# Anomaly Detection Technique for Honeynet Data Analysis

BY

SYED NAEEM FIRDOUS

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

**KING FAHD UNIVERSITY OF PETROLEUM & MINERALS**

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

# MASTER OF SCIENCE

**In**

COMPUTER NETWORKS

JANUARY 2011

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by Syed Naeem Firdous under the direction of his thesis advisor and approved by his thesis committee, has been presented to and accepted by Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** in **COMPUTER NETWORKS**.

Thesis Committee

Dr. Mohammed H. Sqalli (Advisor)

Dr. Khaled H. Salah (Co-Advisor)

Dr. Marwan Abu-Amara (Member)

Dr. Zubair Ahmed Baig (Member)

Dr. Farag Azzedin (Member)

Dr. Basem AL-Madani (Department Chairman)

Dr. Salam Adel Zummo (Dean of Graduate Studies)

16/3/11
Date

*Dedicated*

*to*

*my beloved father and mother*

# Acknowledgements

Verily all praise is due to Allah (s.w.t). We praise Him; we seek His help and forgiveness. Peace and blessings be upon Prophet Muhammad (peace and blessings of Allah be upon him), the last of Allah's Messengers and Prophets, and upon his family, his Companions and all those who follow in his footsteps until the end of time. I would like to express my deepest gratitude to my parents and all my family members for their love, prayers, constant support and encouragement.

I acknowledge the support and facilities provided by King Fahd University of Petroleum & Minerals (KFUPM). I would like to express my sincere gratitude to my thesis advisor Dr. Mohammed Houssaini Sqalli for his guidance, support and invaluable advice through-out this thesis. I also thank my committee members Dr. Khaled Salah (co-advisor), Dr. Marwan Abu-Amara, Dr. Zubair Baig and Dr. Farag Azzedin for their feedback, support and advice given during this thesis work.

I am also thankful the chairman of Computer Engineering Department, Dr. Basem Al-Madani for their cooperation and providing department facilities.

I also thank all my fellow graduate students and all my friends who made this an enjoyable experience.

# Table of Contents

# Table of Figures

# List of Tables

# THESIS ABSTRACT

**Name**: Syed Naeem Firdous

**Title**: Anomaly Detection Technique for Honeynet Data Analysis

**Major Field**: Computer Networks

**Date of Degree**: Jan-2011

*A Honeynet is a network designed by the Honeynet Project organization to gather information on security threats. Honeynet is a controlled network of vulnerable computers (honeypots) and is being used by organizations to proactively improve their network security as it is designed to identify malicious traffic and unauthorized activities on the network. Honeynet captures a substantial amount of data and logs for analysis in order to identify malicious activities. The analysis of this large amount of data is a challenging task. The main aim of this work is to use an anomaly detection technique to detect new and unknown attacks using data collected by Honeynets. In this work we propose an anomaly detection technique that will efficiently analyze the Honeynet data and will help in learning the behavior and techniques used by the hackers. In this work, we have used feature-based and volume-based schemes to identify anomalies in Honeynet traffic. A detailed analysis of various traffic features were carried out and the most appropriate features for Honeynet traffic were selected. The anomaly detection was done by using threshold values of entropy distributions for feature-based parameters and volume changes for volume-based parameters. The behavior of various anomalies was defined using the selected features and their respective threshold values. In this work, we show that our proposed technique which utilizes both feature-based and volume-based parameters is effective in detecting most types of anomalies seen in Honeynets.*

# ملخص الرسالة

**الاســـــــــــم**: سيد نعيم فردوس

**عنوان الرسالة**: استخدام تقنية الكشف عن السلوك الشاذ بهدف تحليل بيانات شبكات العسل

**التخصـــــص**: شبكات الحاسب الآلي

**تاريخ التخــرج**: محرم ١٤٣٢

صممت شبكات العسل من قبل منظمة مشروع شبكات العسل العالمية بهدف جمع المعلومات عن التهديدات الأمنية داخل شبكات الحاسب الآلي. شبكة العسل هي عبارة عن شبكة من أجهزة الحاسب الآلي مراقبة ومعرضة للهجوم، ويتم استخدامها من قبل الهيئات والمنظمات والشركات بهدف استباق حدوث هجمات وتحسين أمن الشبكات. كما أنها مصممة لتحديد حركة المرور الضارة والأنشطة غير المصرح به على الشبكة. تلتقط شبكات العسل كمية كبيرة من البيانات والسجلات لتحليلها من أجل تحديد الأنشطة الخبيثة. يعتبر تحليل هذا الكم الكبير من البيانات مهمة صعبة. والهدف الرئيسي من هذا العمل هو استخدام تقنية الكشف عن السلوك الشاذ بهدف الكشف عن وقوع هجمات جديدة وغير معروفة باستخدام البيانات التي تم جمعها عن طريق شبكات العسل. نقترح في هذا العمل استخدام تقنية الكشف عن السلوك الشاذ والتي سيتم عن طريقها تحليل البيانات بكفاءة مما سيساعد على تعلم السلوك والأساليب المستخدمة من قبل المتسللين لشبكات الحاسب الآلي. في هذا العمل استخدمنا تقنية تعتمد على طرق تستند إلى استخدام ميزات وحجم البيانات من أجل تحديد الحالات الشاذة في حركة شبكات العسل. وقد أجري تحليل تفصيلي عن الميزات المختلفة المرتبطة بحركة المرور وتم اختيار الميزات الأكثر ملاءمة لحركة شبكات العسل. وقد تم الكشف عن السلوكات الشاذة باستخدام قيم عتبة توزيع الانتروبي بالنسبة للمعلمات المستندة إلى ميزات البيانات واستخدام التغييرات في الحجم بالنسبة للمعلمات المستندة إلى حجم البيانات. وقد تم تعريف السلوكات الشاذة المختلفة باستخدام الميزات المحددة وقيمها. تبين لنا من خلال هذا العمل أن أسلوبنا المقترح الذي يستخدم كلا من معلمات الميزات وحجم البيانات يعتبر أسلوبا فعالا للكشف عن معظم أنواع السلوكات الشاذة المتواجدة داخل شبكات العسل.

# *Chapter 1*

# Introduction

Computer network security is a major area of concern for different people from normal home users to businesses trying to protect their resources from unauthorized access. The moment a computer is connected to the Internet, the computer is physically connected to millions of computers in the network. There is constant threat from malicious users who are trying to disrupt normal operations or trying to steal sensitive or proprietary information. Network security is a prominent feature of the network ensuring accountability, confidentiality, integrity, and above all protection against many external and internal threats such as hacking, denial of service attacks, worms, Trojans, etc. Computer network security is generally taken as providing protection at the boundaries of an organization, keeping the bad guys (e.g. black hat hackers, script kiddies, etc.) out.

## 1.1 Honeynet

The Honeynet is a network designed to gather information on security threats which can be used by the organizations to proactively improve their network security. A Honeynet can be used to assist system administrators in identifying malicious traffic in the

enterprise network. By its very nature, a Honeynet has no production value and should not be generating or receiving any traffic. Any traffic to or from the Honeynet is suspicious in nature. The key requirements to successfully implement a Honeynet, is data control, data capture, and data analysis[1].

The Honeynet Project is an international volunteer organization dedicated to computer security research. It was founded in 1999 and holds non-profit status to provide a global research perspective, and the organization is strongly committed to the ideals of the Open Source movement. The goal of the Honeynet Project is to learn about the tools, tactics, and motives involved in computer network attacks, which is primarily carried out through the use of honeypots and honeynets. A series of honeypots and Honeynet related technologies have been developed to help support the Honeynet Project's research goals, all of which are freely available for download from the Honeynet website[2]. The members of the Honeynet project regularly generate research publications on new threats and technology improvements and are released in the form of Know Your Enemy (KYE) white papers, which are available from the Honeynet website[2].

A honeypot has been defined as a security resource whose value lies in being probed, attacked, or compromised[2]. There are two types of honeypots[3]: High Interaction and Low Interaction. High-interaction honeypots provide real systems, applications, and services for attackers to interact with. The advantages of high-interaction honeypots are that we can capture extensive amounts of information by giving attackers real systems to interact with. It enables us to learn the full extent of their behavior, everything from new root-kits to international IRC sessions. Honeynet, Sebek, and CaptureHPC are some of the examples of high-interaction honeypots[2]. Low-interaction honeypots provide

emulated services and they are easy to install and deploy. These types of honeypots capture limited information about the hackers and they are generally useful to understand a specific activity of the hacker. Honeyd, Nepenthes, and Google Hack are some of the examples of low-interaction honeypot.

Figure 1-1 shows a GenII Honeynet architecture, which is a highly controlled network used to contain and analyze attacks.

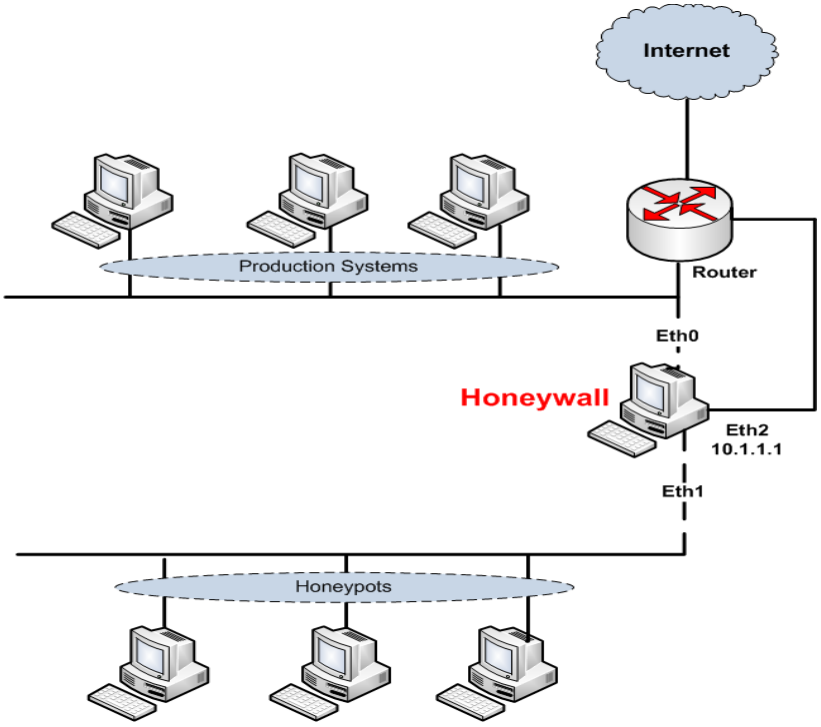


*Figure 1-1 : Honeynet Architecture*

Gen II and Gen III honeynets have 3 layers of data capture. These 3 layers are:

- Firewall: Honeywall has built in firewall which is used for data control and it also creates logs that are useful for analyzing various activities on the honeypot.
- IDS: Honeywall uses Snort IDS system for alert generation and to identify attacks. Snort creates its log files and also captures the packets passing through

the honeywall. Honeywall also uses Hflow daemon which coalesces data from snort and other tools into a database.

- Honeypot Data Capture: The third of data capture is at the honeypot itself which runs tools like Sebek (kernel module) which collects keystroke information at the honeypot.

The honeywall collects data such as Iptable logs, keystoke logs, file and I/O data, passive OS fingerprints, snort alerts, netflow data, and full binary packet dumps[4]. In the Honeynet, data coalescing is done by using a backend daemon called Hflow2. It coalesces data from snort, p0f, and sebek into a relational database (i.e., MySql). Hflow2 is placed below the helper tools Snort and POf as shown in Hflow2 architecture (refer to Figure 1-2). The main responsibilities of Hflow2 are to[5]:

- Synchronizes the helper tools
- Generates and store a composite view in the database and in a PCAP file

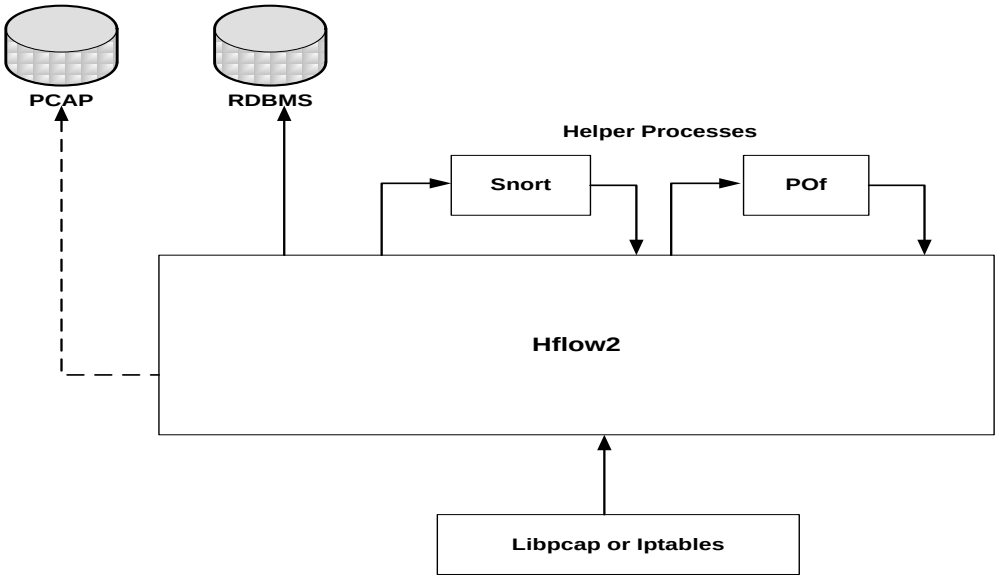


*Figure 1-2 : Honeynet Hflow2 Architecture[5]*

Currently, a Honeynet gathers a lot of network data and this sometimes makes it difficult to analyze this huge data set. However, a honeynet's real potential will not be realized until organizations can effectively deploy multiple honeynets and correlate the information they collect. The main objective of this thesis is to propose a data analysis technique of honeynet data to identify anomalies. This would help the administrators to better use the Honeynet data to understand the vulnerabilities in the system and take necessary actions to protect their systems from malicious activities.

## 1.2 Anomaly Detection

Anomaly Detection refers to a technique of detecting patterns that are different from the normal behavior. Anomaly detection helps to identify new or unknown patterns in any data set. The abnormal patterns with in any data set are referred to as anomalies, outliers, exceptions, peculiarities, etc.[6]. Figure 1-3 shows the regions which are labeled as normal or outliers.

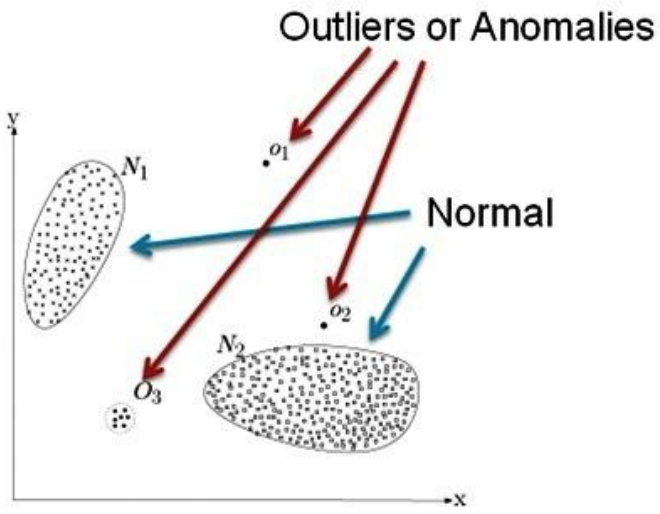


*Figure 1-3 Anomalies or Outliers[6]*

Anomaly detection is a very useful concept due to its wide application in various fields. An anomalous behavior in the network could indicate a compromised machine or a machine transmitting sensitive data out of the network. There are various challenges in an anomaly detection approach such as defining the normal behavior and abnormal behavior, capturing most of the normal behavior, etc. Due to this reason, most of the existing anomaly detection schemes tackle only a specific problem.[6].

In information theory, Entropy is defined as a measure of uncertainty or randomness associated with a random variable [7] or in this case data coming to a honeynet network. Entropy provides the measure of deviation in data items. Entropy can be used to detect anomalies in a given data set by finding out the variations in the entropy value. The entropy values of a sample of size $n$ lies in the range$[0\,, \log n]$. The entropy takes a minimum value of 0 when there is no variation in the data items (e.g., single IP address or port) and entropy takes the maximum value of $\log n$ when all the data items are distinct or the variation is large. In entropy-based detection techniques, the entropy of a random variable X with possible values $\{x_1, x_2, x_3 \ldots\ldots x_n\}$ can be calculated as

$$H(X) = -\sum_{i=1}^{n} P(x_i)\ \log P(x_i)$$

Suppose we randomly observe X for a fixed time window $w$, then $P(x_i) = m_i/m$, where $m_i$ is the frequency or number of times we observe X taking the value $x_i$, i.e.

$$m = \sum_{i=1}^{n} m_i$$

$$H(X) = -\sum_{i=1}^{n} (m_i/m) \log (m_i/m)$$

Where:

$H(X) =$ Entropy.

$m_i$ = number of packets with $x_i$ as the traffic feature

$m$ = total number of packets

If we want to calculate the probability of a traffic feature,

$$\mathrm{P}(x_i) = \frac{\text{Number of packets with } x_i \text{ as traffic feature}}{\text{Total number of packets}}$$

Here, the total number of packets is the number of packets seen for a time window T.

## 1.3 Summary

The concept of Honeynet and anomaly detection was introduced in this chapter. The different types of Honeynets were presented and the Honeynet architecture was explained in detail. This chapter has provided a brief introduction to anomaly detection concepts and the use entropy in anomaly detection has been discussed.

# *Chapter 2*

# Literature Survey

This section discusses the literature survey of various papers done in order to understand the research work done by other researchers in similar field. The success of a Honeynet depends on the way the data is collected and how we can interpret the results to better understand the vulnerabilities in our systems. The Honeynet implementation involves the deployment of a honeywall gateway and honeypots. Data capture involves monitoring and logging all the activities and traffic that enters or leaves the Honeynet. The data capture in a Honeynet is done in three layers and different formats of data are captured. The following section presents the details of the types of data captured in the honeywall.

## 2.1 Types of Data Collected in Honeynet

The Honeynet architecture includes a gateway and honeypots. The honeypots are placed behind the gateway and all the traffic to the honeypot goes through the gateway. This results in data being captured at different points within the honeynet network. The three layers of data capture are [8]:

1. All network activity (packets and full packet payload) must be captured in PCAP binary format (OpenBSD libpcap standards) and rotated on a daily basis (backing up the captured data).
2. Firewall logs must be converted to IPTables ASCII format. The data collected is stored in binary format and is used for off-line analysis of the collected traffic.
3. System activity is recorded using the format provided by Sebek. The data captured could be in the form of the attackers' keystrokes and/or operating system logs.

The Honeynet data captured must be stored in a secure location and there should be no data stored locally in the honeypots. A central location must guarantee the integrity, authenticity, and confidentiality of information. Another aspect to make data collection secure is implementing encryption [9]. Some of the standards for data capture with in a honeynet are specified in [8]. Table 2-1 summarizes[1] the types of data collected in honeynet systems.

*Table 2-1: Different Types of Data that can be captured in Honeynet*

| Data Types | Tools Required for Analysis | Size of Collected Data |
|---|---|---|
| MySQL backup (Consists of tables of Hflow database) | MySQL Admin tool, Walleye | Size if dependent on the amount of data collected |
| PCAP network traces | Wireshark, Network Miner etc. | Usually in MBs or larger |
| System Logs | Text Editor or VI | Size is based on the collected data |
| Logs created by other honeynet implementations (Nepenthes, Dionaea, HoneyD etc.) | Manual analysis and scripts | Size is based on the collected data |

The various honeypot implementations result in the collection of large amount of data of different types like: packet-captures, tcpdump data, malicious binaries, keystroke logs, and URLs of malicious websites [8]. The raw data collected from a Honeynet can be used to derive additional information on a hacker's activities. However, it becomes difficult to analyze the captured data without the use of automated analysis tools. The "needlessstack" data overload (too much data and different types of data) is one of the main challenges for Honeynet analysts [8]. Honeynets are now used widely by many researchers and network operators to analyze the network vulnerabilities. However, high-interaction honeypots collect a large amount of data from various data sources making it difficult to manage honeypots and to understand the collected data[10].

In a Honeynet, data analysis can be performed using the standard analysis tools like tcpdump, ethereal, and Walleye. The introduction of hflow and walleye web interface greatly improved the Honeynet data analysis by integrating different data sources and reducing the time to analyze the Honeynet data [5]. The limitations of Walleye user interface include the use of very constrained query language and no direct GUI access to process centric viewpoints ( i.e., failure to observe Sebek data when the system is working as designed) [5]. This implies that walleye has a limitation of not being able to show results specific to certain events and it sometimes fails to read data related to Sebek from the database. The Honeynet data analysis tools have undergone several revisions. The second generation data analysis tools use a relational model only for Sebek [11] data. The third generation tools use a relational model to store both network (flow) data and host (sebek) data [5].

## 2.2 Anomaly Detection Technique

In network security, anomaly detection plays a major role in detecting network security breaches or intrusions. Unlike its counterpart known as misuse-based or signature-based detection, the anomaly detection techniques are very useful in detecting new and unknown attack patterns. It is especially useful for detecting attacks such as [12].

- New buffer overflow attacks carrying shellcode
- New exploits
- Intentionally stealthy attacks (e.g., using ADMutate to transform a shellcode)
- Variants of existing attacks in new environments (e.g., worms using different file names as they propagate)

There exist in literature two main categories of detection techniques applied to network traffic:

**Volume-based detection techniques [13], [14], [15], [16]**: A volume-based detection scheme is useful when identifying anomalies that cause large change of traffic volume, for example, in a flooding attack or certain types of DoS attacks. The anomalies that do not cause large traffic volume changes cannot be detected by volume-based detection.

**Feature-based detection techniques**: [17], [18] The feature-based detection scheme uses the distributional changes of packet header details like IP addresses and port numbers to detect anomalies. Feature-based detection techniques require header inspection of each packet and this is time consuming and not applicable with real time constraints.

Lakhina et al [17] proposed an anomaly detection method using traffic feature distributions in which they argue that distributions of packet features like IP addresses

and ports are useful in detecting a wide range of anomalies in the network traffic. The authors stated that by using entropy along with traffic feature distribution, they can sensitively detect a wide range of anomalies. It also helps in clustering the anomalies into different clusters. In their experiment, they used network wide traffic (whole network traffic) as the data source as it contains various types of normal and abnormal traffic. The authors noted that identifying the nature of anomalies in a huge data set is a challenging task as the anomalies are a moving target. An anomaly detection system that depends on a predefined set of anomalies is inefficient as the anomalies are varying constantly. The authors pointed out that most of the anomalies affect the distributional aspects of traffic features like IP addresses and port numbers. The main difference between the method used by [17] and previous work is that they used traffic feature (such as IP address and ports) distributions to detect anomalies compared to using traffic volume to detect anomalies. They note that not all anomalies cause volume changes in traffic but most of the anomalies can be effectively detected using traffic feature distribution. The traffic features used by the authors are: source and destination IP addresses, source port, and destination port. The authors used the Principal Component Analysis (PCA) for traffic anomaly detection, which is used to separate the normal and anomalous behavior through dimensionality reduction. In our work we are using traffic coming only to Honeynet and we are using both traffic feature distributions and volume parameters to detect anomalies.

