dst_port=	src_intf=1	dst_intf=I	rc=101	pckt_len=	ttl=127	pr_info=offset 8	
A0BB0034	tr	892814	########	FWDeny	Denied	pri=4	disp=Den	policy=Ur	protocol=	src_ip=10	src_port=	dst_ip=67	dst_port=	src_intf=1	dst_intf=I	rc=101	pckt_len=	ttl=127	pr_info=offset 8	
A0BB0034	tr	892795	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	header=C	rule_nam	src_u
A0BB0034	tr	892786	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	header=R	rule_nam	src_u
A0BB0034	tr	892775	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	header=A	deflate\\	rule_
A0BB0034	tr	892767	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	header=A	en;q=0.5\\	rule_
A0BB0034	tr	892761	########	ProxyHTTI	HTTP requ	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=525	proxy_act	rcvd_byte	sent_byte	src_u
A0BB0034	tr	892759	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	header=A	image/*;q	*/*;q=
A0BB0034	tr	892751	########	ProxyMat	ProxyAllc	pri=6	disp=Allo	policy=Te	protocol=	src_ip=10	src_port=	dst_ip=10	dst_port=	src_intf=1	dst_intf=(	rc=590	proxy_act	header=U	rule_nam	src_u

Sheet18 | Sheet19 | Sheet20 | Sheet21 | Sheet22 | Sheet23 | Sheet24 | Sheet25 | Sheet26

Figure 5.3: Data separation

5.3 Data Reduction

By analysing the data, we found that the following signatures of alerts are of false alerts or useless alerts.

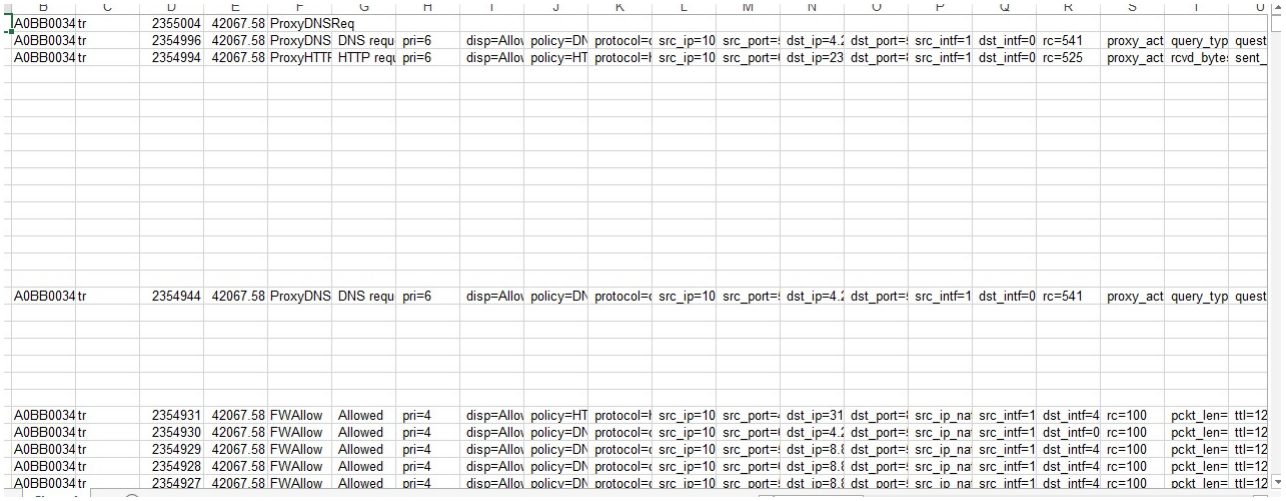
1. **ProxyMatch :** When user requests on the restricted domain the request is denied and this kind of alert is generated. Here When the user expressions or IP of this domain is alresdy added in Firewall rules, this kind of alert is generated [22].

2. **FWAllow :** When the network packet is allowed than this kind of alert is generated [22].