Nychis G., et al [18] presented an interesting work by conducting an empirical evaluation of using entropy for anomaly detection. The authors mainly focused on analyzing the effectiveness of using different traffic features and behavioral features distributions for anomaly detection. The behavioral features include the degree of distribution measuring

the number of distinct source and destination IP addresses that each host communicates with. They conducted various experiments and showed that the IP address and port distributions are strongly correlated and provide similar detection capabilities. The behavioral and flow size distributions are less correlated and hence detect anomalies that are usually not detected by address and port distributions. The authors calculated the correlation between different feature pairs based on the entropy values to find the correlated feature pairs. The authors suggested that the selection of traffic feature distributions must be made carefully and it must not be restricted to port/address features. In our work we are using the feature pairs that have the best detection capabilities in Honeynet traffic. The traffic features were compared and the best features found during the training phase were used for anomaly detection.

Kind A et al. [19] have proposed a new approach to the feature-based anomaly detection of Lakhina et al [17]. In their proposed approach the authors created histograms of the different traffic feature distributions and then modeled histogram patterns which are used to detect anomalies. They detect anomalies in four stages: select features and construct histograms, map into metric space, cluster and extract models, and finally classify the anomalies. In their approach, the authors use various traffic features like source and destination addresses, port numbers, TCP flags, etc. In this approach, PCA has been used for dimensionality reduction instead of differentiating between normal and abnormal traffic as done in [17]. The main difference of this approach is in the use of histograms to detect anomalies instead of using entropy. In our thesis work we are using entropy values of different features along with clustering techniques to detect anomalies in Honeynet traffic.

Ping, D. and S. Abe [15] proposed an IP packet size entropy-based DoS detection scheme in which changes in the IP packet size entropy (IPSE) is used to detect possible DoS attacks. The authors note that various applications have different packet size profiles and this distribution changes in the presence of potential DoS attacks. The authors illustrated that the various applications have default packet sizes with respect to request/response data (various services have default packet sizes based on the service provided). For example, FTP applications have 40 byte acknowledgement and full packet data of 1500 bytes. In the presence of attacks, the generated packets are of identical sizes irrespective of the response from the victim. The threshold of entropy is obtained by self-learning from legitimate traffic data. After setting the threshold value, the entropy that exceeds this value indicates the presence of attack traffic. The IPSE approach was able to detect short term as well as long term attacks; which is an improvement over the traditional volume-based schemes. In our approach we utilized the detection capabilities of volume-based schemes along with the feature-based detection schemes.

Thonnard, O. and M. Dacier [20] proposed a clustering-based approach to detect attack patterns in Honeynet data. In their approach, they specifically use time signature to cluster the Honeynet data. Time series is defined as a sequence of data points measured at successive times separated by uniform time intervals. Figure 2-1 shows how two attacks have temporal similarities even though they target different ports. They conducted experiments on large data sets collected from 44 worldwide distributed honeypots.

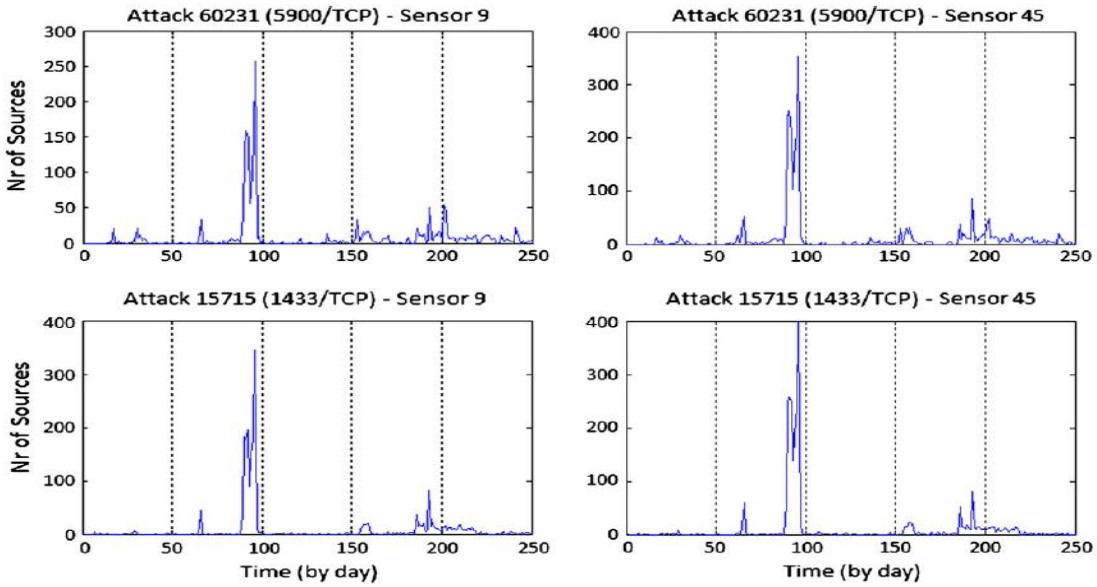


*Figure 2-1 Two different attacks detected using two different sensors [20]*

The attack source is identified as an IP address that targets the honeynpot on a given day with certain port sequence. The network characteristics used by the authors include: (i) the number of virtual machines targeted on a platform, (ii) the number of packets sent to each virtual machine, (iii) the total number of packets sent to the platform, (iv) the duration of the attack session, (v) the average inter-arrival time between packets, and (vi) the associated port sequence. In our work we are applying an entropy based anomaly detection technique to detect anomalies in Honeynet data.

Al-Haidari et al [21], proposed an entropy-based countermeasures against DoS attacks. In their work they used packet size entropy to and the corresponding threshold values to distinguish normal traffic and attack traffic. In their work they illustrated that entropy-based scheme enhances the performance of the firewalls in terms of throughput, delay and availability by isolating the attack traffic from the legitimate traffic. In our work we are using the threshold levels used in this approach and applying it in detecting anomalies in honeynet traffic.

## 2.3 Summary

This chapter presented the research done by various other researchers in the anomaly detection techniques. In the first part the main problems related to Honeynet data analysis was presented. In the later part different anomaly detection approaches proposed by various researchers were presented. The main difference between the existing work and in our proposed approach is that we are using both feature-based detection scheme and volume-based detection scheme to detect anomalies in the Honeynet traffic. The main purpose of this contribution is to aid the analysts in analyzing Honeynet data.

# *Chapter 3*

# Honeynet Data Analysis Problem and Proposed Solution

This chapter focuses mainly on the main problems that are faced in Honeynet data analysis and the proposed solution. It discusses the anomaly detection approach used in this thesis work. The later part of the chapter provides a detailed description of the anomaly detection approach used in this work. The current Honeynet system does not include anomaly detection schemes to identify anomalies in the Honeynet traffic. Anomaly detection is useful for detecting zero day attacks and unknown attacks in the network. A Honeynet also collects a substantial amount of data and any incoming data to the Honeynet is considered malicious. Many Honeynet deployments currently use Snort (a Signature-based Intrusion Detection tool), to detect malicious activities, but it is known to generate high rate of false positives [22].

## 3.1 Problem Statement

Honeynet captures information that can be used by administrators to improve their network security, but the size of the data collected can be overwhelming [23]. Honeynets

depend on a signature-based detection scheme, manual analysis and expertise to identify malicious activities. For example the high-interaction honeypot Honeywall CD-ROM uses Snort: a signature-based detection tool to detect malicious activities. The signature-based detection schemes cannot detect new attacks [24]. Honeynet traffic is different from any other network wide traffic as it has little or no production traffic. Any traffic that enters or leaves the Honeynet is suspicious by nature. However, in order to identify malicious activities in this traffic requires manual analysis and expertise of identifying such attacks.

There are very few anomaly detection techniques addressing the Honeynet systems and most of the traffic is analyzed manually which requires expertise to identify different types of attacks. The few existing approaches mostly focus on detecting botnets and worm or virus outbreaks as they analyze traffic collected from low-interaction honeypot sensors setup across the world. Honeynet traffic is different from other types of network traffic as every packet that enters or leaves the Honeynet is considered malicious. Due this most of other anomaly detection approaches which rely on regular network wide traffic are not well suited for this type of traffic [20]. As stated before Honeynet is used by various organizations to proactively improve their security hence the malicious activities must be detected quickly and with relative ease. Another more important use of Honeynets is to identify the tools tactics or behaviour of different attacks and share these findings. In order to address these issues we propose a simple and easy to use anomaly detection technique which can be used to identify anomalies in Honeynet traffic and also to identify the behaviour of various malicious activities.

In our proposed approach, we are using both feature-based and volume-based detection schemes to identify anomalies in Honeynet traffic. Most of the other research work in literature is focused mainly in comparing the effectiveness of these two techniques or to propose a technique based on either one of these. In our work the main focus is to identify the best features that can be used to identify anomalies in Honeynet traffic and also to identify the behaviour patterns of various types of attacks.

The main aim of this work is to effectively analyze and identify attacks in the network traffic collected by Honeynet deployments. In this work, we propose an anomaly detection technique which uses both feature-based and volume-based schemes to detect anomalies in the Honeynet traffic. The proposed detection scheme can be used to detect outliers in the Honeynet traffic and help in identifying the behavior models for various types of attacks. The proposed approach uses a combination of packet header details and volume changes to identify malicious activities. This approach has not been used before to analyze Honeynet traffic. Apart from this, the combination of features is selected by extensive evaluation of various features.

The above proposed method is composed of the following main steps:

1. Analyzing Honeynet traffic data and identifying the candidate features suitable for anomaly detection.
2. Selecting the features that provide good detection capabilities from those available in the literature as well as those obtained from the manual data analysis.
3. Designing and implementing a suitable anomaly detection technique.
4. Identifying anomalies in Honeynet data by classifying the traffic into normal or abnormal traffic.

It must be noted that since the main idea of using a Honeynet is to perform offline data analysis of the captured data, real time constraints in detecting anomalies do not apply. However, anomaly detection should be done in a reasonable amount of time to perform an effective and quick data analysis.

The Honeynet project is a useful resource to learn the tools, motives and tactics of the blackhat community. Using an anomaly detection scheme in Honeynet will greatly improve the data forensics and the detection of unknown and new attacks.

Assumptions: Data source used for anomaly detection will in the form of PCAP files captured from Honeynet systems

## 3.2 Analyzing Honeynet Test Data Set

In order to detect anomalies in the network traffic, we first need to analyze different Honeynet traffic data sets to understand the difference between the normal and abnormal behaviors. Honeynet traces were collected mainly from the honyenet.org site which mainly includes the scan of the month (SOM) challenges and Forensic Challenges released by the Honeynet organization [25]. The other sources of traces were from hack.lu 2009 Information Security Visualization Contest [26] and from the Honeynet deployment in KFUPM. The Honeynet traces that were used are list in Table 3-1.

*Table 3-1: Honeynet Traffic Test Datasets used for Analysis*

| Traffic Data Set Name & Source | Description | Traffic Details |
|---|---|---|
| Pcap Attack Trace, Honeynet.org – Forensic Challenge | The network traffic captured in the file attack-trace.pcap relates to an automated malware attack that exploits the Windows Local | 348 packets<br>Total duration 16 sec |

| Traffic Data Set Name & Source | Description | Traffic Details |
|---|---|---|
| | Security Authority (LSA) Remote Procedure Call (RPC) service. | |
| Scan 28 - Honeynet.org – Scan of the Month | Trace collected by the Mexico Honeynet Team - Italian blackhats break into a Solaris server then enable IPv6 tunneling for communications. | Two traces:<br>Day1: 18843 Packets – 24 Hours<br>Day 3: 123123 Packets – 24 Hours |
| Scan 14 - Honeynet.org – Scan of the Month | This trace is about a successful Windows NT attack. | 6707 packets<br>Total Duration 20 Hours |
| Scan 19 - Honeynet.org – Scan of the Month | Trace of Redhat Linux 6.2 honeypot compromise. | 24440 packets<br>Total Duration 23 Hours |

The traces provided by the Honeynet organization are instances of real compromises that were captured by different honeynet chapters. The main reasons for releasing such challenges are to help the network security analysts to hone their forensic and analysis skills to get an in-depth knowledge of real attacks. These traces proved crucial in our work to characterize and identify the important features in the honeynet traffic. As these traces are collected in a real environment and specifically in a honeynet setup, it was of more importance to our work.

These traces were analyzed to identify the suitable characteristics / features that can be used for anomaly detection. The analysis was done using tools like Wireshark and NetMiner. The classification techniques that were used in this work are:

**Feature-based classification:** Uses features like Source Address, Destination Address, Port, protocol, etc. Various features were analyzed and a list of possible features was selected.

**Volume-based classification:** In this scheme, the change in traffic volume was used to detect anomalies in the network traffic.

The list of features that were recorded from the literature and identified during test data analysis is mentioned in

Table 3-2.

*Table 3-2 List of Features Selected from Test Data Analysis and Literature*

| Traffic Features | Description |
|---|---|
| Source Address[17] | Source IP address: This parameter indicates the entropy of the unique IP addresses of incoming connections to the honeypot |
| Destination Address[17] | Destination IP address: The destination IP entropy indicates the number of external connections initiated by the honeypot. |
| Source Port[17] | Port number of the source machine. This attribute indicates the number of source ports that are visible during each interval. |
| Destination Port[17] | Destination port number. This parameter indicates the number of destination ports visible during each interval. |
| Indegree[18] | Number of distinct Hosts that connect to the observed host. This parameter indicates the number of incoming connections to the honeypot. |
| Outdegree[18] | Number of distinct IP address the observed host connects to. This feature measures the number of outgoing connections from the honeypot. |

| | |
|---|---|
| Packet Size Distribution[15] | Various packet sizes visible in the network traffic |
| Application Protocol Used | Application protocol seen during the conversation (eg. SSH, SMTP, FTP etc) |
| Origin of IP address - Country | The distribution of countries from which the observed host gets connections |
| **Volume Features** | **Description** |
| Average number of bytes per TCP packet per minute [27] | Average TCP packet size per minute |
| Average number of bytes per UDP packet per minute [27] | Average UDP packet size per minute |
| Average number of bytes per ICMP packet per minute [27] | Average ICMP packet size per minute |
| Sum of average packet size[27] | Aggregate sum of |
| Total Payload Bytes | Total bytes seen in the five minute interval |
| **Volume Features** | **Description** |
| Average Inter-arrival times | Average inter-arrival time of packets in five minute interval |
| Average Payload Size | Average payload size seen during the five minute interval |
| Total Packets | Total packets seen during the five minute interval |

Some of the features which provided redundant information were eliminated such as application protocol used can be also be detected using the ports used. Similarly, instead of using the average packet sizes for different transport protocols we choose average packet size.

*Table 3-3: Traffic Features used for a Detailed Analysis*

| Traffic Features | Volume Features |
|---|---|
| Source Address | Average Packet Inter-arrival Time |
| Destination Address | Total Payload bytes received during the interval |
| Source Port | Average Payload size during the interval |

| | |
|---|---|
| Destination Port | Total Packets received during the interval |
| Packet Size Distribution | |
| Indegree & Outdegree | |

## 3.3 Feature Evaluation

The features listed in Table 3-3 were tested for their detection capabilities. For testing these features, we used entropy-based traffic feature distributions and volume distribution (for volume-based features) to determine the type of anomaly that the particular feature could detect. The real honeynet traces obtained from Honeynet.org were used to test the effectiveness of each individual feature. The entropy distributions were obtained by calculating the entropy values of each five minute interval using Equation 3-1.

$$H(X) = -\sum_{i=1}^{n} P(x_i)\ log\ P\ (x_i) \qquad \textit{Equation 3-1 Entropy}$$

The sliding window concept was used to gather entropy values in overlapping intervals (refer to Figure 3-1) so that any valuable information is not missed in cases where anomaly overlaps across multiple intervals.

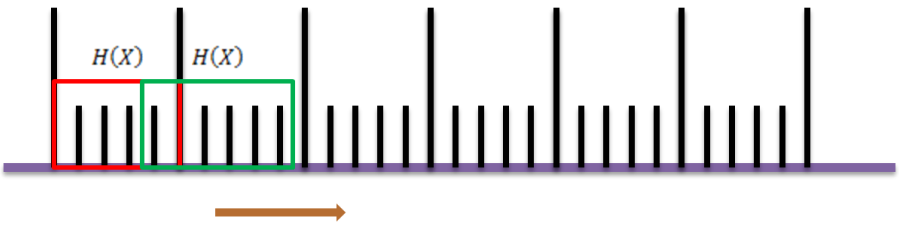


*Figure 3-1: Sliding window used for calculating entropy*

The entropy values of each feature were recorded and further manual analysis of was done to identify the normal behavior and anomalous behavior. Initially, all the features listed in Table 3-3 were tested, and later the best features that can be used for anomaly

detection were selected. Different features show different variations in the entropy values and the features that detected the most anomalies were determined.

## 3.3.1 Data Set: Scan 28.

This dataset was published in the scan of the month challenges in the honeynet.org website. The trace was collected by the Mexico Honeynet Team which is about the Italian blackhats that broke into a Solaris server and then enabled IPv6 tunneling for communications. The day1 traffic set has the honeypot being compromised and the day3 traffic consists of the IPv6 tunneling enabled by the blackhats for communication.

### *3.3.1.1 Day 1 Traffic*

The destination port entropy does not show much activity in the first 9 hours after which there is a drastic change in the traffic behavior (refer to Figure 3-2). When we check the volume feature, i.e., the total packets in the interval after the $9^{th}$ hour, it is clear that there was a malicious activity after the 9th hour (refer to Figure 3-5). The manual analysis of the PCAP trace reveals that the honeypot was probed for a specific vulnerability and then compromised during this time. The destination port entropy value first increases; as the target machine was probed for vulnerable ports. Then, the specific attack on a vulnerable port was launched to access the command line of the target machine. Once the access to the shell was obtained, the rootkit was downloaded and installed into the honeypot. This is visible as a spike shown in the total packets per interval plot. The packet size entropy indicates the different packet sizes visible during the interval in which the attack occurred. For example, during a network scan, a constant packet size will be used; and during data transfer, there could be packets of smaller and larger sizes. The packet size

entropy shows visible changes in the entropy values when the system was compromised (refer to Figure 3-6).

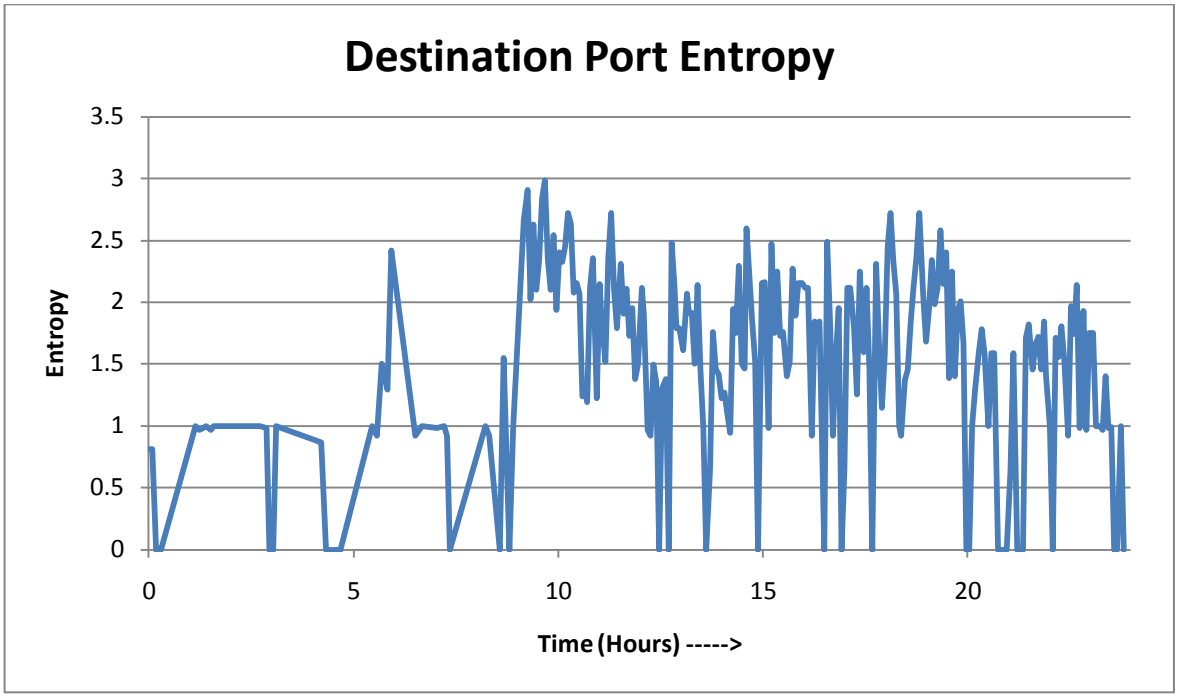


*Figure 3-2: Destination Port Entropy in Day1 traffic of Scan28 data set*

*Figure 3-3: Destination IP entropy of Day1 in Scan 28 data set*

The destination IP entropy and outdegree show similar behavior but destination IP entropy gives a better understanding of the changes in the traffic behavior (refer to Figure 3-3 and Figure 3-4).

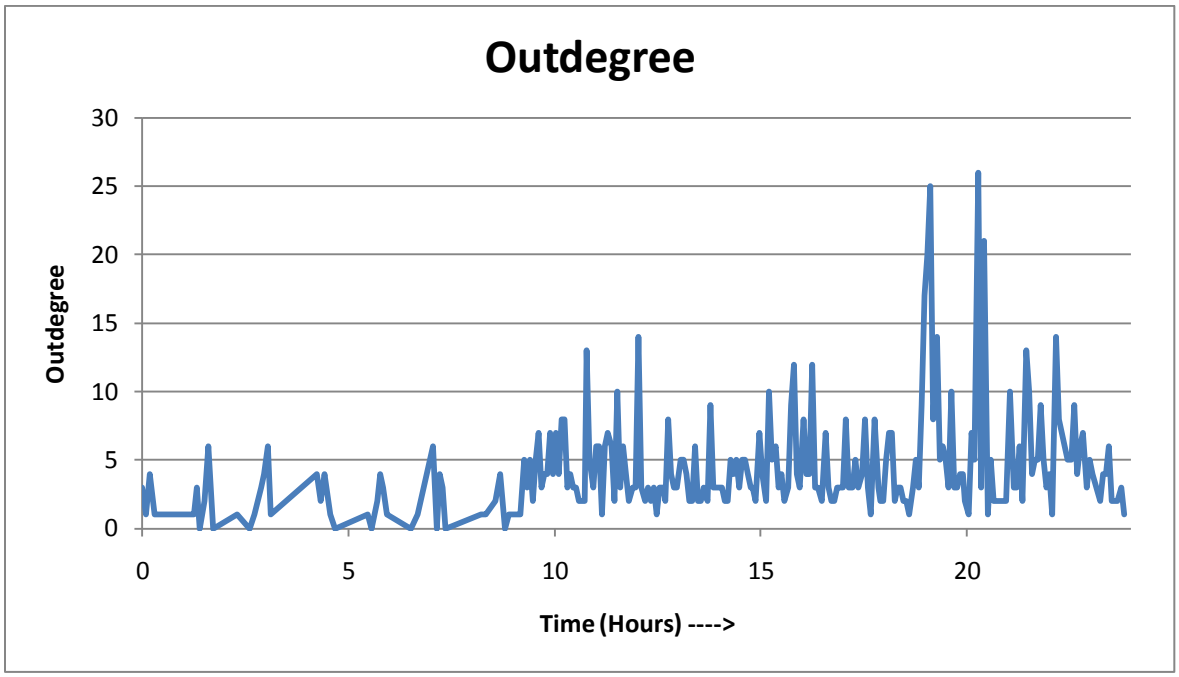


*Figure 3-4: Outdegree in Day1 traffic of Scan28 data set*

*Figure 3-5: Total Packets per Interval in Day1 traffic of Scan28 data set*

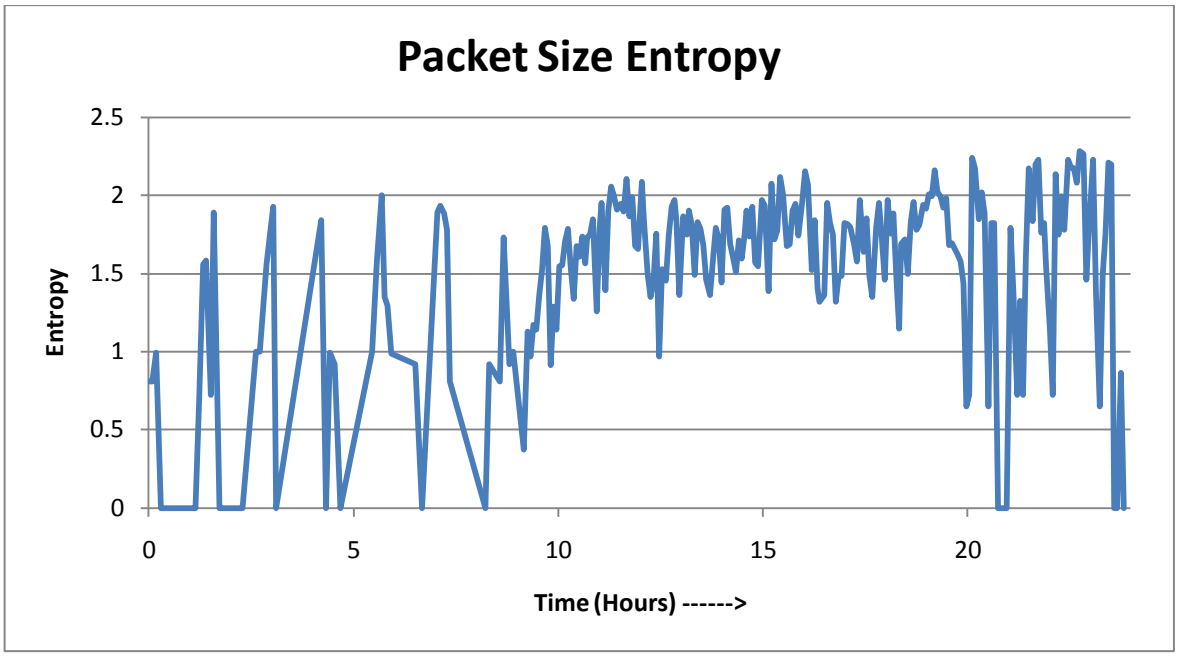


*Figure 3-6: Packet Size Entropy in Day1 traffic of Scan28 data set*

The use of different feature distribution helps in understanding the behavior of the anomaly. The features that gave a clear indication of anomaly were destination port entropy, source port entropy and total payload bytes and total packets. The packet size entropy also showed the change in behavior but it does not help in understanding the anomaly behavior.

#### *3.3.1.2 Day 3 traffic*

The day 3 traffic shows less activity in the initial hours, but around the 6th hour the traffic pattern changes. The manual analysis of the trace shows that the hacker had initiated an IRC connection to an external server. The source port entropy (refer to Figure 3-7) shows a drastic increase in the entropy value around the 15th hour. The honeypot was used to attack other machines and a detailed analysis shows that a port scan / DoS attack was initiated against other machines. During the attack the target machine was probed for open ports and different source ports were used to probe the target ports.

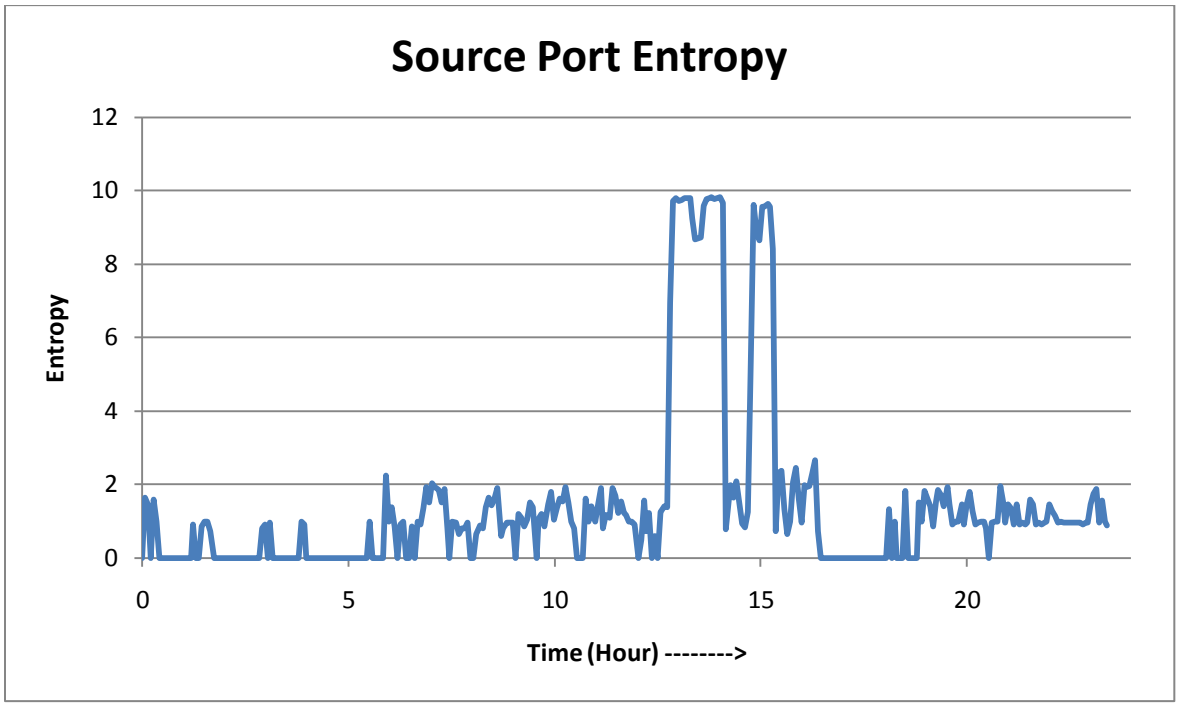


*Figure 3-7: Source Port Entropy in Day3 traffic of Scan28 data set*

The destination IP entropy and outdegree do not give a very clear picture of the changes in the traffic. The only variation that can be seen is around the 20th hour where there are few peaks. (Refer to Figure 3-8 and Figure 3-9)

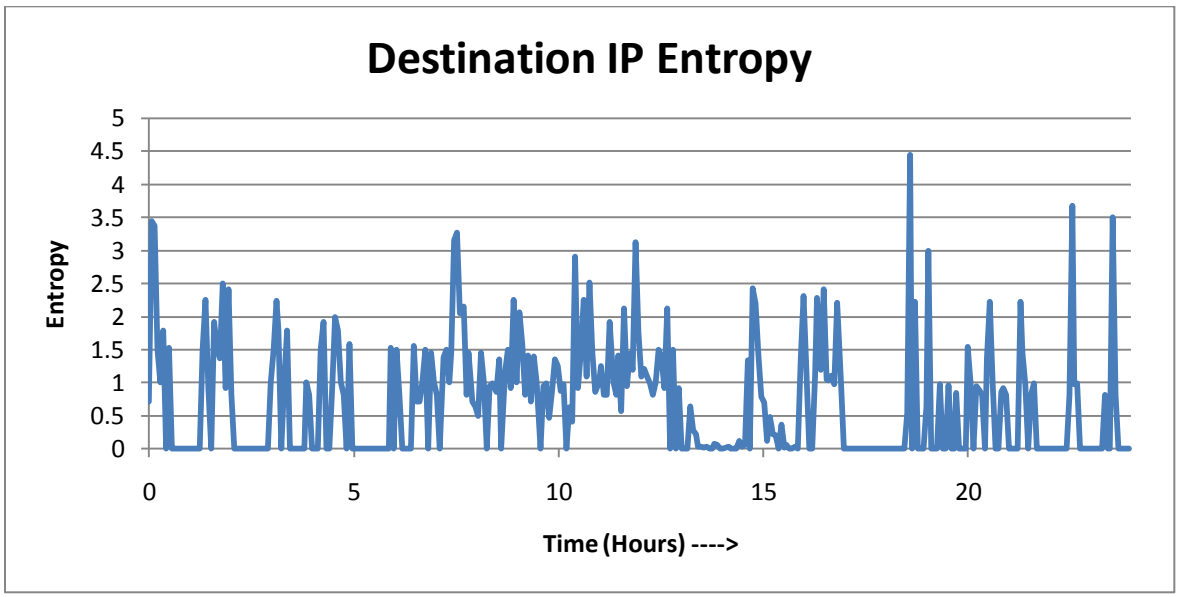

*Figure 3-8: Destination IP Entropy in Day3 traffic of Scan28 data set*

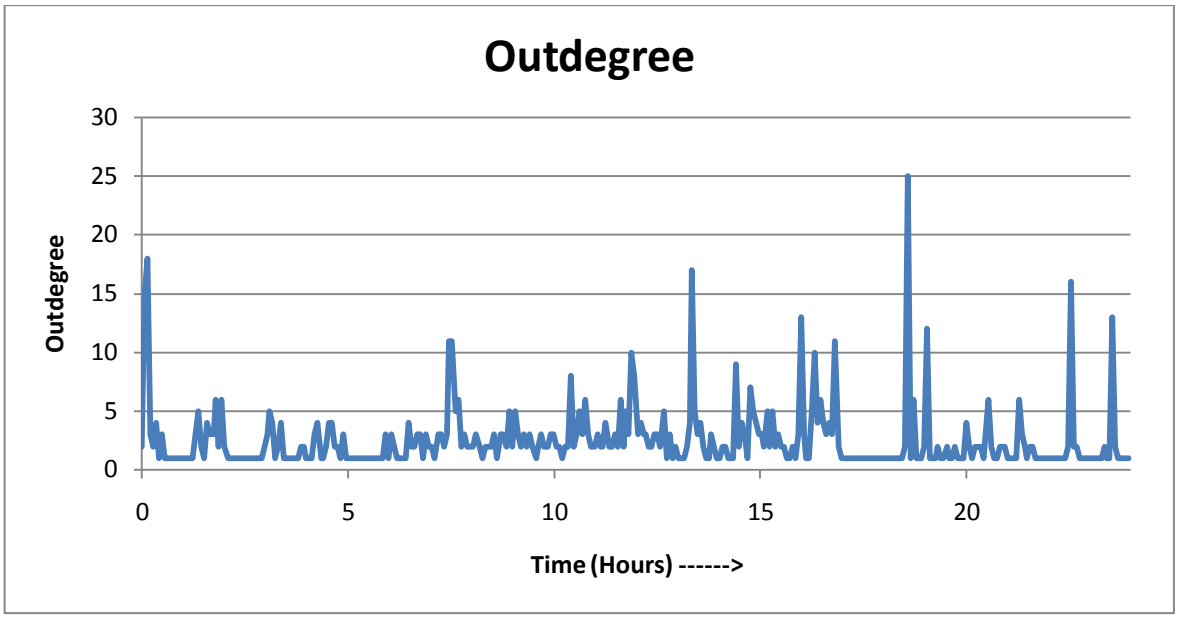


*Figure 3-9: Outdegree in Day3 traffic of Scan28 data set*

The total payload bytes show that a large amount of data was transferred after the 15th hour and a manual analysis of the trace reveals that a file was transferred to the honeypot (refer to Figure 3-10). The dominant features that were helpful in detecting the events in the trace were source and destination port entropy, total payload bytes and total packets.

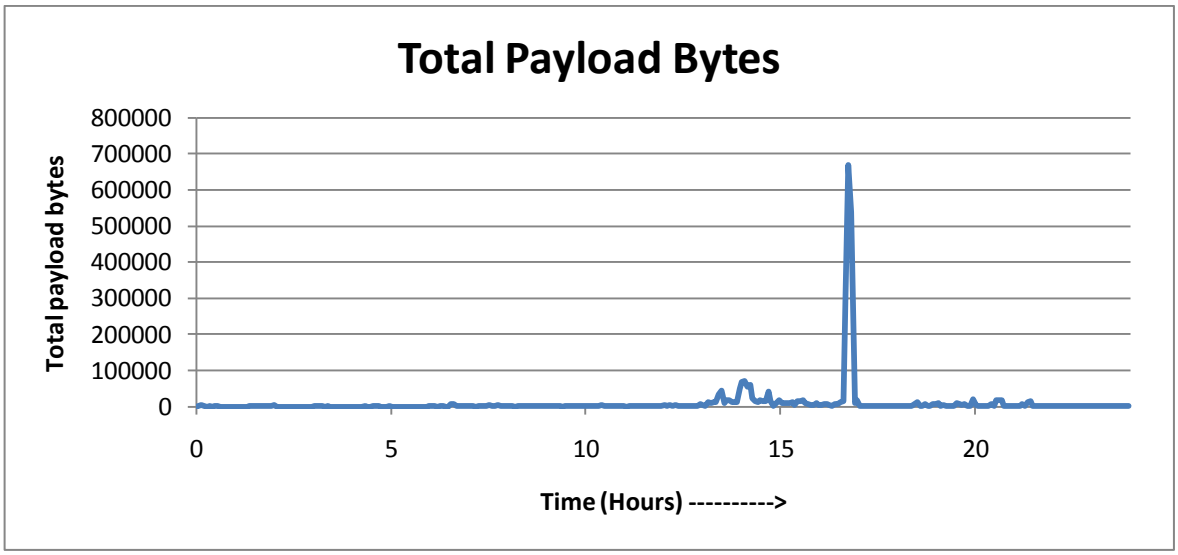


*Figure 3-10: Total Payload Bytes Day3 traffic of Scan28 data set*

### 3.3.2 Data Set: Scan 14

This trace is about a Windows NT machine attacked successfully. The attacker exploited a vulnerability in Microsoft® Data Access Components (MDAC) that could allow a web site visitor to take unauthorized actions on a web site hosted using Internet Information Server. The destination IP entropy shows a sudden increase in the entropy during the compromise of the target machine (refer to Figure 3-11).

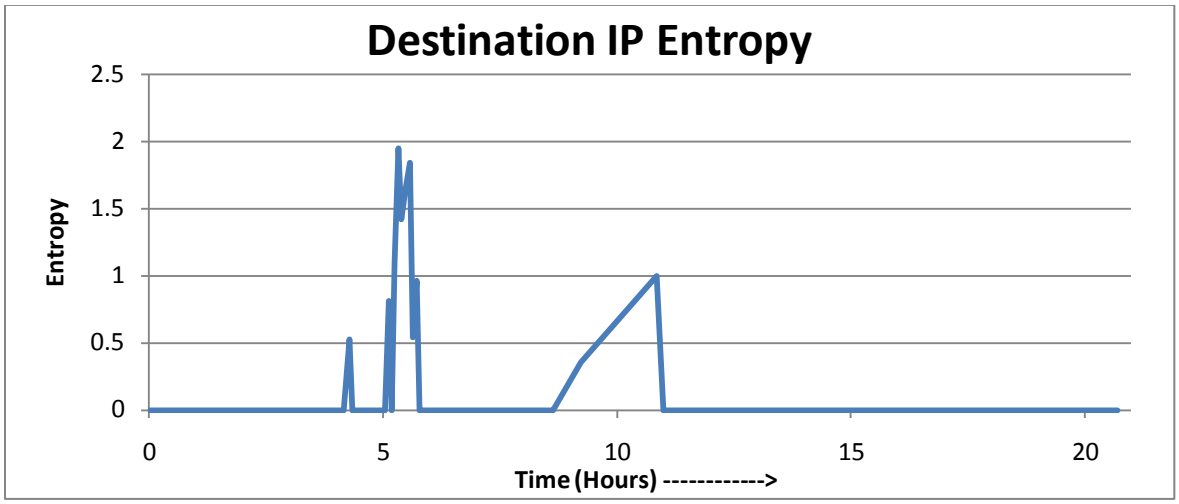


*Figure 3-11: Destination IP Entropy for Scan14 challenge*

*Figure 3-12: Destination Port Entropy for Scan14 challenge*

The destination port entropy shows a different behavior during the period when the target machine was being compromised (refer to Figure 3-12). The volume features: total packets and total payload bytes (refer to Figure 3-13 and Figure 3-14) show the intervals when large data or files were transferred to the target machine. Both total packets and total payload bytes show large variation when some data transfer took place.

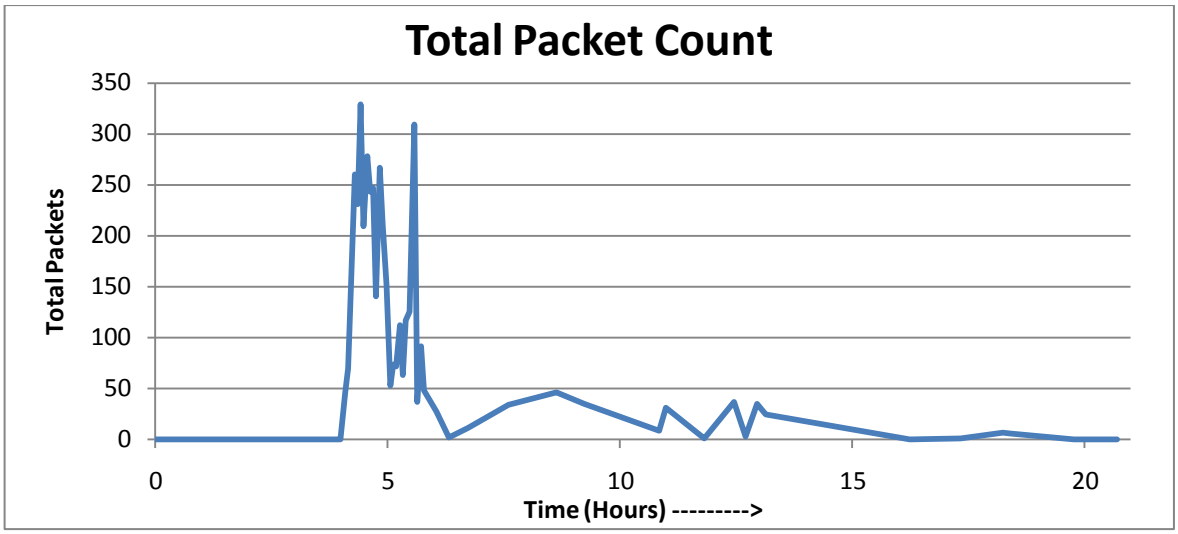


*Figure 3-13: Total Packets for Scan14 challenge*

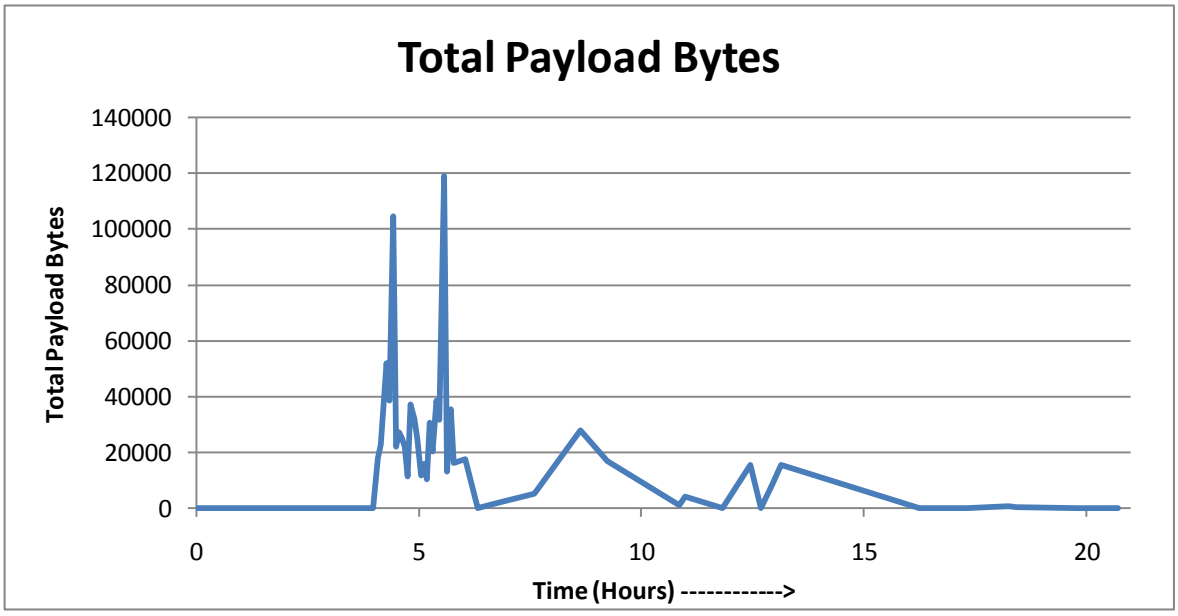


*Figure 3-14: Total Payload Bytes for Scan14 challenge*

It is clear from these trace that even in a short duration trace it is possible to detect the anomalies using the entropy of traffic features and volume features.

### 3.3.3 Data Set: Scan 19

This trace was captured during a Red Hat Linux honeypot compromise. The attacker exploited the vulnerability in the wu-ftpd (Washington University FTPD software) package. After compromising the machine the attacker used three different modes to connect and execute the commands. The destination port entropy (refer to Figure 3-15) shows that there was not much traffic for nearly 20 hours and then there is a sudden dip in the entropy followed by a sharp increase. The dip in the entropy occurred when the attacker tried to exploit the specific vulnerability in the honeypot. The attacker then downloaded and installed rootkit programs to the honeypot this based on the peak visible in total packets and total payload bytes received (refer to Figure 3-16 and Figure 3-17). Later the attacker performed a port scan of the compromised machine.