3. **src_intf=1-Trusted :** This indicates that the packet is generated from the trusted source [22].

4. **pri = 6 :** This shows the normal allowed traffic. basically, pri shows the priority of the log. for pri = 1 it is showing critical mode, for pri = 6 it is showingwarning mode and for pri = 6 it is showing normal traffic mode [22].

Now using this all signatures, We have removed the useless data. Figure 5.4 shows the alert database after reduction.



A0BB0034	tr	2355004	42067.58	ProxyDNSReq															
A0BB0034	tr	2354996	42067.58	ProxyDNS	DNS requ	pri=6	disp=Allov	policy=DN	protocol=(	src_ip=10	src_port=(	dst_ip=4.2	dst_port=5	src_intf=1	dst_intf=0	rc=541	proxy_act	query_typ	quest
A0BB0034	tr	2354994	42067.58	ProxyHTTF	HTTP requ	pri=6	disp=Allov	policy=HT	protocol=I	src_ip=10	src_port=(	dst_ip=23	dst_port=8	src_intf=1	dst_intf=0	rc=525	proxy_act	rcvd_byte:	sent_
A0BB0034	tr	2354944	42067.58	ProxyDNS	DNS requ	pri=6	disp=Allov	policy=DN	protocol=(	src_ip=10	src_port=(	dst_ip=4.2	dst_port=5	src_intf=1	dst_intf=0	rc=541	proxy_act	query_typ	quest
A0BB0034	tr	2354931	42067.58	FWAllow	Allowed	pri=4	disp=Allov	policy=HT	protocol=I	src_ip=10	src_port=(	dst_ip=31	dst_port=8	src_ip_na	src_intf=1	dst_intf=4	rc=100	pckt_len=	ttl=12
A0BB0034	tr	2354930	42067.58	FWAllow	Allowed	pri=4	disp=Allov	policy=DN	protocol=(	src_ip=10	src_port=(	dst_ip=4.2	dst_port=5	src_ip_na	src_intf=1	dst_intf=0	rc=100	pckt_len=	ttl=12
A0BB0034	tr	2354929	42067.58	FWAllow	Allowed	pri=4	disp=Allov	policy=DN	protocol=(	src_ip=10	src_port=(	dst_ip=8.8	dst_port=5	src_ip_na	src_intf=1	dst_intf=4	rc=100	pckt_len=	ttl=12
A0BB0034	tr	2354928	42067.58	FWAllow	Allowed	pri=4	disp=Allov	policy=DN	protocol=(	src_ip=10	src_port=(	dst_ip=8.8	dst_port=5	src_ip_na	src_intf=1	dst_intf=4	rc=100	pckt_len=	ttl=12
A0BB0034	tr	2354927	42067.58	FWAllow	Allowed	pri=4	disp=Allov	policy=DN	protocol=(	src_ip=10	src_port=(	dst_ip=8.8	dst_port=5	src_ip_na	src_intf=1	dst_intf=4	rc=100	pckt_len=	ttl=12