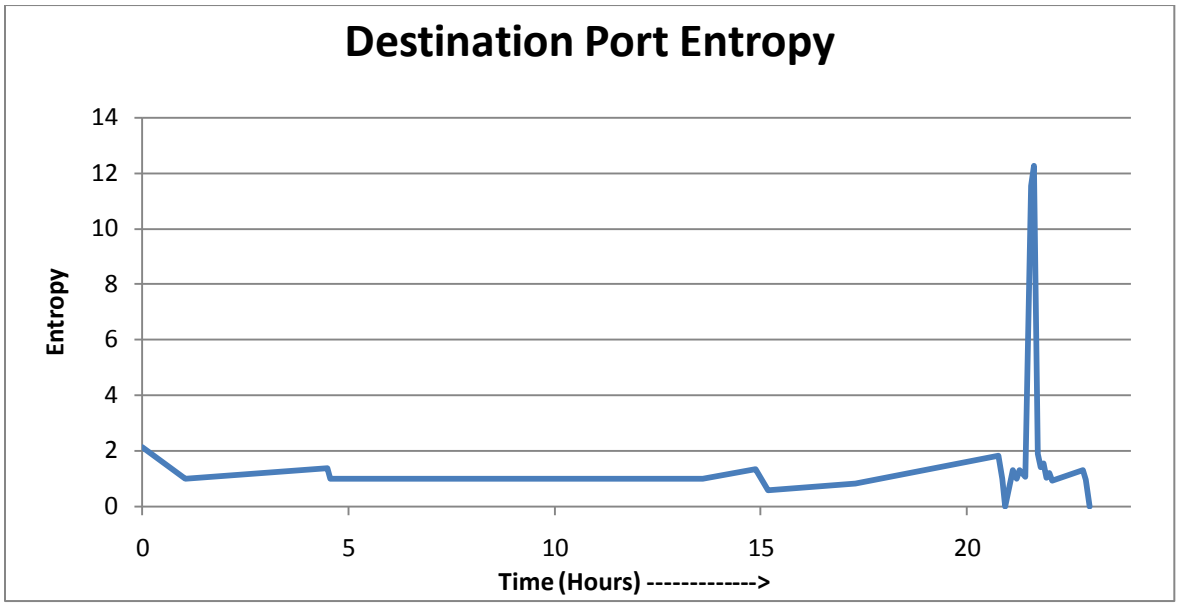


*Figure 3-15: Destination Port Entropy for Scan19 challenge*

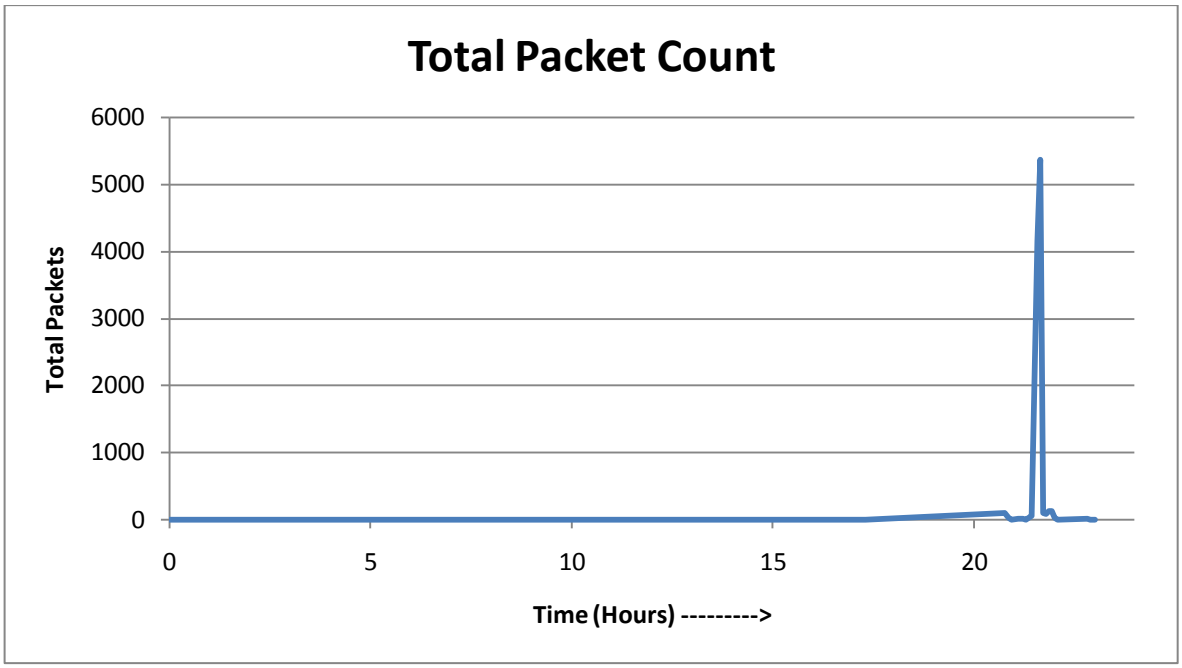


*Figure 3-16: Total Packets for Scan19 challenge*

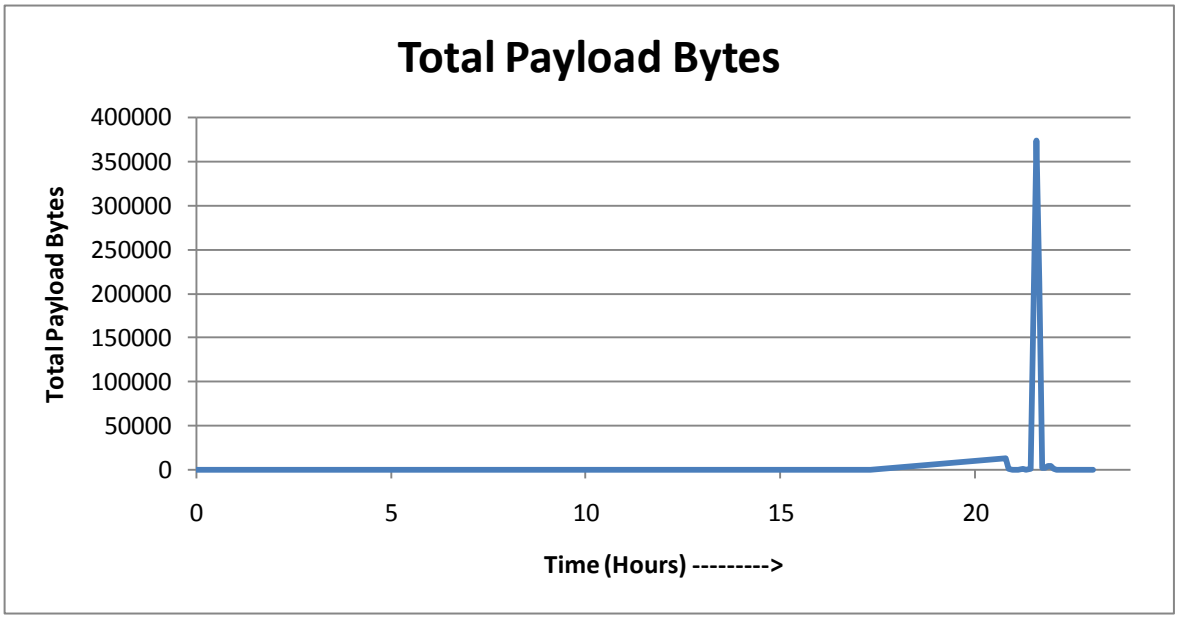


*Figure 3-17: Total Payload Bytes for Scan19 challenge*

The importance of volume features is clear in this trace as they help in understanding the attacker's behavior during the system exploit. The other parameters like outdegree and indegree were not very useful in giving a good understanding of the behavior.

### 3.3.4 Dataset: SSH based Honeypot traffic

The feature analysis tests were also carried out on a large dataset collected from an SSH based honeypot which includes 12 days of traffic. The data set includes mainly SSH traffic and an unknown number of anomalies. The traffic includes anomalies such as network scans, rootkit file transfers, IRC traffic, etc.

The destination IP entropy indicates the number of external connections initiated by the honeypot (refer to Figure 3-18). The peaks indicate that the honeypot initiated a large number of connections during that interval. The high value of Destination IP entropy indicates that the honeypot was scanning the network. The packet size entropy does not provide a clear indication of the anomalies in the traffic (refer to Figure 3-19). As a result this feature was considered less efficient for the anomaly detection.

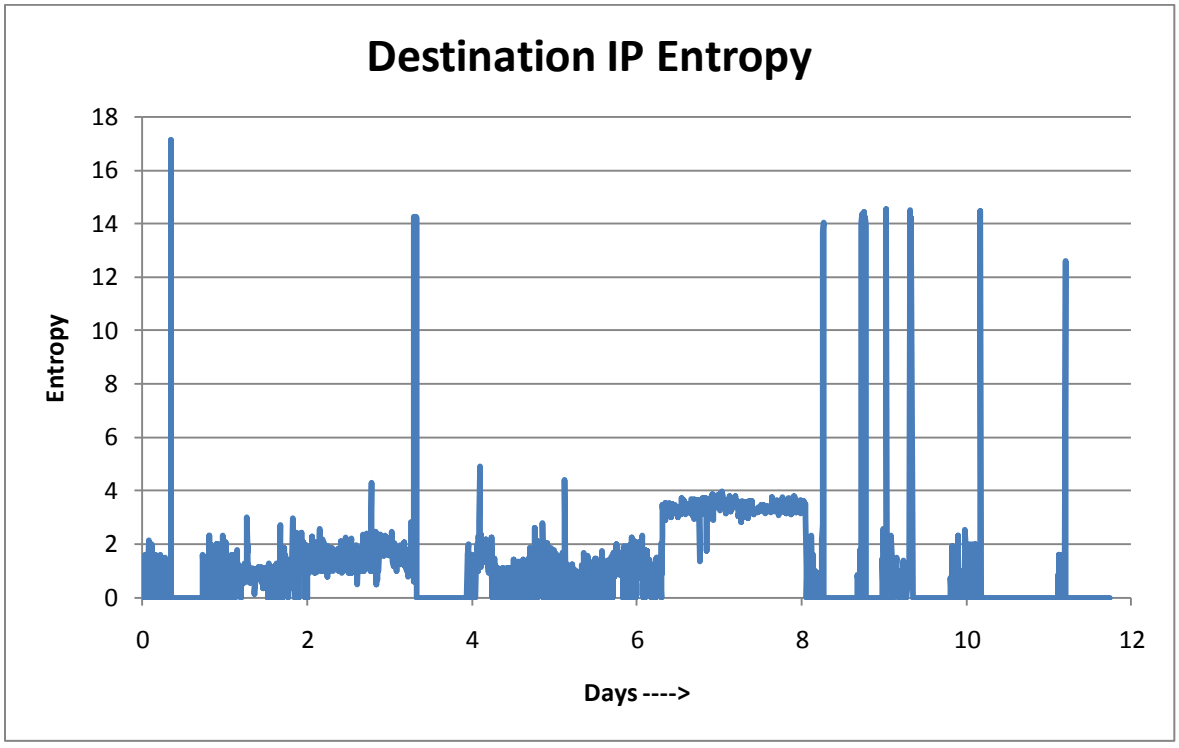


*Figure 3-18: Destination IP entropy of SSH based honeypot trace*

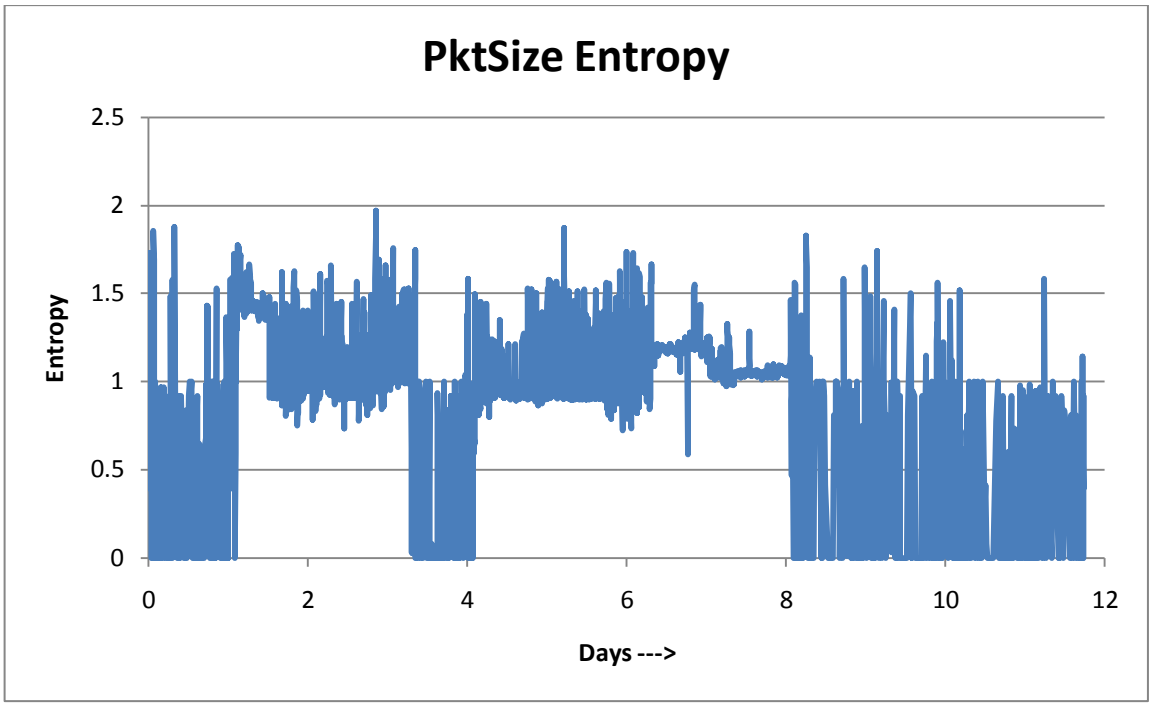


*Figure 3-19: Packet Size entropy of SSH based honeypot trace*

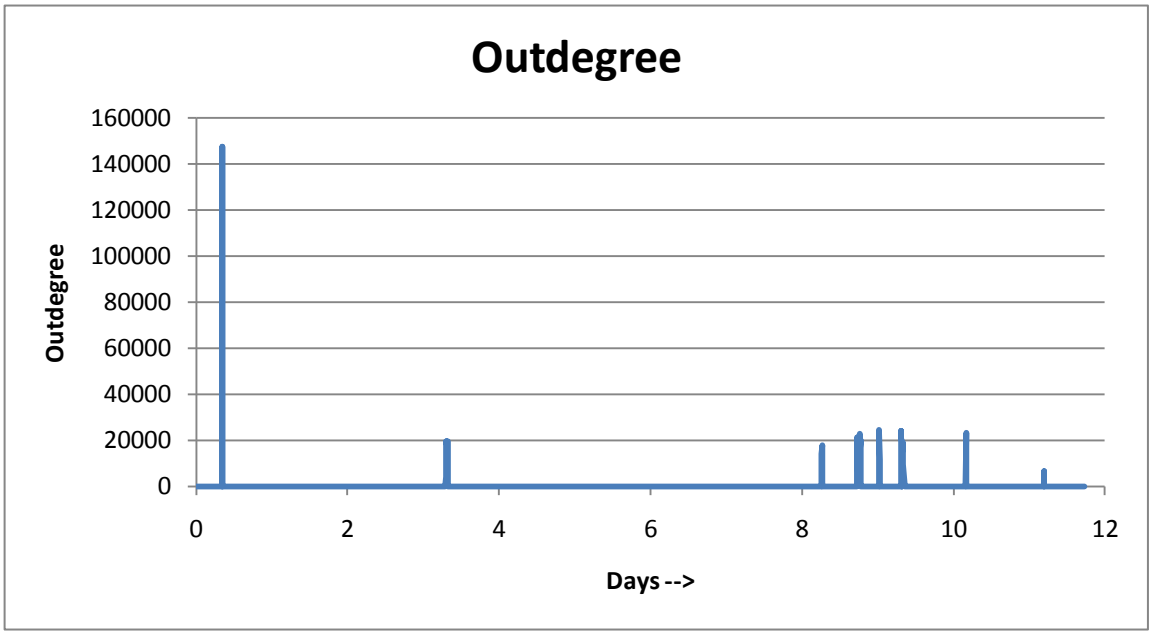


*Figure 3-20: Outdegree distribution of SSH based honeypot trace*

The peaks in the Figure 3-20 are due to the network scan activity done by the infected honeypot. However the information of outdegree is similar to destination IP entropy and

is less clear compared to it. The source IP entropy in Figure 3-21 shows a similar behavior as the destination IP entropy.

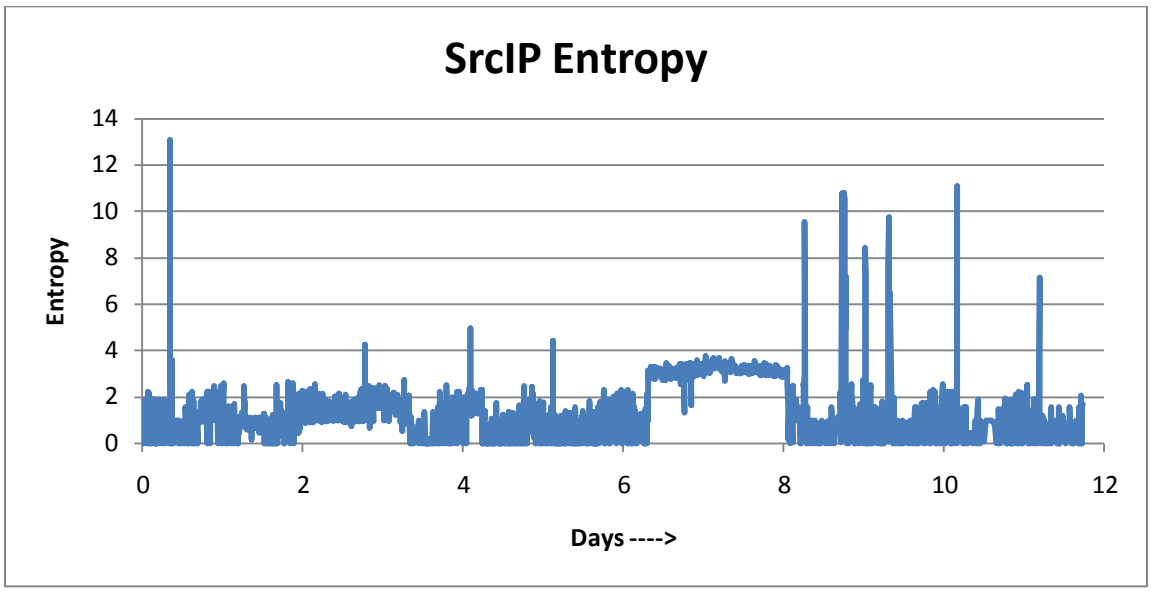


*Figure 3-21: Source IP Entropy distribution of SSH based honeypot trace*

The indegree (refer to Figure 3-22) also does not show all the anomalies and due this feature was selected for anomaly detection.

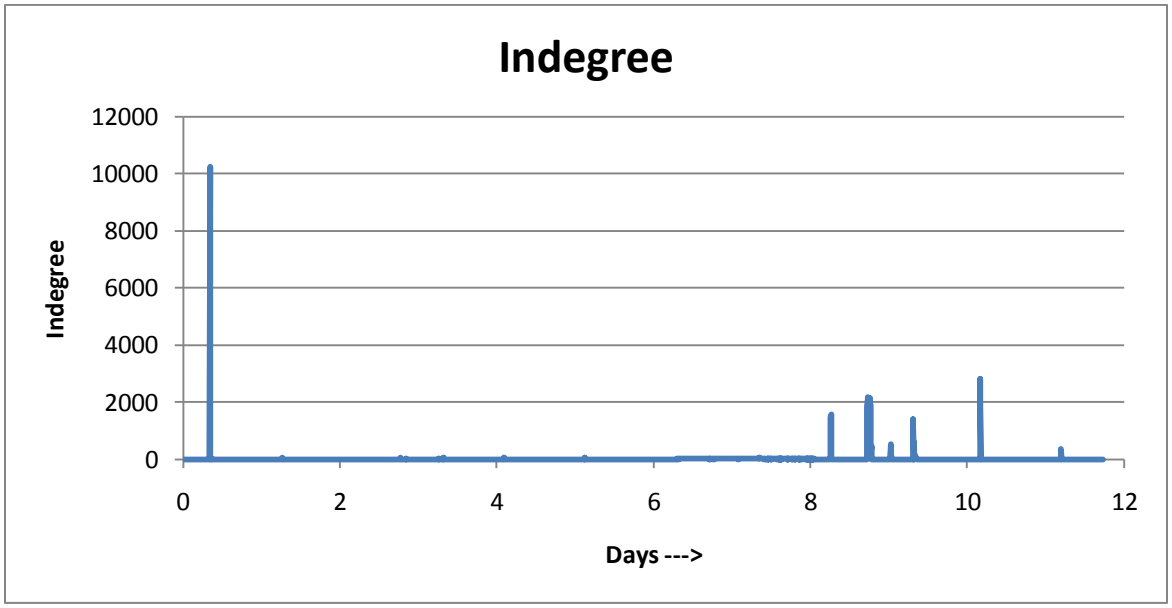


*Figure 3-22: Indegree distribution of SSH based honeypot trace*

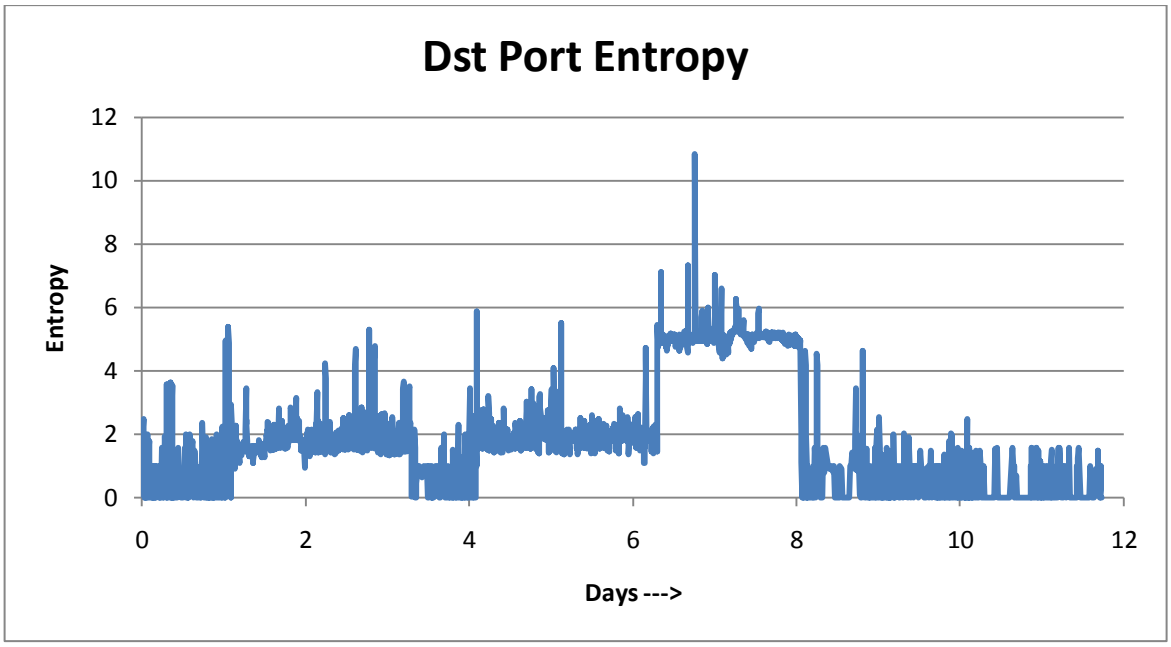


*Figure 3-23: Destination Port Entropy distribution of SSH based honeypot trace*

The destination port entropy shows the peaks that were visible during anomalies. The port scan activity is indicated by the peak at 6.7 days (refer to Figure 3-23).

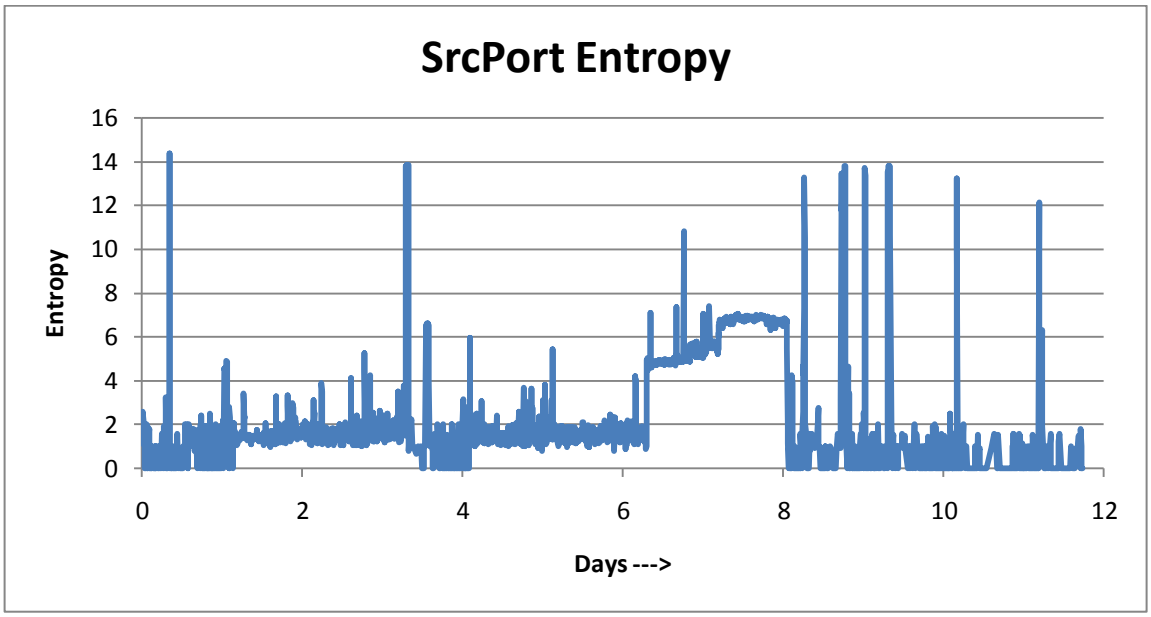


*Figure 3-24: Source Port Entropy distribution of SSH based honeypot trace*

The source port entropy shows (refer to Figure 3-24) the peaks that occurred during the network scan event and this feature along with destination IP entropy help us in understanding the network scan anomaly. Based on the analysis of the feature parameters of SSH based trace, it is clear that destination IP entropy, source port entropy and destination port entropy had better detection capabilities than the other features. Features like outdegree could detect only one kind of anomaly (network scan) hence they were considered less efficient.

Volume based features like total payload bytes also helped in understanding the behavior and the anomalous events. The Figure 3-25 shows that before the network scan event a large data transfer took place. When we manually analyzed the trace we found that this was related to a malicious file transfer which was later used to initiate the network scan activity.

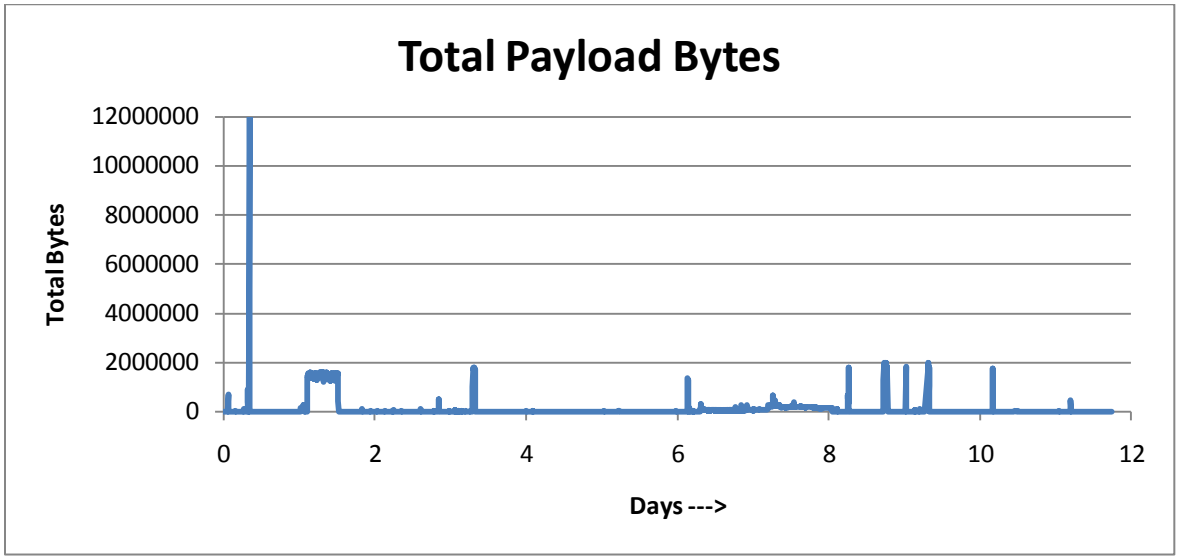


*Figure 3-25: Total Payload Bytes distribution of SSH based honeypot trace*

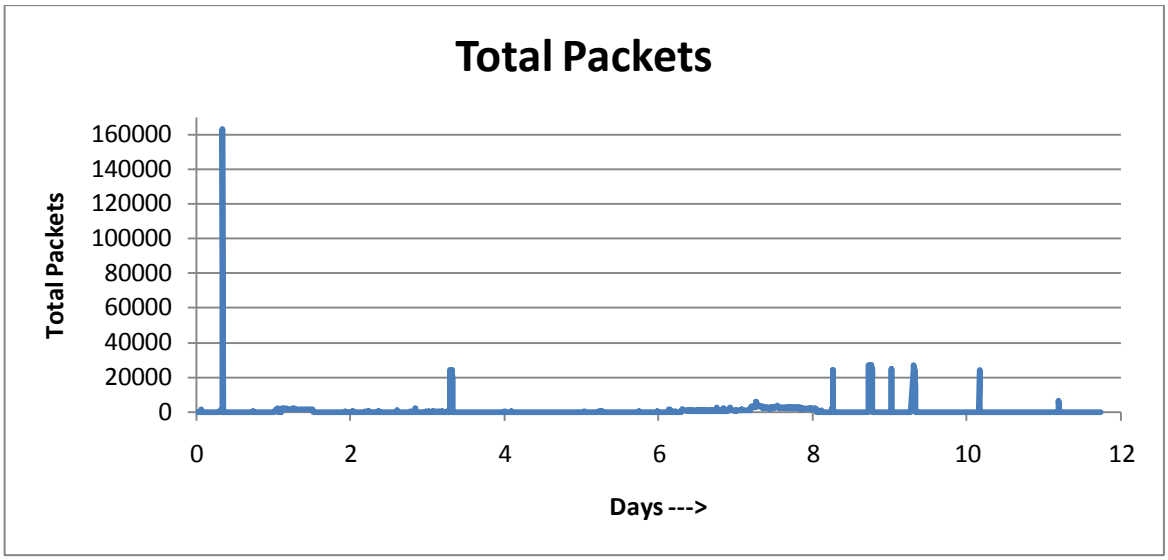


*Figure 3-26: Packet Count Distribution of SSH based honeypot trace*

The packet count feature (refer to Figure 3-26) also gives better understanding of the anomalies.

# 3.4 Combining different features to detect anomalies

Using individual features helps only in detecting certain anomalous events and it does not give a clear understanding of the anomaly that occurred. To get a clear understanding of the exact events that occurred, we need to look into a combination of features. This is useful to detect certain anomalies that were not visible using a single feature. A number of combinations of the above listed features were tested to identify the useful feature combinations and get a better understanding of the anomaly.

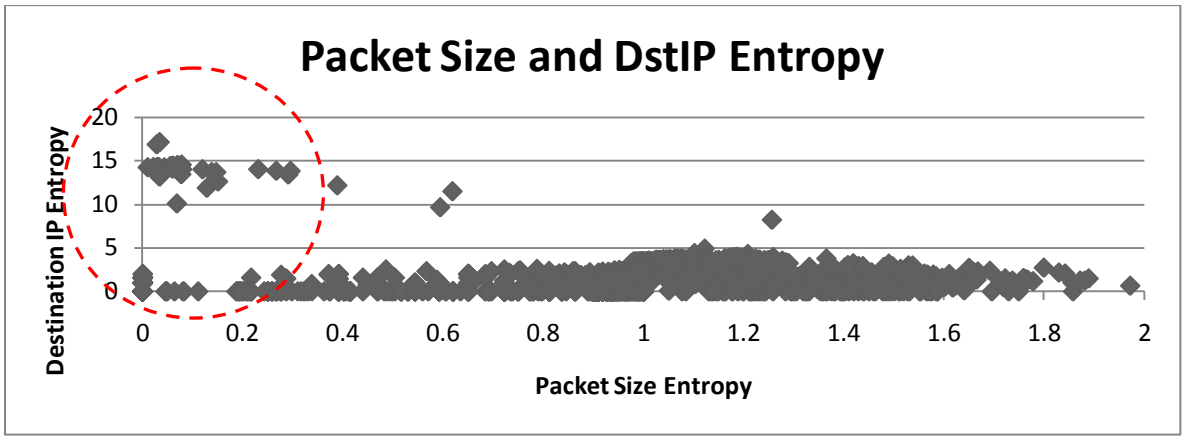


*Figure 3-27: Combination of Packet Size entropy and Destination IP entropy of SSH based honeypot trace*

The combination of packet size entropy and destination IP entropy (refer to Figure 3-27) shows that during a network scan activity (high destination IP entropy) a single packet size was mostly used which is indicated by the low packet size entropy and high destination IP entropy.

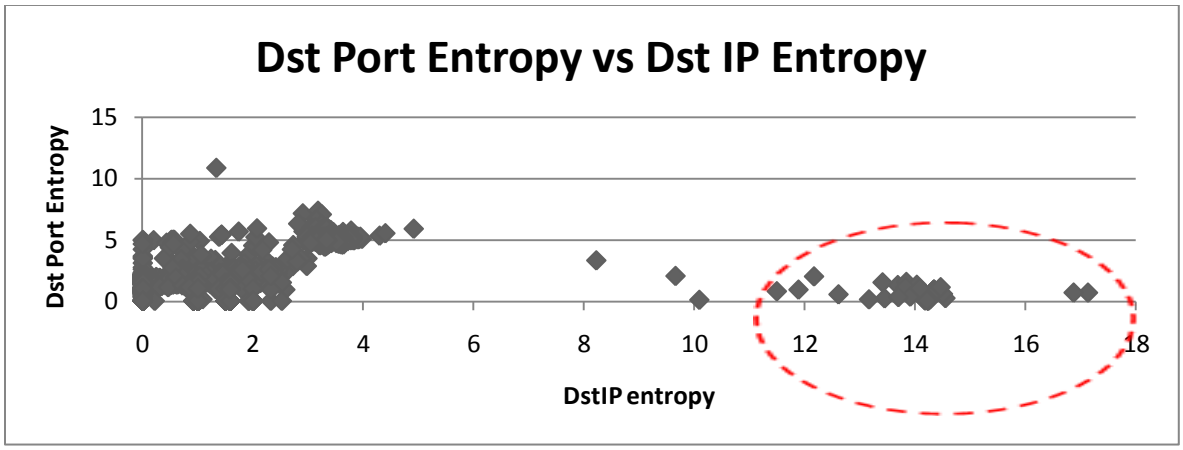


*Figure 3-28: Destination Port entropy and Destination IP entropy combination of SSH honeypot trace*

The destination IP entropy and the destination port entropy show visible groups, i.e., clusters, indicating events with similar behaviors. In Figure 3-28 the group with high

destination IP entropy and low destination port entropy indicates a network scan where a large number of IP addresses are being scanned for similar port. The point with high destination port entropy and low destination IP entropy is related to port scan activity.

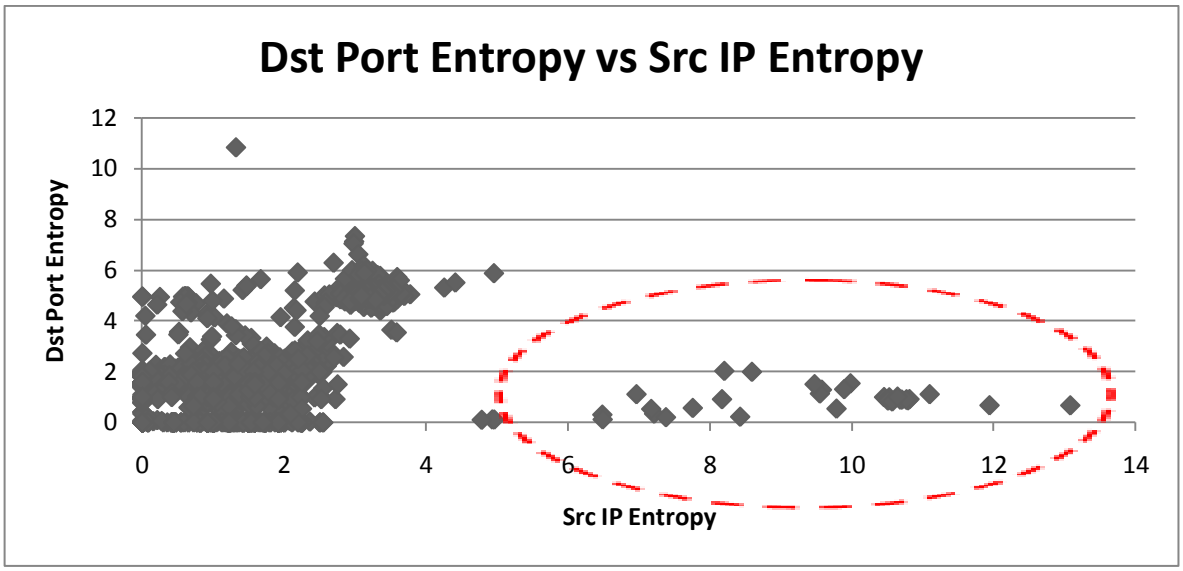


*Figure 3-29: Destination Port entropy and Source IP entropy combination of SSH honeypot trace*

The Figure 3-29 shows the combination of destination port entropy and source port entropy. This combination almost shows similar behavior as the previous combination of destination IP and destination port. The reason for having a similar behavior because the source IP entropy showed similar behavior of destination IP entropy as discussed in the previous section 3.3.4.

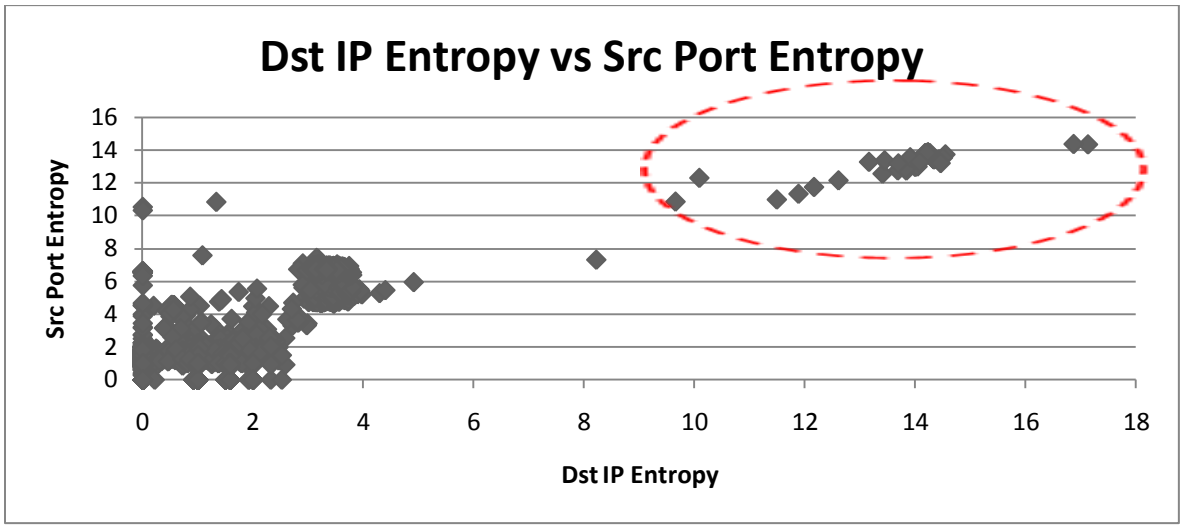


*Figure 3-30: Destination IP entropy and Source Port entropy combination of SSH honeypot trace*

The combination of source port entropy and destination IP entropy (refer to Figure 3-30) shows the regions where network scanning took place. The value of the source port entropy and destination IP entropy is high which indicates that a network scan activity happened and the source machine was using different ports with each connection. The manual analysis of this trace reveals that the honeypot was compromised and then a malicious file transferred to it, which was later used for scanning the network.

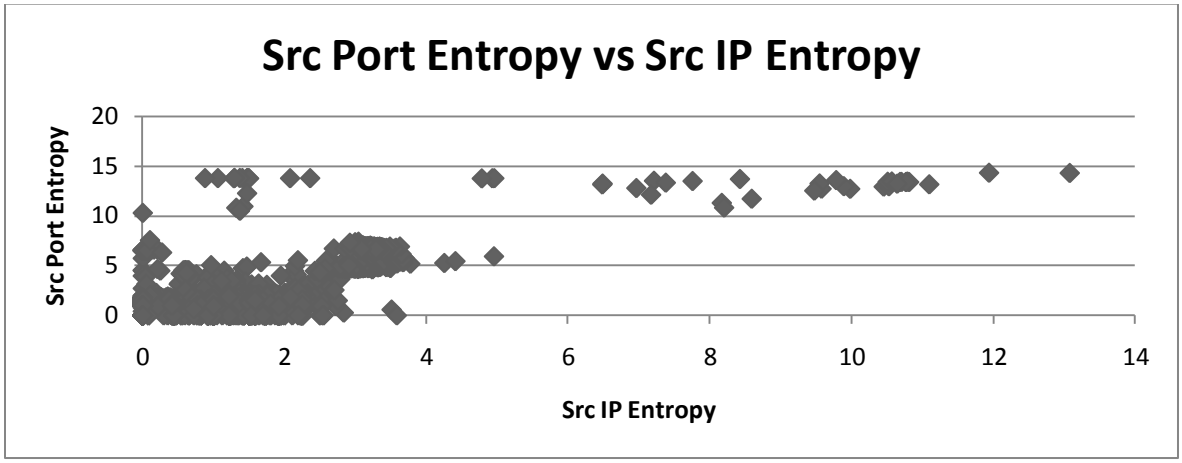


*Figure 3-31: Source IP entropy and Source Port entropy combination of SSH honeypot trace*

The combination of source IP entropy and source port entropy (refer to Figure 3-31) shows that during network scan the source IP address was less due to fact that only one IP was scanning network.

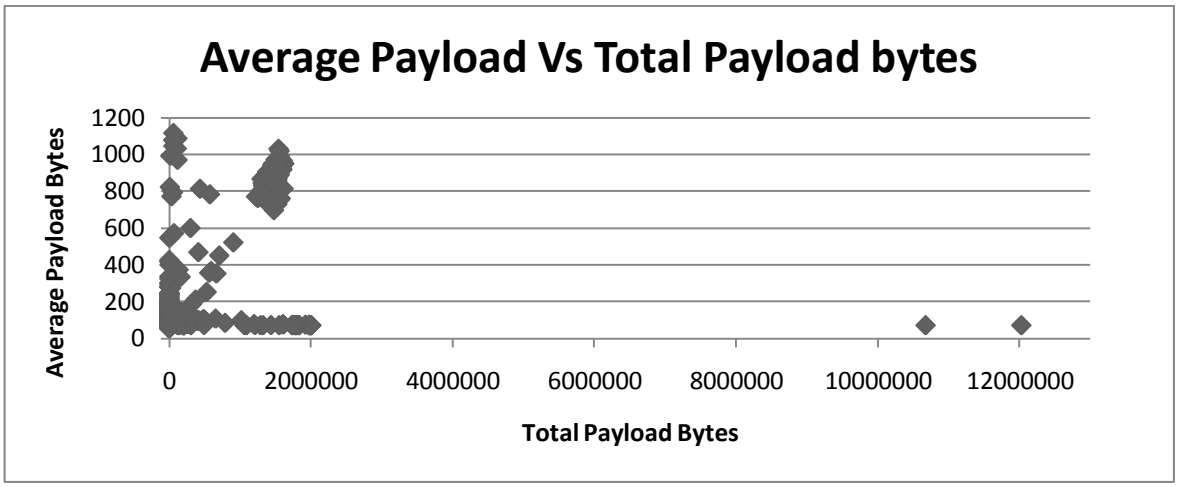


*Figure 3-32: Packet Count and Avg. Payload Size combination of SSH honeypot trace*

The combination of packet count and average payload size shows that during a network scan, a large number of packets were used with small payload sizes (refer to Figure 3-32). It also seen from the plot that large sized packets were used for transferring malicious files to the honeypot.

Based on the combinations of different features it is clear that different combination of features give better understanding of the anomaly behavior instead of just using single feature. For example it was found using combinations that during network scan anomaly, honeypot was:

- Scanning various IP addresses
- Connecting using different ports to the destination IP address
- Targeting a single port.
- Sending many packets of small size

The above findings were deduced based on the different combinations used. In next step we try to combine more than one feature (three features) to see if these separate findings can be detected by a combination of three features.

## 3.5 Combining Three Features to Detect Visible Anomalous Groups

When combining different features, we can see different patterns that can help us detect anomalous regions and normal regions. Using three features helps in getting a better visualization of the different clusters present in the honeynet data. Also, viewing features in three dimensions shows better the behavior of the anomaly as it indicates the variations of different features during such anomaly. We performed various tests using different combinations of the features to identify those features that provide the best distinction between the different clusters or groups in the analyzed data. The combination of source IP, destination IP and destination port are shown in Figure 3-33. This combination does not show many cluster regions because the source IP entropy and destination IP entropy had almost similar behavior.

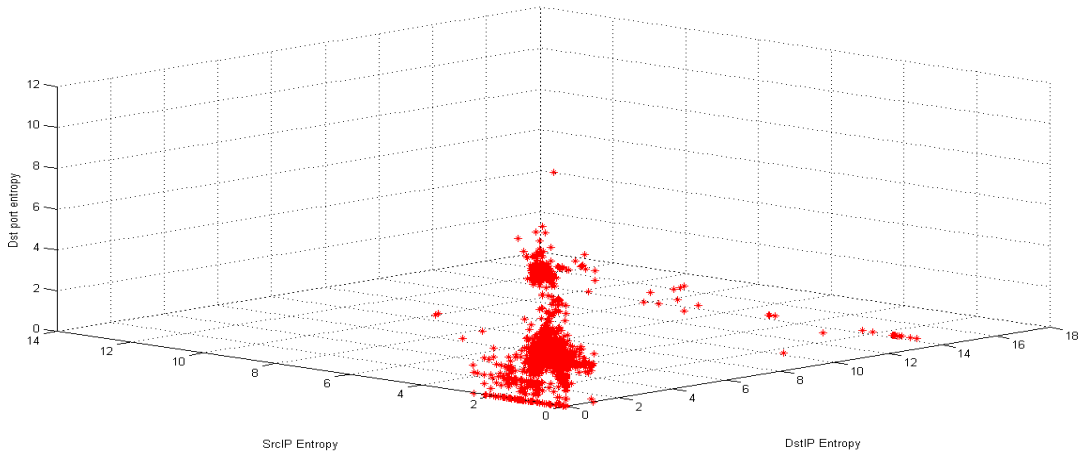


*Figure 3-33: Combination of Destination port, Source IP and Destination IP Entropy values*

The combination of destination IP entropy, source IP entropy and packet size entropy (refer to Figure 3-34) does not show much visible patterns and most of the points are concentrated close to destination IP and packet size entropy.

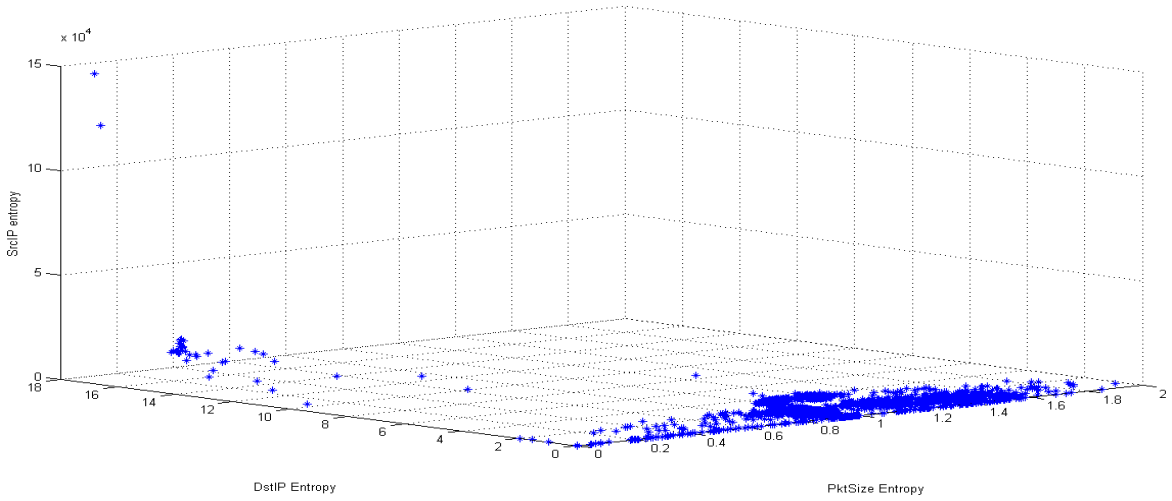


*Figure 3-34: Combination of Source IP, Destination IP and Packet Size Entropy Values*

The combination of source port entropy, source IP entropy and destination IP entropy few regions that can be related to networks scan activity and bruteforce attempts.(refer to Figure 3-35). The biggest cluster is related to the normal traffic where the entropy values are in the range of 0 to 3. But this combination does not show certain groups of events such as port scan and scanning of honeypot for open IRC channels.