Figure 5.4: Filtered data-1

5.4 Final Reduced alert data

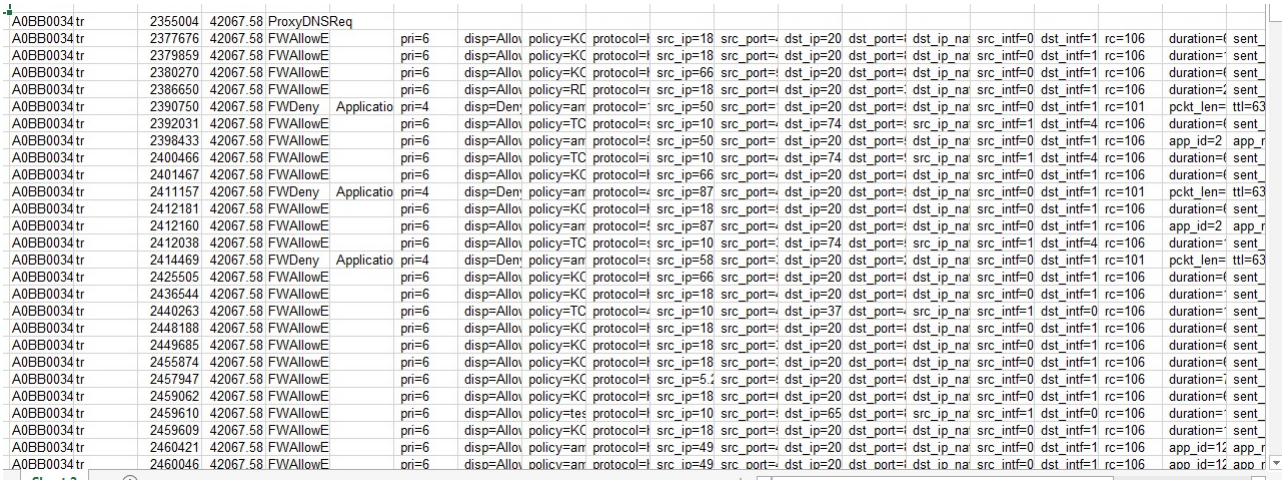
In the Figure 5.4 we can see that the useless data is removed. The removed data is shown as blank row. Now we are removing that blank rows and merging all the data. As shown in Figure 5.5 the data is merged. Now these remaining alerts are having following warning keywords.

1. **pri=1 :** The priority of the log. The priority is used only for Net IQ reporting. If it is set to 1 than it shows critical mode.

2. **pri=4 :** If it is set to 4 than it shows warning mode.

3. **policy=Unhandled-Internal-Packet-00 :** These alarms are caused by events associated with each policy.

he size of this alert is around 11.8 as shown in 5.6.