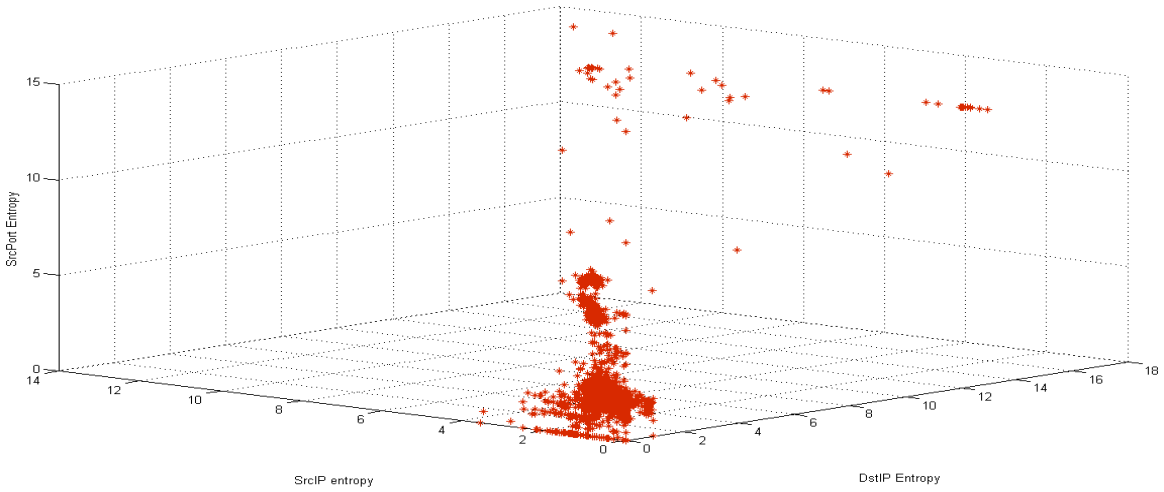


*Figure 3-35: Combination of Source IP, Source Port, Destination IP entropy values*

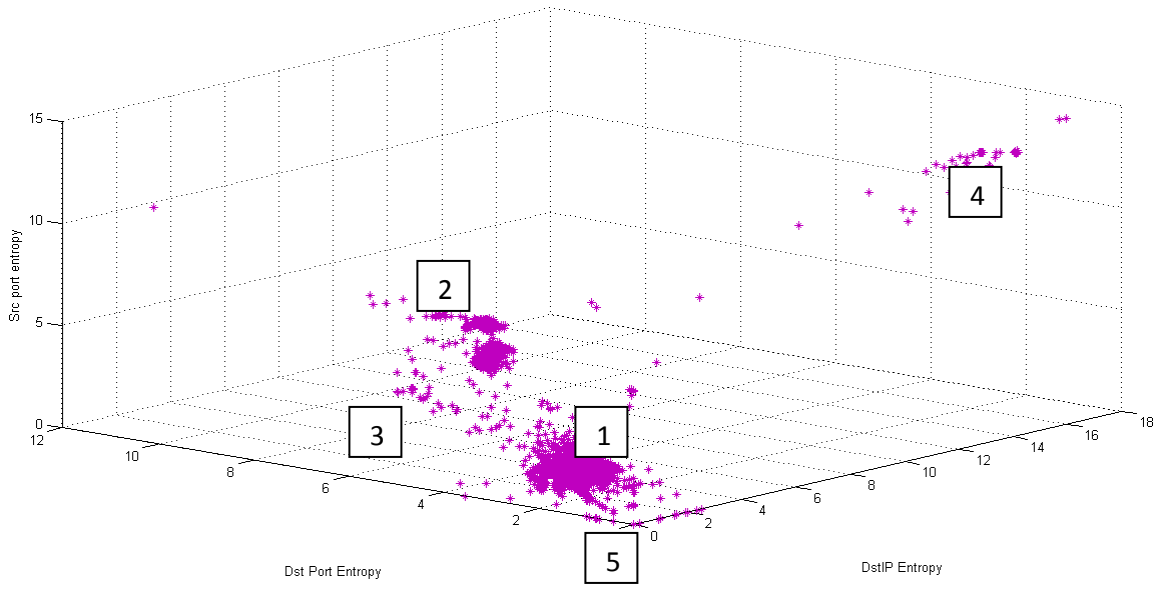


*Figure 3-36: Combination of Destination IP, Destination Port, Source port entropy values*

The combination of source port, destination port, and destination IP entropies shows visible clusters; which can be attributed to different anomalous events. In the Figure 3-36, cluster 1 in includes a region having entropy values of 0 to 2.8 for all three features. The second cluster represents the scanning by the honeypot for different IRC channels. This is based on the entropy values and the manual analysis of the trace. In this region the source

port entropy is high and the destination IP entropy is also high as the honeypot is scanning for different IP addresses.

The third cluster includes a region where there were bruteforce attempts to log into the SSH service running on the honeypot. In this region the source port is high and the destination port is low as they are targeting the SSH port. The fourth cluster indicates the network scan performed by the honeypot; which scans the SSH port on the destination machines using different ports for each connection. The region closer to zero mostly represents the IRC traffic as there one or two machine communicating with each other using the IRC ports. The detection capabilities of various features are summarized in the Table 3-4.

*Table 3-4: Summary of Detection Capabilities of various features*

| Traffic Feature | Detection Capabilities |
|---|---|
| Packet Size Entropy | Shows good variations but does not help in understanding the anomaly. |
| Destination IP Entropy | Shows large variations during specific anomalies and gives good indication of anomaly. |
| Source IP Entropy | Shows less variations in the traffic compared to destination IP entropy. |
| Destination Port Entropy | Shows large variations for various anomalies. |
| Source Port Entropy | Shows large variations for various anomalies. |
| Average Packet Inter-Arrival Time | Shows good variations but not very useful in understanding the anomaly behavior |
| Total Payload Bytes | Shows good variations during most of the anomalies and when used with other features gives good understanding of the anomaly |

| Traffic Feature | Detection Capabilities |
|---|---|
| Total Packets | Shows good variations during anomalies and very useful in understanding the anomalies. |
| Average Payload Size | Shows good variations during anomalies but does not aid in understanding the anomaly behavior. |

The features were first individually analyzed and then various combinations of features were compared to identify the best feature combinations that can differentiate between normal and abnormal traffic. After testing various combinations, we conclude that the combination of destination port entropy, source port entropy, and destination IP entropy provide better detection capabilities. On the other hand, the volume features: total payload bytes and total packets have better detection capabilities and are very useful in detecting certain types of anomalies; which are not detected by traffic features. For example, certain malicious files transferred to the honeypot were not detected by the by feature-based techniques; instead volume-based features detected these events. Therefore, instead of just looking at the feature-based techniques, we also need to use the volume-based techniques in order to detect most types of anomalies in a Honeynet.

## 3.6 Difference between Honeynet Traffic and Network Wide Traffic

The Honeynet traffic is different from any other network wide traffic since the Honeynet is not intended to receive any production traffic. Any traffic that enters or leaves the Honeynet is suspicious by nature. However, in order to identify malicious activities in the Honeynet traffic, we need to perform manual analysis and have expertise in identifying such attacks. Due the fact that honeynet traffic is different compared to traffic seen in

normal end-user machines, most of other anomaly detection approaches which rely on regular network wide traffic are not well suited for this type of traffic [20]. The honeynet systems are generally configured with unused IP addresses and generally should not be getting any traffic. Therefore, most of the traffic that honeynets will receive is due to probes or malwares or due to other problems in the network. However, traffic collected from other machines connected to the network will be getting traffic from different sources and will require different parameters and techniques to identify malicious events.

The difference between the Honeynet traffic and the network wide traffic can be understood by comparing the Honeynet traffic collected from honeypots and the DARPA 99 [28] traffic dataset. For instance, In the Honeynet traffic, most of the conversation is between the honeypot and other machines while this is not the case for the network wide traffic such as the one present in DARPA 99. Figure 3-37 shows that most of the IP conversations have the honeypot IP as a common endpoint.

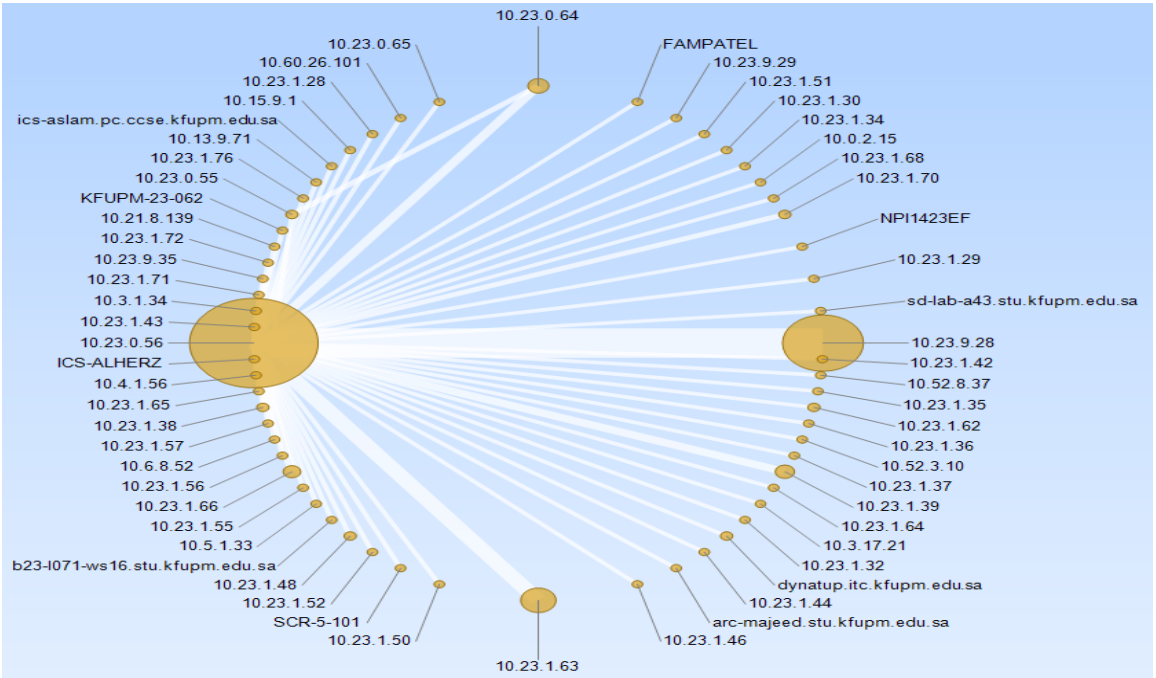


*Figure 3-37: IP conversations – Honeynet Traffic*

In contrast, the network wide traffic of DARPA 99 has many different IP addresses that are communicating with each other as seen in Figure 3-38.

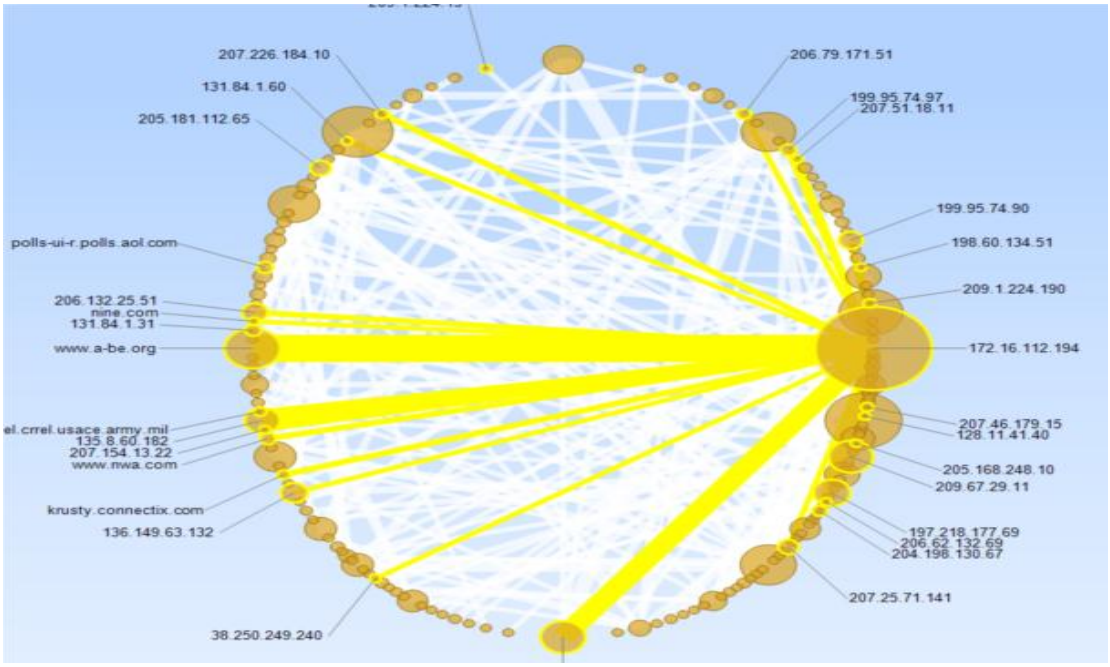


*Figure 3-38: IP conversations – DARPA99 Dataset*

In addition, comparing the entropy values for different features gives a better idea of the difference in these types of traffic. Figure 3-39 shows the normal day traffic in terms of destination port entropy in the honeynet traffic when no attacks were seen in the collected trace.

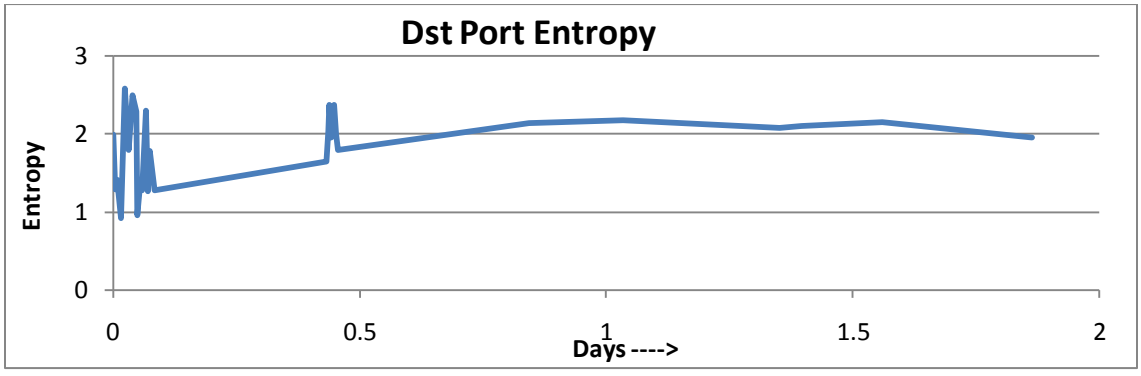


*Figure 3-39: Destination Port Entropy – Normal Honeynet Traffic*

Figure 3-40 shows the destination port entropy seen from one of the IP addresses in the network wide traffic with no attacks. This shows that even during normal traffic, period peaks are visible. This indicates that the threshold levels in these two types of traffic are different and therefore the same levels cannot be used in both cases to detect anomalies. The reason for this is that our technique considers any traffic coming to honeypots as malicious while in the production machines not all traffic is malicious.

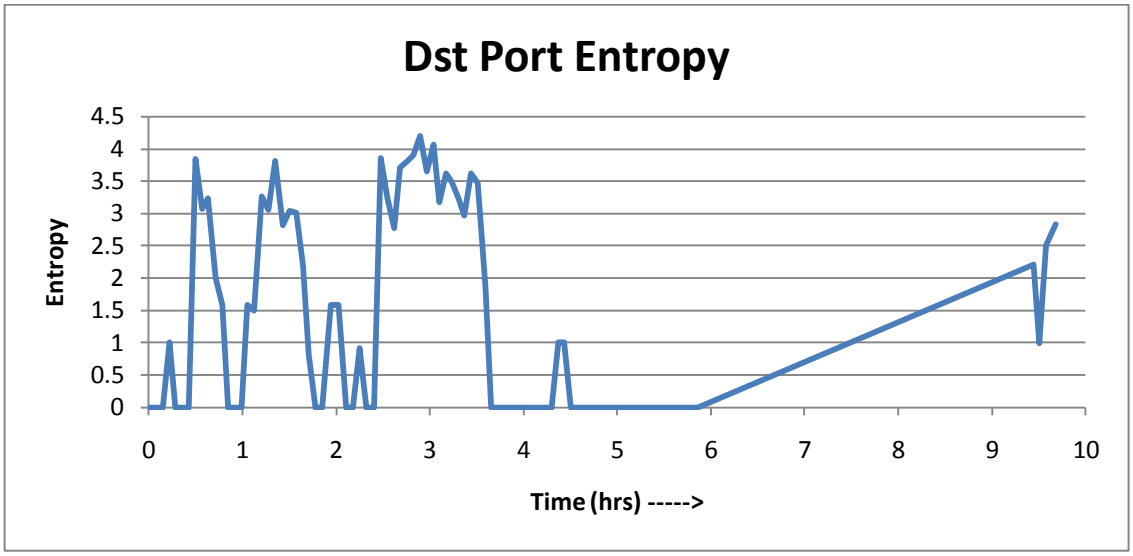


*Figure 3-40: Destination Port Entropy – Normal DARPA99 Dataset Traffic*

Figure 3-39 and Figure 3-40 also show the difference between honeynet traffic and traffic collected from normal machines. In honeynet traffic, some of the peaks shown in Figure 3-40 would indicate malicious activities, but in this case, normal traffic in the DARPA 99 dataset has peaks which could be due to some user sessions.

The network wide traffic collected during attacks also shows that different production machines in the network get different types of traffic. Figure 3-41 and Figure 3-42 show the variation of destination port entropies seen from two different IP addresses.

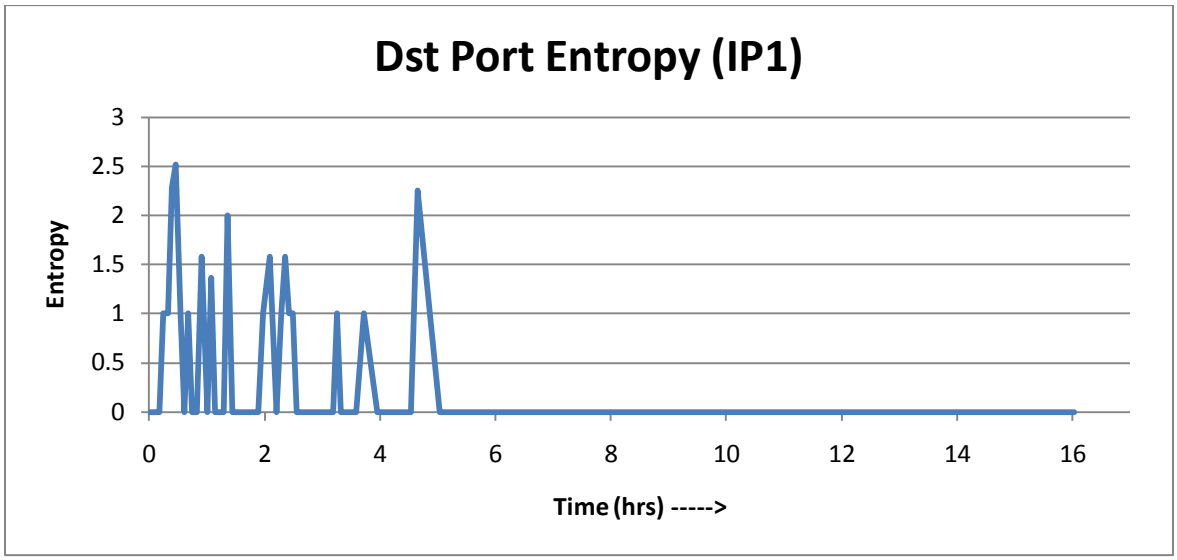


*Figure 3-41: Destination Port Entropy IP1 – Attack Traffic DARPA99 dataset*

*Figure 3-42: Destination Port Entropy IP2 – Attack Traffic DARPA99 dataset*

Figure 3-43 shows the variation of the destination port entropy for the entire trace containing attacks. The behavior of the network traffic is different compared to the traffic of individual machines. Therefore, to detect anomalies in this type of traffic, a different algorithm has to be designed; and our proposed technique is not suitable for such traffic types.

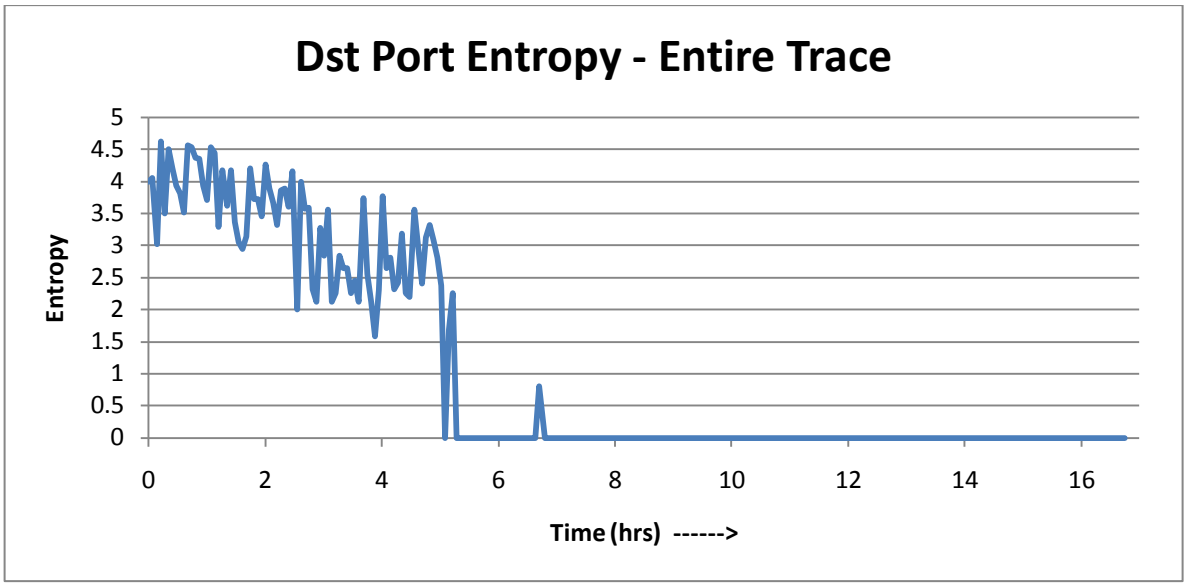


*Figure 3-43: Destination Port Entropy – Entire Trace: Attack Traffic DARPA99 dataset*

# 3.7 Detecting Anomalies

Anomalies are generally identified by distinguishing between normal and abnormal behavior. In entropy based systems, small variation in the values of traffic feature results in low entropy and large variation results in high entropy value. Detecting anomalies also involves the tracking of significant changes in the traffic volume. It is also identified that tracking significant variation in entropy is very useful to identify the presence of an anomaly in the traffic dataset [29].

In our proposed approach, anomalies are identified using the five selected features. The classification between normal and abnormal traffic is done using the variations in the corresponding features. For example in Honeynet network during normal behavior very less variations are seen (Refer Table 3-5). However there are significant traffic changes during the presence of anomalies. (Refer to Table 3-6) Based on a through manual analysis of the test data sets we found that during normal traffic, entropy based features

had an entropy variation in the range of 0 and 3. Similarly for volume-based features variations were in the range of 0 to 3000 bytes for total payload bytes and 0 to 50 packets for total packet count.

*Table 3-5: Normal Traffic Values Range*

| DIP | DP | SP | TB | PC |
|---|---|---|---|---|
| 0 | 1.31 | 1.31 | 228 | 6 |
| 1 | 1.52 | 0.98 | 444 | 4 |
| 0 | 1.87 | 2.04 | 2631 | 20 |
| 1 | 0.918 | 1.58 | 3 | 1 |
| 0 | 1.62 | 0.33 | 168 | 8 |

*Table 3-6: Abnormal Traffic Values Range*

| DIP | DP | SP | TB | PC |
|---|---|---|---|---|
| 0.52 | 3.56 | 4.46 | 12118 | 152 |
| 0 | 3.22 | 4.20 | 13971 | 138 |
| 0 | 3.37 | 3.61 | 70497 | 185 |
| 17.14 | 0.67 | 14.33 | 141048 | 5702 |
| 16.87 | 0.677 | 14.36 | 181988 | 7023 |
| 0.419 | 11.55 | 11.53 | 374099 | 4152 |
| 0.218 | 12.26 | 12.26 | 214096 | 5374 |

Apart from the anomaly detection threshold the entropy values and volume changes can be used to define various thresholds levels. These levels can be used for identifying the behavior of anomalies in the Honeynet traffic.

**Very High Entropy or Very High Volume**: This level is used for high entropy values and high volume of data. Based on the tests made on the traces only few anomalies (network Scan and Port Scan) had high entropy values. The entropy values greater than 7

are considered as very high entropy. Volume changes greater than 500KB and packet count greater than 2000 packets are considered very high volume change.

**High Entropy and High Volume**: This level is used for entropy values that lie between 5 and 7. Based on the experimental results it can be understood that certain anomaly types like bruteforce attacks or fuzzers result in high entropy values. The reason for certain anomalies to have high entropy is due to fact that they initiate too many connections from different ports to crack the passwords or the vulnerability of different applications. Volume changes between 50kb and 500kb as well as packet count between 500 to 2000 packets are considered high volume change.

**Medium Entropy and Medium Volume**: This is used for entropy values that are greater than the normal range and less than the high entropy. The entropy values that lie between 3 and 5 are considered medium. Most of anomalies lie in this range as they cause enough changes in the entropy values to cross the normal range. The reason is that most of anomalies target specific ports and hence make do not require port scans and hence the entropy values are slightly less compared to high entropy. Volume changes between 3kb and 50kb as well as packet count between 50 to 500 packets are considered medium volume change.

**Zero entropy value**: This entropy value is used for cases during which only one dominant feature value is present in the trace. For example if only one destination IP is visible during the five minute interval then an entropy value of zero is recorded. This level is used only for feature-based parameters and not applicable to volume-based parameters. Also the situation in which this level is considered anomaly is when there is zero entropy for the three feature-based parameters and there is medium volume change.

# 3.8 Anomaly Behavior Detection

It is essential to analyze the behavior of the various anomalies detected in the training data sets after having already defined the needed features to detect such anomalies. Such an analysis of the behavior of the various anomalies detected will help in recommending a behavior pattern for a detected anomaly type. Hence, this section presents the various entropy value ranges that were recorded for different anomaly types found in the training data sets. These entropy values were used to understand the behavior of the anomalies and to recommend a behavior pattern for a detected anomaly type. The behavior pattern is recommended using the five features that were selected for anomaly detection. The entropy values that were recorded in the presence of an anomaly are presented in Table 3-7 to

Table 3-10. It is presented as a range because the anomaly is recorded over multiple intervals and hence produces many entropy values during the anomaly. The entropy values for malicious file download as they did not show any significant changes in all the case and were in the normal region; hence they are omitted in tables below and are marked with X.

*Table 3-7: Anomalies detected in Scan 28 Trace*

| Anomaly Type | Dst IP Entropy | Dst Port Entropy | Source Port entropy | Total Payload Bytes | Total Packet Count |
|---|---|---|---|---|---|
| System Compromise | 0 to 2.2 | 2.02 to 2.988 | 2.02 to 3.11 | 4547 to 742346 | 22 to 1491 |
| Malicious file download | X | X | X | 392336 to 742346 | 753 to 1491 |
| IRC | 0 to 2.5 | Many 0 | Many 0 | 6200 to 19048 | 10 to 97 |

| | | | | | |
|---|---|---|---|---|---|
| communication Day 1 | | points 1 to 2.5 | points 1 to 2.6 | | |
| IRC Communication Day 3 | 0 | 0 | 0 | 1657 to 8652 | 15 to 75 |
| ICMP (DDos) | 0.721 to 3.4 | 0 to 1.38 | 0 to 1.63 | 6348 to 16177 | 6 to 58 |
| Port Scan / Dos Attack Day 1 | 0 to 0.91 | 7.09 to 8.685 | 6.95 to 9.81 | 153112 to 764302 | 406 to 3197 |
| Port Scan/Dos Day 3 | 0 to 0.39 | 4.99 to 7.424 | 5.29 to 9.61 | 56066 to 169238 | 674 to 2773 |

*Table 3-8: Anomalies detected in Scan 14 trace*

| Anomaly Type | Dst IP entropy | Dst Port Entropy | Source Port entropy | Total Payload Bytes | Total Packet Count |
|---|---|---|---|---|---|
| System Compromise | 0 to 1.84 | 3.15 to 3.56 | 3.065 to 4.465 | 12118 to 13971 | 138 to 152 |
| Malicious File download | X | X | X | 16805 to 70497 | 145 to 185 |
| Running various commands | 0 to 1.95 | 0 to 2.71 | 1 to 3.85 | 1312 to 8638 | 45 to 116 |

*Table 3-9: Anomalies detected in Scan 19 trace*

| Anomaly Type | Dst IP entropy | Dst Port Entropy | Source Port entropy | Total Payload Bytes | Total Packet Count |
|---|---|---|---|---|---|
| System compromise | 0.9893 | 1.8078 | 2.159 | 1191 to 13145 | 33 to 102 |
| Malicious file download | X | X | X | 374099 | 4152 |
| Port scan | 0.218 to 0.419 | 11.5 to 12.263 | 11.53 to 12.26 | 214096 | 5374 |

*Table 3-10: Anomalies detected in SSH based Honeypot trace*

| Anomaly Type | Dst IP entropy | Dst Port Entropy | Source Port entropy | Total Payload Bytes | Total Packet Count |
|---|---|---|---|---|---|
| ICMP flood | 1.584 | 0 | 0 | 14372 to 56636 | 14 to 55 |
| Malicious file download | X | X | X | 103512 to 1271603 | 1727 |
| Network Scan | 16.87 to 17.14 | 0.0370.67 | 14.3 | 117906 to 10677114 | 1603 to 163519 |
| Bruteforce | 0 to 1.4 | 0 to 4.39 | 3.98 to 6.53 | 29680 to 76402 | 494 to 1947 |
| IRC Conversations | 0 to1.58 | 0 to 1.79 | 0 to 1.78 | 26263 to 10660 | 229 to 249 |
| Port scan | 2.00 to 4.91 | 4.51 to 5.877 | 3.66 to 5.94 | 15289 | 154 |
| Honeypot Scanning for IRC channels | 3.33 to 3.91 | 4.54 to 5.33 | 4.47 to 7.24 | 17505 to 175470 | 254 to 2566 |

### 3.8.1 Anomaly Behavior

The anomaly behavior is recommended based on the values detected for the selected features. Based on the analysis of the various training data sets that will be presented next it was found that not all the features are required to define the behavior of all the anomalies. Certain anomalies can be defined using just two or three features while others require all the features. The reason for this is that certain anomalies such as the ICMP flood anomaly are independent of specific features such as port entropies that do not pertain to such anomalies. Accordingly, certain features have values in the normal range in all instances of the same anomaly in different training data sets due to which they do not aid in identifying that anomaly.

The following set of tables summarizes the analysis of the behavior of all the anomalies based on the various training data sets. It should be noted that the feature that was considered less important to define the behavior of the anomaly is grayed out in the corresponding tables. The values recorded for the system compromise event from the different training data sets is shown in Table 3-11. Based on the recorded values it can be concluded that the behavior of the system compromise anomaly is Medium Destination Port Entropy, Medium Source Port Entropy, High Total Payload Bytes, and Medium Total Packet Count. In this case, the destination IP entropy is less significant because during the system compromise there is only one target machine being exploited and hence there is no significant change in the entropy values.

*Table 3-11: Anomaly Type: System Compromise*

| Trace | Dst IP entropy | Dst Port Entropy | Source Port entropy | Total Payload Bytes | Total Packet Count |
|---|---|---|---|---|---|
| Scan 28 | 0 to 2.2 | 2.02 to 2.988 | 2.02 to 3.11 | 4547 to 742346 | 22 to 1491 |
| Scan 14 | 0 to 1.84 | 3.15 to 3.56 | 3.065 to 4.465 | 12118 to 70497 | 138 to 185 |
| Scan 19 | 0.9893 | 1.8078 | 2.159 | 1191 to 13145 | 33 to 102 |
| SSH-based Honeypot | 1.222 | 2.089 | 2.077 | 343184 | 385 |

Table 3-12 shows the values recorded for malicious file downloads in different training data sets. Based on these values the behavior of malicious file download can be defined as Very High Total Packet Bytes, High Packet Count. Note that the entropy values are omitted from Table 3-12 as they did not show any significant changes in the different training data sets and were in the normal range. The reason for this is that during the

malicious file downloads; there is no significant change in the entropy values since most of the communication occurs between two machines using specific ports, i.e., FTP, HTTP, etc.

*Table 3-12: Anomaly Type: Malicious File Download*

| Trace | Dst IP entropy | Dst Port Entropy | Source Port entropy | Total Payload Bytes | Total Packet Count |
|---|---|---|---|---|---|
| Scan 28 | | | | 392336 to 742346 | 753 to 1491 |
| Scan 14 | | | | 16805 to 70497 | 145 to 185 |
| Scan 19 | | | | 374099 | 4152 |
| SSH-based Honeypot | | | | 103512 to 1271603 | 1727 |

Table 3-13 shows the values recorded during the IRC communication noticed in the different training data sets. Based on these values the behavior of the IRC communications can be defined as Zero Destination IP entropy, Zero Destination Port entropy, Zero Source Port entropy, Medium Total Payload bytes, and Medium Total Packet Count.

*Table 3-13: Anomaly Type: IRC communication*

| Trace | Dst IP entropy | Dst Port Entropy | Source Port entropy | Total Payload Bytes | Total Packet Count |
|---|---|---|---|---|---|
| Scan 28 | 0 to 2.5 | Many 0 points 1 to 2.5 | Many 0 points 1 to 2.6 | 6200 to 19048 | 10 to 97 |
| SSH based honeypot | 0 to 1.58 | 0 to 1.79 | 0 to 1.78 | 26263 to 10660 | 229 to 249 |
| Scan 28 | 0 | 0 | 0 | 1657 to 8652 | 15 to 75 |

Table 3-14 shows the various values recorded during the ICMP flood anomaly. The values indicate the anomaly behavior as High Total Payload Bytes and Medium Total Packet Count. The reason that this anomaly does not cause any changes to port entropies is that ICMP is a layer 3 protocol and does not include the ports that are used by layer 4 protocols.

*Table 3-14: Anomaly Type: ICMP flood*

| Trace | Dst IP entropy | Dst Port Entropy | Source Port entropy | Total Payload Bytes | Total Packet Count |
|---|---|---|---|---|---|
| Scan 28 | 0.721 to 3.4 | 0 to 1.38 | 0 to 1.63 | 6348 to 16177 | 6 to 58 |
| SSH based honeypot | 1.584 | 0 | 0 | 14372 to 56636 | 14 to 55 |

Table 3-15 shows the values recorded during the port scan anomaly. Based on the values the behavior of port scan anomaly can be defined as Very High Destination Port Entropy, Very High Source Port entropy, High Total Payload Bytes, and Very High Packet Count. Since this anomaly basically scans the ports on the target machine, it is independent from the Destination IP entropy.

*Table 3-15: Anomaly Type: Port Scan*

| Trace | Dst IP entropy | Dst Port Entropy | Source Port entropy | Total Payload Bytes | Total Packet Count |
|---|---|---|---|---|---|
| Scan 28 | 0 to 0.91 | 7.09 to 8.685 | 6.95 to 9.81 | 153112 to 764302 | 406 to 3197 |
| Scan 28 | 0 to 0.39 | 4.99 to 7.424 | 5.29 to 9.61 | 56066 to 169238 | 674 to 2773 |
| Scan 19 | 0.218 to 0.419 | 11.5 to 12.263 | 11.53 to 12.26 | 214096 | 5374 |
| SSH based trace | 2.00 to 4.91 | 4.51 to 5.877 | 3.66 to 5.94 | 15289 | 154 |

Table 3-16 shows the variation of different features during the network scan anomaly. Based on the recorded values the network scan behavior can be defined as Very High Destination IP entropy, Very High source Port Entropy, High Total Payload Bytes, and Very High Total Packet Count. Note that the network scan involves the scanning of a large number of IP addresses, and, therefore, it is independent of the Destination Port entropy.

*Table 3-16: Anomaly Type: Network Scan*

| Trace | Dst IP entropy | Dst Port Entropy | Source Port entropy | Total Payload Bytes | Total Packet Count |
|---|---|---|---|---|---|
| SSH based honeypot | 16.87 to 17.14 | 0.037 to 0.67 | 10.97 to 14.3 | 117906 to 10677114 | 1603 to 163519 |

Table 3-17 shows the variation of the different parameters recorded during the bruteforce anomaly. Based on these values the behavior of the bruteforce anomaly can be defined as Medium Destination port entropy, High Source Port Entropy, High Total Payload Bytes, and Medium Total packet count. During bruteforce attempts, most of the communication occurs between two machines and hence it does not cause significant changes in the destination IP entropy.

*Table 3-17: Anomaly Type: Bruteforce*

| Trace | Dst IP entropy | Dst Port Entropy | Source Port entropy | Total Payload Bytes | Total Packet Count |
|---|---|---|---|---|---|
| SSH based honeypot | 0 to 1.4 | 0 to 4.39 | 3.98 to 6.53 | 29680 to 76402 | 494 to 1947 |

Table 3-18 lists the various anomalies and their associated behavior in terms of different features. Indentifying the behavior of different anomalies will help in detecting similar anomalies in other data sets. Using a large number of data sets will help in defining the behavior of the anomaly better. This information can then be used to detect similar anomalies by comparing the behavior to the listed behavior.

*Table 3-18: Behavior of Different Anomaly Types*

| Anomaly | Dst IP entropy | Dst Port Entropy | Source Port entropy | Total Payload Bytes | Total Packet Count |
|---|---|---|---|---|---|
| System Compromise | | M | M | H | M |
| Malicious File Download | | | | VH | H |
| IRC communications | Z | Z | Z | M | M |
| ICMP flood | | | | H | M |
| Port Scan | | VH | VH | H | VH |
| Network Scan | VH | | VH | H | VH |
| BruteForce | | M | H | M | H |

# 3.9 Summary

This chapter provided detailed information on the different data sets used for testing the features. Various features were tested for the detection capability and then the best features that provided better detection capability were shortlisted. The combinations of different features were also tested to identify the behavior of the anomalies and based on the results the best combination was identified. The values recorded for different features were then utilized to define the anomaly behavior.

# *Chapter 4*

# Results

This chapter focuses on the results and discussions of the detection technique proposed in Chapter 3. The main sections of this chapter include the results obtained from the traces that were used to test the proposed technique and to determine its efficiency.

## 4.1 Experimental Setup

The evaluation of the proposed technique was carried out by developing Java code for extracting entropy values from the traces. The jNetPcap Java API was used for developing the code to read the PCAP trace files and then the entropy values for every five minute interval were calculated for different features. The results were then plotted using the features mentioned in Chapter 3. The traffic features that were selected were: Destination IP Entropy (DIP), Destination Port Entropy (DP), and Source Port Entropy (SP). The volume features that were selected are: Total Payload Bytes (TB), Total Packets Count (PC).

The trace files that were used for obtaining the results are:

- SSH based honeypot: trace taken from the Information Security Visualization Contest - hack.lu 2009 [26]
- Scan 27: Honeynet.org Scan of the Month Challenge, March 2003.
- Trace having Synthetic Anomalies: Anomalies inserted using penetration testing tools provided in BackTrac 4.1 [30]
- Dionaea capture trace: this trace was collected from the KFUPM network.

# 4.2 Description of Traces Used

## 4.2.1 SSH Honetpot Trace

SSH-based honeypot trace was collected from the Information Security Visualization Contest - hack.lu 2009 [26]. The details of this trace are summarized in Table 4-1.

*Table 4-1: SSH based honeypot trace details*

| Attributes | Values |
|---|---|
| Trace Name | SSH Honeypot Trace |
| Source | Information Security Visualization Contest - hack.lu 2009 |
| Duration | 12 days |
| Format | PCAP trace file |
| Size | 708 Megabytes |
| Number of Packets | 4323191 |
| Average Packet Size | 155.68 Bytes |

Figure 4-1 represents the total packets distribution of different transport layer protocols which shows a domination of TCP packets. As this trace is from an SSH based honeypot most of the packets were SSH packets as shown in Figure 4-2. This trace also has a large number of IRC packets which indicates that an IRC channel was running in the honeypot.

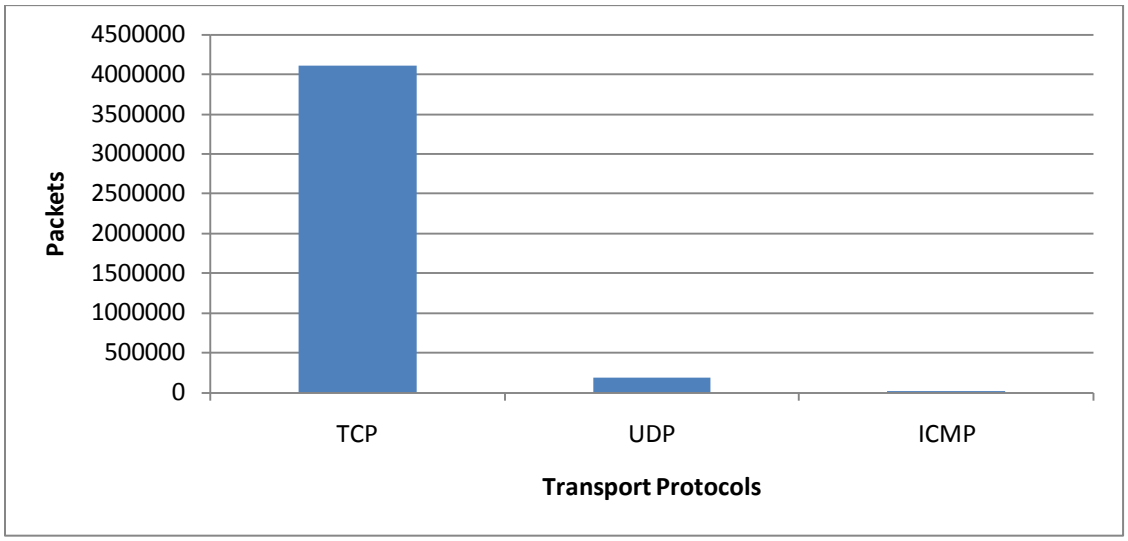


*Figure 4-1: Packet Distributions of Transport Protocols*

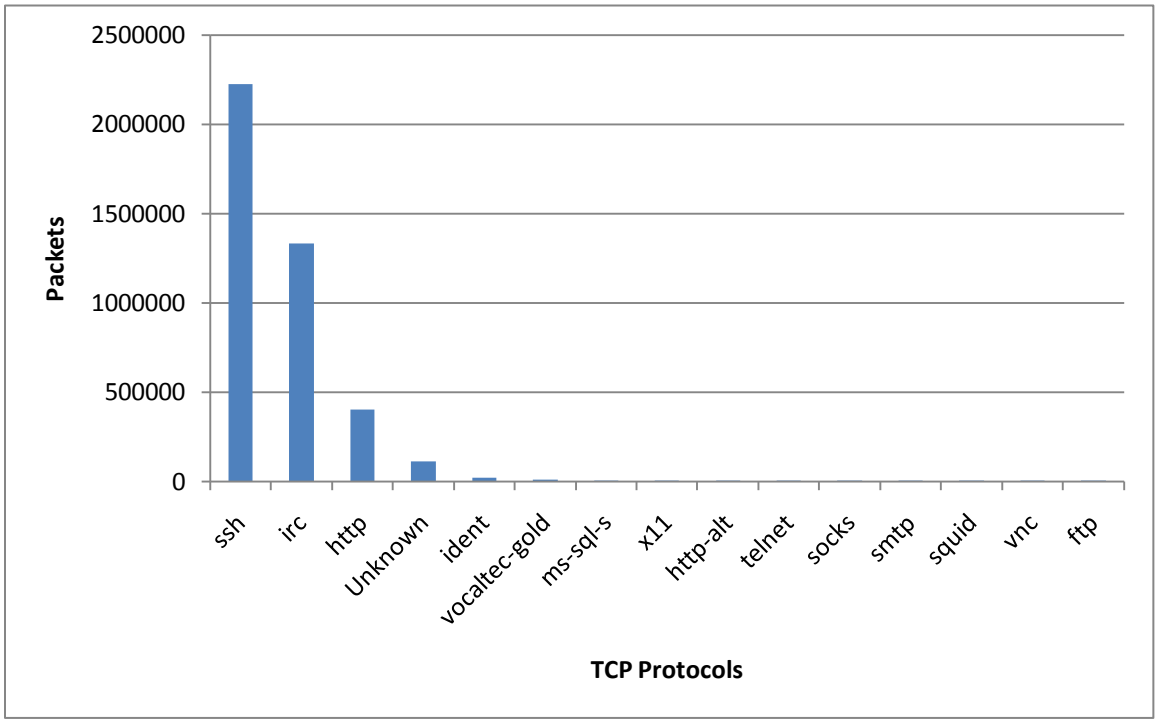


*Figure 4-2: Packets Distribution of TCP Protocols*

The results were collected by running the Java code to extract the entropy values for the selected features and then 3-D plots were generated to determine the anomalies.

### 4.2.2 Scan 27 Honeynet.org: Scan of Month Challenge

This trace was collected from Honeynet.org which releases the Scan of the Month Challenge [25]. This trace was collected by the Azusa Pacific University Honeynet Project team from an un-patched Windows 2000 honeypot. The details of this trace are provided in Table 4-2.