A0BB0034 tr	2355004	42067.58	ProxyDNSReq															
A0BB0034 tr	2377676	42067.58	FWAllowE		pri=6	disp=Allo	policy=KC	protocol=I	src_ip=18	src_port=4	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=106	duration=(	sent_
A0BB0034 tr	2379859	42067.58	FWAllowE		pri=6	disp=Allo	policy=KC	protocol=I	src_ip=18	src_port=4	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=106	duration=(	sent_
A0BB0034 tr	2380270	42067.58	FWAllowE		pri=6	disp=Allo	policy=KC	protocol=I	src_ip=66	src_port=4	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=106	duration=(	sent_
A0BB0034 tr	2386650	42067.58	FWAllowE		pri=6	disp=Allo	policy=RD	protocol=(	src_ip=18	src_port=4	dst_ip=20	dst_port=3	dst_ip_na	src_intf=0	dst_intf=1	rc=106	duration=(	sent_
A0BB0034 tr	2390750	42067.58	FWDeny	Applicatio	pri=4	disp=Den	policy=an	protocol='	src_ip=50	src_port=	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=101	pckt_len=	ttl=63
A0BB0034 tr	2392031	42067.58	FWAllowE		pri=6	disp=Allo	policy=TC	protocol=(	src_ip=10	src_port=4	dst_ip=74	dst_port=5	src_ip_na	src_intf=1	dst_intf=4	rc=106	duration=(	sent_
A0BB0034 tr	2398433	42067.58	FWAllowE		pri=6	disp=Allo	policy=an	protocol=(	src_ip=50	src_port=	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=106	app_id=2	app_r
A0BB0034 tr	2400466	42067.58	FWAllowE		pri=6	disp=Allo	policy=TC	protocol=i	src_ip=10	src_port=4	dst_ip=74	dst_port=5	src_ip_na	src_intf=1	dst_intf=4	rc=106	duration=(	sent_
A0BB0034 tr	2401467	42067.58	FWAllowE		pri=6	disp=Allo	policy=KC	protocol=I	src_ip=66	src_port=4	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=106	duration=(	sent_
A0BB0034 tr	2411157	42067.58	FWDeny	Applicatio	pri=4	disp=Den	policy=an	protocol=4	src_ip=87	src_port=4	dst_ip=20	dst_port=5	dst_ip_na	src_intf=0	dst_intf=1	rc=101	pckt_len=	ttl=63
A0BB0034 tr	2412181	42067.58	FWAllowE		pri=6	disp=Allo	policy=KC	protocol=I	src_ip=18	src_port=4	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=106	duration=(	sent_
A0BB0034 tr	2412160	42067.58	FWAllowE		pri=6	disp=Allo	policy=an	protocol=(	src_ip=87	src_port=4	dst_ip=20	dst_port=5	dst_ip_na	src_intf=0	dst_intf=1	rc=106	app_id=2	app_r
A0BB0034 tr	2412038	42067.58	FWAllowE		pri=6	disp=Allo	policy=TC	protocol=(	src_ip=10	src_port=4	dst_ip=74	dst_port=5	src_ip_na	src_intf=1	dst_intf=4	rc=106	duration='	sent_
A0BB0034 tr	2414469	42067.58	FWDeny	Applicatio	pri=4	disp=Den	policy=an	protocol=(	src_ip=58	src_port=3	dst_ip=20	dst_port=2	dst_ip_na	src_intf=0	dst_intf=1	rc=101	pckt_len=	ttl=63
A0BB0034 tr	2425505	42067.58	FWAllowE		pri=6	disp=Allo	policy=KC	protocol=I	src_ip=66	src_port=4	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=106	duration=(	sent_
A0BB0034 tr	2436544	42067.58	FWAllowE		pri=6	disp=Allo	policy=KC	protocol=I	src_ip=18	src_port=4	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=106	duration='	sent_
A0BB0034 tr	2440263	42067.58	FWAllowE		pri=6	disp=Allo	policy=TC	protocol=4	src_ip=10	src_port=4	dst_ip=37	dst_port=4	src_ip_na	src_intf=1	dst_intf=0	rc=106	duration='	sent_
A0BB0034 tr	2448188	42067.58	FWAllowE		pri=6	disp=Allo	policy=KC	protocol=I	src_ip=18	src_port=4	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=106	duration=(	sent_
A0BB0034 tr	2449685	42067.58	FWAllowE		pri=6	disp=Allo	policy=KC	protocol=I	src_ip=18	src_port=3	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=106	duration=(	sent_
A0BB0034 tr	2455874	42067.58	FWAllowE		pri=6	disp=Allo	policy=KC	protocol=I	src_ip=18	src_port=4	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=106	duration=(	sent_
A0BB0034 tr	2457947	42067.58	FWAllowE		pri=6	disp=Allo	policy=KC	protocol=I	src_ip=5.2	src_port=4	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=106	duration=3	sent_
A0BB0034 tr	2459062	42067.58	FWAllowE		pri=6	disp=Allo	policy=KC	protocol=I	src_ip=18	src_port=4	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=106	duration=(	sent_
A0BB0034 tr	2459610	42067.58	FWAllowE		pri=6	disp=Allo	policy=tes	protocol=I	src_ip=10	src_port=4	dst_ip=65	dst_port=4	src_ip_na	src_intf=1	dst_intf=0	rc=106	duration='	sent_
A0BB0034 tr	2459609	42067.58	FWAllowE		pri=6	disp=Allo	policy=KC	protocol=I	src_ip=18	src_port=4	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=106	duration='	sent_
A0BB0034 tr	2460421	42067.58	FWAllowE		pri=6	disp=Allo	policy=an	protocol=I	src_ip=49	src_port=4	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=106	app_id=12	app_r
A0BB0034 tr	2460046	42067.58	FWAllowE		pri=6	disp=Allo	policy=an	protocol=I	src_ip=49	src_port=4	dst_ip=20	dst_port=4	dst_ip_na	src_intf=0	dst_intf=1	rc=106	app_id=12	app_r

Figure 5.5: Filtered data-2

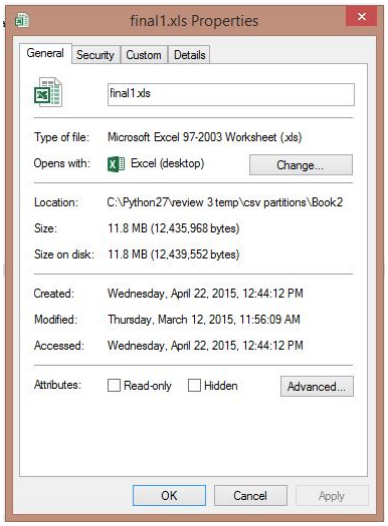


Figure 5.6: Filtered data size

5.5 Difference between original dataset and final dataset

Here, The difference between the raw data and final reduced data is computed. We calculated the total number of rows of original data which is aroung 1000K and we calculated the rows of final reduced data whch is nearby 32K. We calculated the percentage of reduction using this data as shown in Figure 5.7.

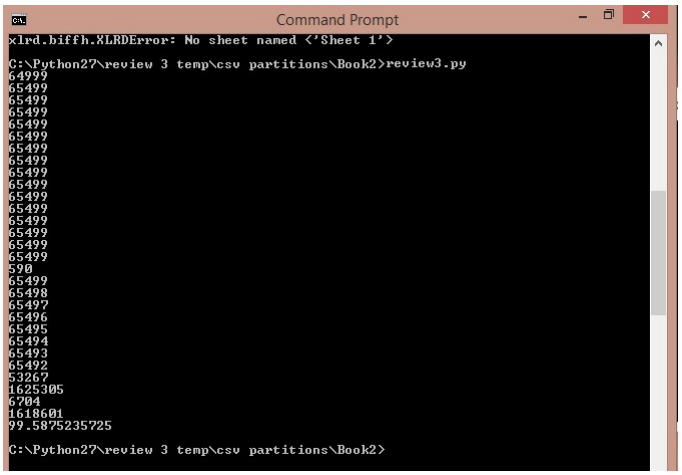


Figure 5.7: Difference between original dataset and final dataset

Chapter 6

Technical requirements and Feasibility

- **Tools:-**

  1. To reduce the size of alert log the algorithm is developed in **Python2.7**
  2. To make the training dataset or to work on the false alerts we need the alert log from any **Firewall/IDS**
  3. To parse the alerts and remove the unwanted alerts we need the extensions that are **xlsxwriter,xlutils and win32com.client**
  4. Python is open-source and with many extensions to develope the rules of alert reduction.
  5. Python is platform independent and OOP language so it can run on any platform that are Windows/Linum.

Chapter 7

Conclusion

After studying different alert log reduction algorithms, it is analyzed that for different applications different algorithms are used. Outlier detection algorithm and neuro-fuzzy algorithm are giving high performance compared to others as the false alerts are reduced by 80-90% and true alerts detection rate is 100%. So we can use these algorithms in different applications as it's accuracy is greater. The proposed algorithm covers most of the limitations of these algorithms and the theoritical result values are 99.5% of Alert reduction rate and 100% of True alert detection rate. So, This proposed algorithm is also efficient than these algorithm and it also can be used where these algorithm are not that much sufficient.

References

[1] Weon, Ill-Young, et al. "A memory-based learning approach to reduce false alarms in intrusion detection." Advanced Communication Technology, 2005, ICACT 2005. The 7th International Conference on. Vol. 1. IEEE, 2008.

[2] Alshammari, Riyad, et al. "Using neuro-fuzzy approach to reduce false positive alerts." Communication Networks and Services Research, 2007. CNSR'07. Fifth Annual Conference on. IEEE, 2007.

[3] Xiao, Fu, and Xie Li. "Using outlier detection to reduce false positives in intrusion detection." Network and Parallel Computing, 2008. NPC 2008. IFIP International Conference on. IEEE, 2008.

[4] Khanchi, Sara, and Fazlollah Adibnia. "False Alert Reduction on Network-Based Intrusion Detection Systems by Means of Feature Frequencies." Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT'09. International Conference on. IEEE, 2009.

[5] Gaonjur, Pravesh, et al. "Using Neuro-Fuzzy Techniques to reduce false alerts in IDS." Networks, 2008. ICON 2008. 16th IEEE International Conference on. IEEE, 2008.

[6] Tosun, Ayse, and Ayse Bener. "Reducing false alarms in software defect prediction by decision threshold optimization." Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement. IEEE Computer Society, 2009.

[7] Nehinbe, Joshua Ojo. "Automated method for reducing false positives." Intelligent Systems, Modelling and Simulation (ISMS), 2010 International Conference on. IEEE, 2010.

[8] Kenaza, Tayeb, and Abdelhalim Zaidi. "Clustering approach for false alerts reducing in behavioral based intrusion detection systems." Machine and Web Intelligence (ICMWI), 2010 International Conference on. IEEE, 2010.

[9] Kalamatianos, Theodoros, Kostas Kontogiannis, and Peter Matthews. "Domain independent event analysis for log data reduction." Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual. IEEE, 2012.

[10] Eslahi, Meisam, H. Hashim, and N. M. Tahir. "An efficient false alarm reduction approach in HTTP-based botnet detection." Computers & Informatics (ISCI), 2013 IEEE Symposium on. IEEE, 2013.

[11] Pietraszek, Tadeusz. "Using adaptive alert classification to reduce false positives in intrusion detection." Recent Advances in Intrusion Detection. Springer Berlin Heidelberg, 2004.

[12] Bakar, Najwa Abu, Bahari Belaton, and Azman Samsudin. "False positives reduction via intrusion alert quality framework." Networks, 2005. Jointly held with the 2005 IEEE 7th Malaysia International Conference on Communication., 2005 13th IEEE International Conference on. Vol. 1. IEEE, 2005.

[13] Bakar, Najwa Abu, Bahari Belaton, and Azman Samsudin. "False positives reduction via intrusion alert quality framework." Networks, 2005. Jointly held with the 2005 IEEE 7th Malaysia International Conference on Communication., 2005 13th IEEE International Conference on. Vol. 1. IEEE, 2005.

[14] Srinivasan, N., and V. Vaidehi. "Reduction of false alarm rate in detecting network anomaly using mahalanobis distance and similarity measure." Signal Processing, Communications and Networking, 2007. ICSCN'07. International Conference on. IEEE, 2007.

[15] Hooper, Emmanuel. "Intelligent detection and response strategies for complex attacks." Aerospace and Electronic Systems Magazine, IEEE 22.11 (2007): 3-12.

[16] Alsubhi, Khalid, Ehab Al-Shaer, and Raouf Boutaba. "Alert prioritization in intrusion detection systems." Network Operations and Management Symposium, 2008. NOMS 2008. IEEE. IEEE, 2008.

[17] Xu, Ming, Ting Wu, and Jingfan Tang. "An IDS alert fusion approach based on happened before relation." Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on. IEEE, 2008.

[18] Barapatre, P. R. A. C. H. I., et al. "Training MLP neural network to reduce false alerts in IDS." Computing, Communication and Networking, 2008. ICCCn 2008. International Conference on. IEEE, 2008.

[19] Khanchi, Sara, and Fazlollah Adibnia. "False Alert Reduction on Network-Based Intrusion Detection Systems by Means of Feature Frequencies." Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT'09. International Conference on. IEEE, 2009.

[20] Taihua, Wang, and Guo Fan. "Associating IDS alerts by an improved apriori algorithm." Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium on. IEEE, 2010.

[21] Njogu, Humphrey Waita, and Luo Jiawei. "Using alert cluster to reduce IDS alerts." Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on. Vol. 5. IEEE, 2010.

[22] http://www.watchguard.com/support/fireware_howto/HowTo_ReadFirewareLogs.pdf