*Table 4-2: Scan 27 trace details*

| Attributes | Values |
|---|---|
| Trace Name | Scan 27 |
| Source | Honeynet.org, Scan of the Month Challenge |
| Duration | 5 Days |
| Format | PCAP trace file |
| Size | 17.6 Megabytes |
| Number of Packets | 54536 |
| Average Packet Size | 316.25 Bytes |

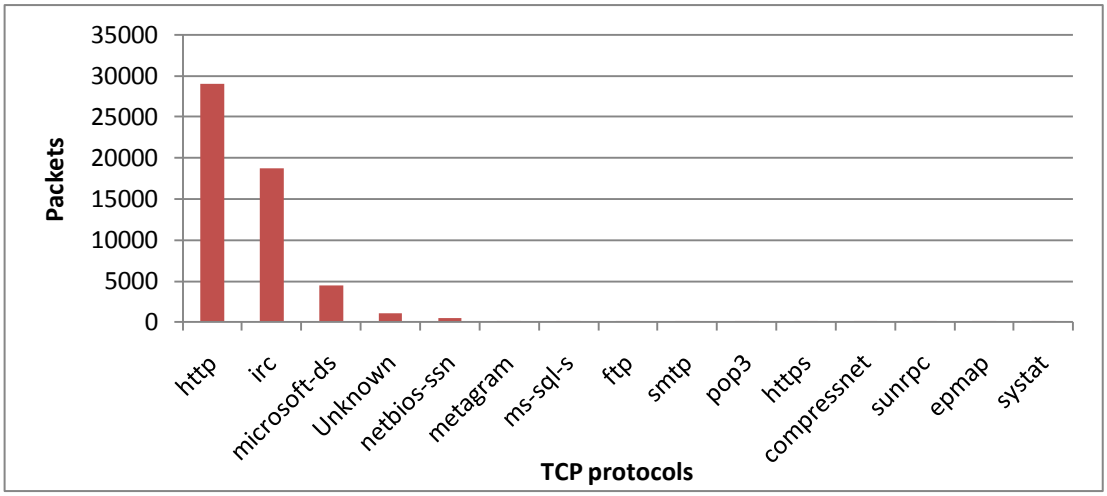


*Figure 4-3: Packet Distribution of TCP protocols*

The trace was dominated by HTTP and IRC packets as shown in Figure 4-3.

### 4.2.3 Lab Trace with Synthetic Anomalies

The third trace that was used for evaluating the detection technique was generated in the Lab setup within KFUPM. A honeynet was setup with Honeywall - a high interaction honeypot and Windows XP honeypot. The BackTrack 4.1 operating system was used as the attacker machine, which was used to attack the windows XP honeypot with different types of attacks. The honeypot was made visible on the network and popular services were activated on it such as IIS web server, FTP server, SSH server etc. The main tools that were used from the BackTrack operating system were:

- Nmap
- Open VAS vulnerability scanner
- Metasploit Penetration Testing Framework 3.0

Metasploit Framework [31] is one of the most popular open source penetration testing tools that are available in the market [32].

We used these tools to generate a trace that includes different types of malicious activities and then used our technique to test whether it can detect these anomalies. Metasploit framework has been used by other authors to generate a similar data set for their anomaly detection techniques. Laskov and Kloft [33] have used the metasploit framework to create a malicious dataset by generating various exploits from the tool. Rieck, and Laskov [34] have also used the metasploit framework to create a malicious dataset. They used various exploits from this framework which are shown in Figure 4-4. Düssel et al [35] also used the metasploit framework to generate malicious dataset for testing their anomaly detection technique.

| HTTP attacks | FTP attacks | SMTP attacks |
|---|---|---|
| HTTP tunnel | 3COM 3C exploit | CMAIL Server 2.3 exp. |
| IIS 4.0 HTR exploit | GlobalScape 3.x exploit | dSMTP 3.1b exploit |
| IIS 5.0 printer exp. | Nessus FTP scan | MS Exchange 2000 exp. |
| IIS unicode attack | ProFTPd 1.2.7. exploit | MailCarrier 2.51 exploit |
| IIS 5.0 WebDAV exp. | Serv-U FTP exploit | Mail-Max SMTP exploit |
| IIS w3who exploit | SlimFTPd 3.16 exploit | Nessus SMTP scan |
| Nessus HTTP scan | WarFTPd 1.65 exp. 1 | NetcPlus Server exploit |
| PHP script attack | WarFTPd 1.65 exp. 2 | Personal Mail 3.x exploit |
| | WsFTPd 5.03 exploit | Sendmail 8.11.6 exploit |
| | WU-FTPd 2.6.1 exploit | |

*Figure 4-4: Exploits used for generating malicious dataset [34]*

The attacks that were generated in our experiment are listed in Table 4-3.

*Table 4-3:Attacks generated against the honeypot*

| Categories | Types of Attacks |
|---|---|
| Port Scan | NMAP regular scan<br>NMAP quick scan<br>NMAP intense scan<br>NMAP slow comprehensive scan |
| Vulnerability Scanning | Open VAS Scanner |
| Database attacks | MYSQL login utility scanner<br>MYSQL database access attempts |
| Server Message Block (SMB) protocol attacks | SMB Negotiate Dialect Corruption<br>(Fuzzers/smb/smb_negotiate_corrupt)<br>Microsoft Workstation Service<br>NetAddAlternateComputerName Overflow<br>Microsoft Server Service Relative Path Stack Corruption<br>Microsoft Server Service NetpwPathCanonicalize Overflow<br>Microsoft Plug and Play Service Overflow<br>Microsoft Print Spooler Service Impersonation Vulnerability |
| DCE/RPC, (Distributed | Endpoint Mapper Service Discovery |

| Categories | Types of Attacks |
|---|---|
| Computing Environment / Remote Procedure Calls) attacks | (scanner/dcerpc/endpoint_mapper)<br>DCERPC TCP Service Auditor<br>Microsoft RPC DCOM Interface Overflow exploit<br>Microsoft Message Queueing Service Path Overflow exploit |
| FTP | Simple FTP Fuzzer<br>FTP attack access gain attempt |
| HTTP IIS web server attacks | Microsoft IIS WebDAV Writ exploit<br>Microsoft IIS 5.0 Printer exploit<br>Microsoft IIS/PWS CGI Fil exploit<br>Microsoft IIS 5.0 WebDAV ntdll.dll Path Overflow |
| SMTP attacks | MS03-046 Exchange 2000 XEXCH50 Heap Overflow exploit |
| SNMP attacks | Network Node Manager Snmp.exe CGI Buffer Overflow |
| Backdoor | Energizer DUO Trojan Code Execution |
| SSH attacks | SSH Key Exchange Init Corruption |

Table 4-4 summarizes the trace details that were used to validate the results.

*Table 4-4: Lab – Trace Details*

| Attributes | Values |
|---|---|
| Trace Name | Lab -Trace |
| Source | Generated in lab environment in a High Interaction honeypot setup |
| Duration | 22 Days |
| Format | PCAP trace file |
| Size | 30 Megabytes |
| Number of Packets | 312599 |
| Average Packet Size | 85.181 bytes |

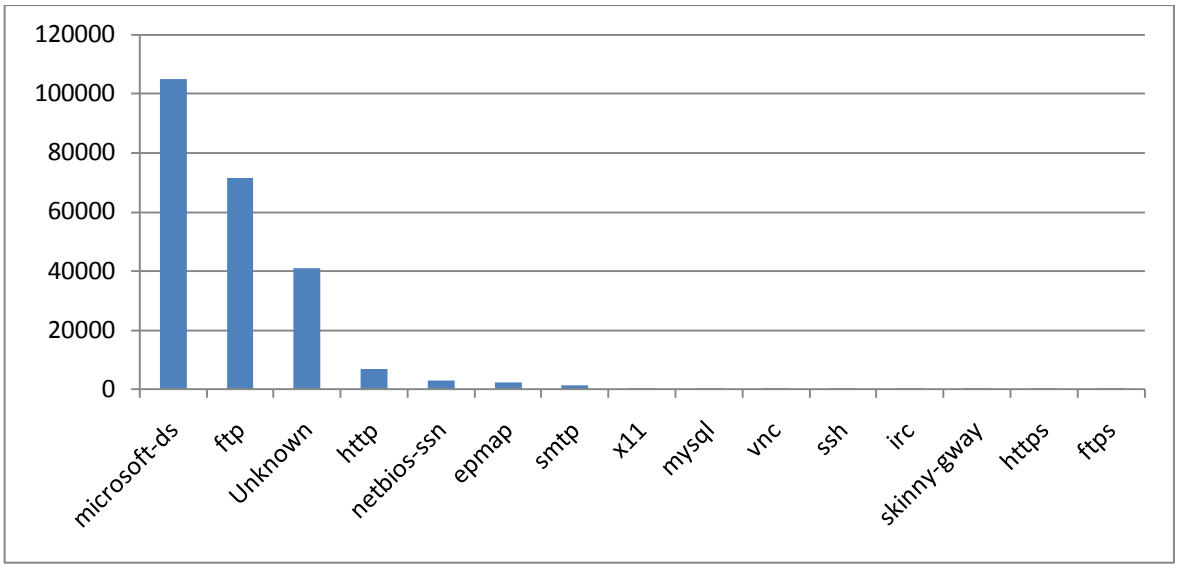


*Figure 4-5: Packets Distribution of TCP Protocols*

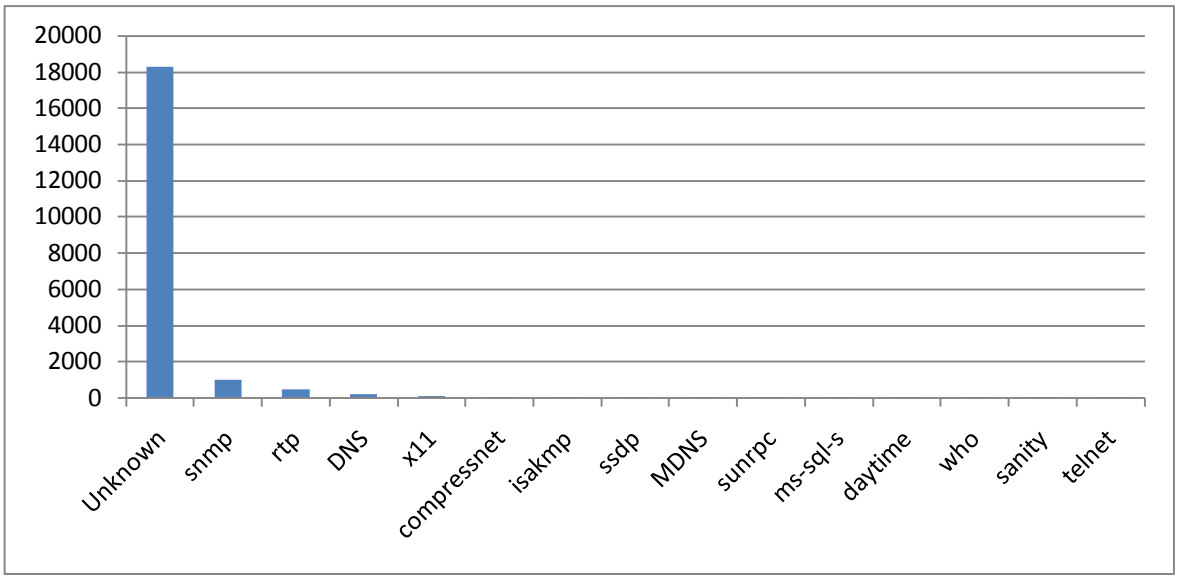


*Figure 4-6: Packet Distribution of UDP Protocols*

Figure 4-5 and Figure 4-6 show the packet distributions of TCP and UDP protocols. The plots show the different types of applications that were targeted in the honeypot.

### 4.2.4 Dionaea Capture Trace

A low interaction honeypot Dionaea was setup and connected in the KFUPM network. The trace was collected for two separate days and our proposed anomaly detection technique was used to detect anomalies in the traffic. The trace details are as given in Table 4-5.

*Table 4-5:Dionaea capture trace details*

| Attributes | Values |
|---|---|
| Trace Name | Dionaea Capture Trace |
| Source | Low interaction honeypot setup in KFUPM |
| Duration | 2 days |
| Format | PCAP trace file (two separate PCAP files) |
| Size | 15.6 Megabytes and 648 kilobytes |
| Number of Packets | Day1 154173 packets, Day3: 3722 |
| Average Packet Size | 316.25 Bytes |

## 4.3 Results

In this section the results of the detection technique are presented. The detection results for the traces discussed in the previous section are presented in detail in this section. The results are comprised of anomalies that were detected using the proposed technique, as well as the anomaly detection rate and corresponding plots. The efficiency of the anomaly behavior detection was presented using recall and precision metrics. The main plots that are presented in the results are the 3-D cluster plot and the two volume feature plots. The three features used for the 3-D plots are the Destination IP Entropy (DIP), Destination Port Entropy (DP) and Source Port Entropy (SP). The features used for volume feature

plots are the Total Payload Bytes (TB) and Total Packet Count (PC). In addition to this the K-means cluster plot and the time-based view of the cluster plot are presented here.

Two levels of anomaly detection were used in this work. The two levels include:

- The first level to detect anomalies based on the threshold levels defined (refer to section 3.6)
- The second level to identify the type of anomaly based on the behavior predicted. (refer to section 3.8.1)

The anomalies were detected based on the threshold levels defined in the previous chapter. The threshold levels used are:

- Very High Entropy / Very High Volume (volume includes both total payload bytes and total packet count)
- High Entropy / High Volume
- Medium Entropy / Medium Volume
- Zero Entropy ( this level is considered an anomaly only when there is substantial volume change; otherwise it will be considered as normal)

### 4.3.1 SSH Honeypot Trace

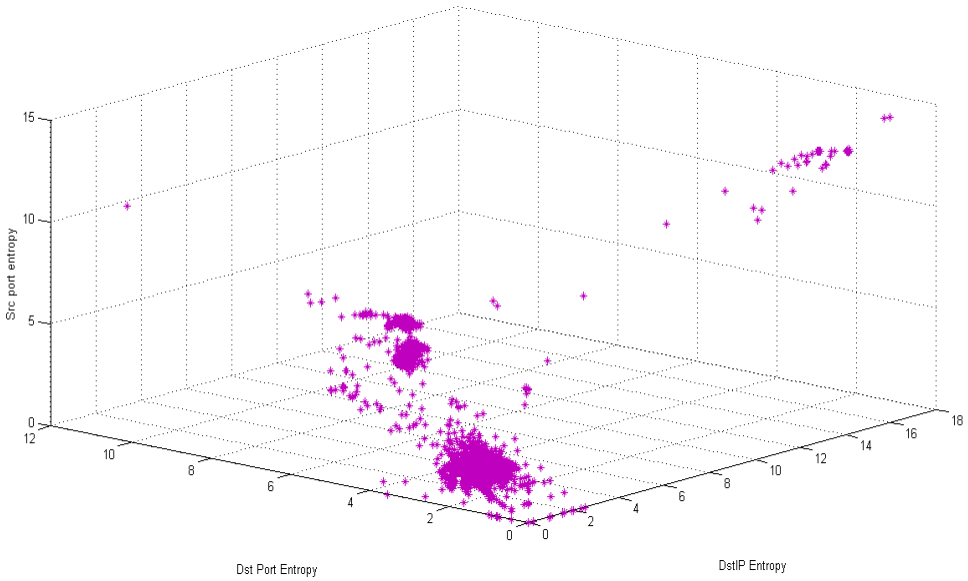


*Figure 4-7: Clusters plot for SSH based honeypot trace*

The cluster plot in Figure 4-7 shows the distribution of entropy values for the three selected features. The large variations and widely scattered points indicate different types of network events captured in the trace.

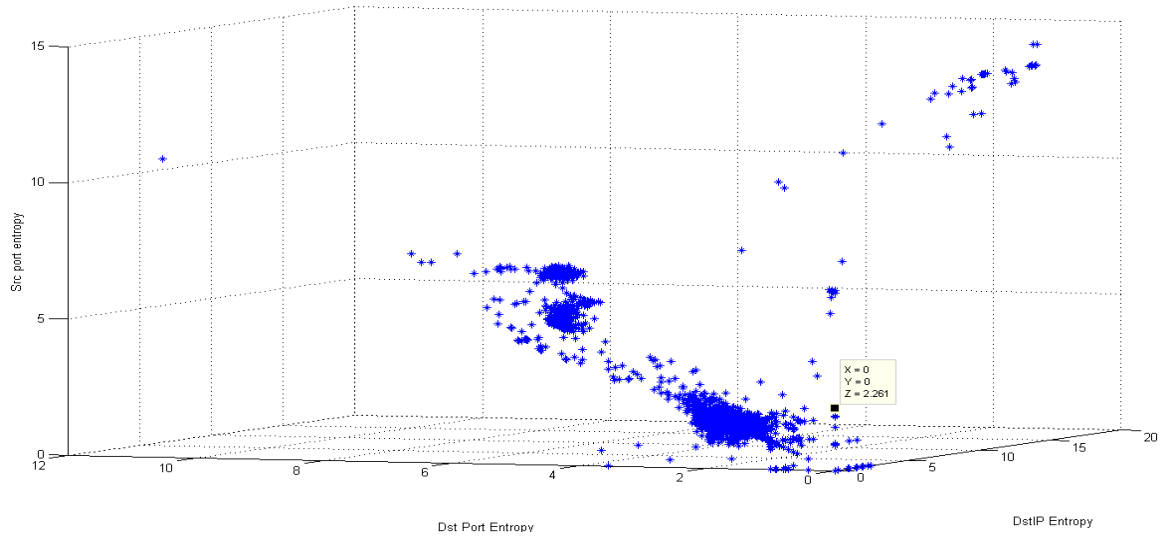


*Figure 4-8: Cluster plot for SSH based honeypot: from another angle*

Viewing the cluster plot from a different angle (refer to Figure 4-8 ) shows a better view of the positions of various points in the 3-D space. The anomalies were detected using both feature-based parameters and volume-based parameters. The volume plots of total payload bytes and total packet count are shown in Figure 4-9 and Figure 4-10. The visible peaks in these plots correspond to certain events in the network traffic and were used to detect the anomalies in the trace.

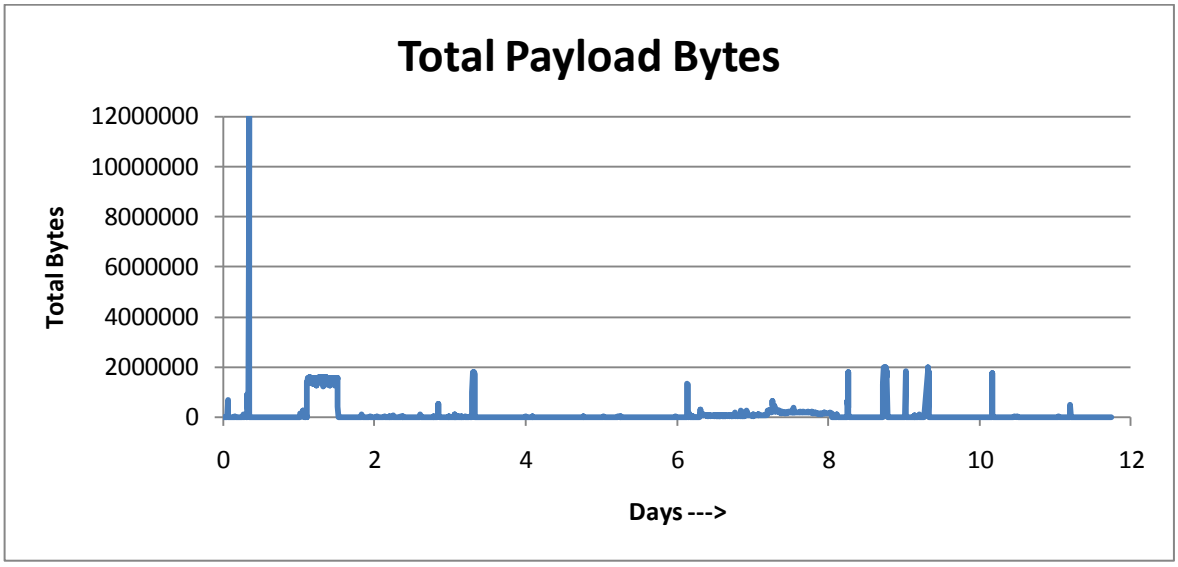


*Figure 4-9: Total Payload Bytes in SSH Honeypot Trace*

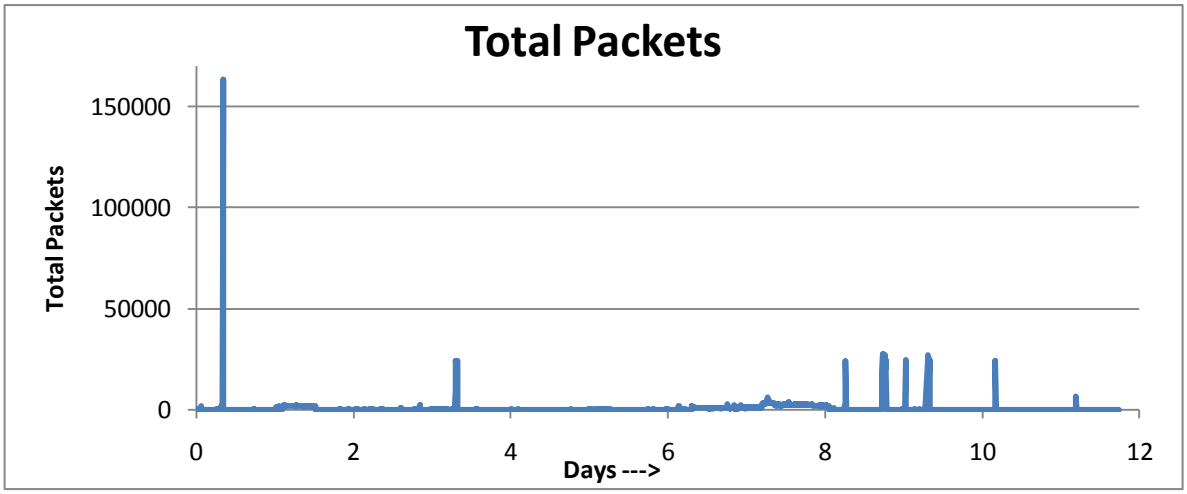


*Figure 4-10: Total Packet Count in SSH Honeypot Trace*

The plots also show the behavior of certain events in the trace, such as the behaviors during malicious file transfers and during a network scan.

#### *4.3.1.1 Types of Anomalies Detected*

The anomalies are detected based on the entropy values collected by running the Java code. The threshold levels mentioned in the beginning of the section were used to detect anomalies in the traffic. A manual analysis is carried out to identify the actual anomaly type. The detected behavior is also compared with the predicted behavior (refer to Table 3-18) to identify the correctness of the predicted behavior.

*Table 4-6 : Categories of anomalies detected*

| Detected Behavior | Type of Anomaly (by Manual Analysis) | Description |
|---|---|---|
| TB(VH), PC(H) | Malicious File downloads | Many malicious files were downloaded to the honeypot like: delles.tar.gz, z.jpg, crazyloop.tgz, ralph.tgz |
| TB (H), PC (M) | ICMP flood | Too Many Large Sized ICMP packets sent to honeypot |
| DIP (VH), SP(VH), TB(H), PC(VH) | Network Scan looking for open SSH ports | Honeypot was compromised by many hackers and after downloading some malicious file a network scan was performed. Many Network scan activities were detected. |
| DP (M), SP(H), TB(M), PC(H) | SSH BruteForce attack | A large number of SSH brute force attempts were made on the honeypot to gain the shell access. |

| | | |
|---|---|---|
| DIP (Z), DP(Z), SP(Z), TB(M), PC(M) | Many IRC conversations | Many IRC conversations visible. This was detected when too many messages were being exchanged. |
| DP (VH), SP(VH), TB(H), PC(VH) | Port Scan | The honeypot was scanned for different open ports from different IP addresses |
| DIP (M), DP(M),SP(H), TB(H), PC(H) | Vulnerability Scan by Honeypot | After the Honeypot was compromised a malicious file was transferred (*crazyloop*) after which the honeypot scanned various machines for open IRC channels |

A total of 41 anomalies which fall in the categories mentioned in Table 4-6 were detected in this trace. The actual number of anomalies present in this trace is unknown as it is not available from the source of this trace file. We also did not find any existing analysis done on this trace to confirm the number of anomalies present. K-means clustering was further used to detect different cluster regions with in the traffic.

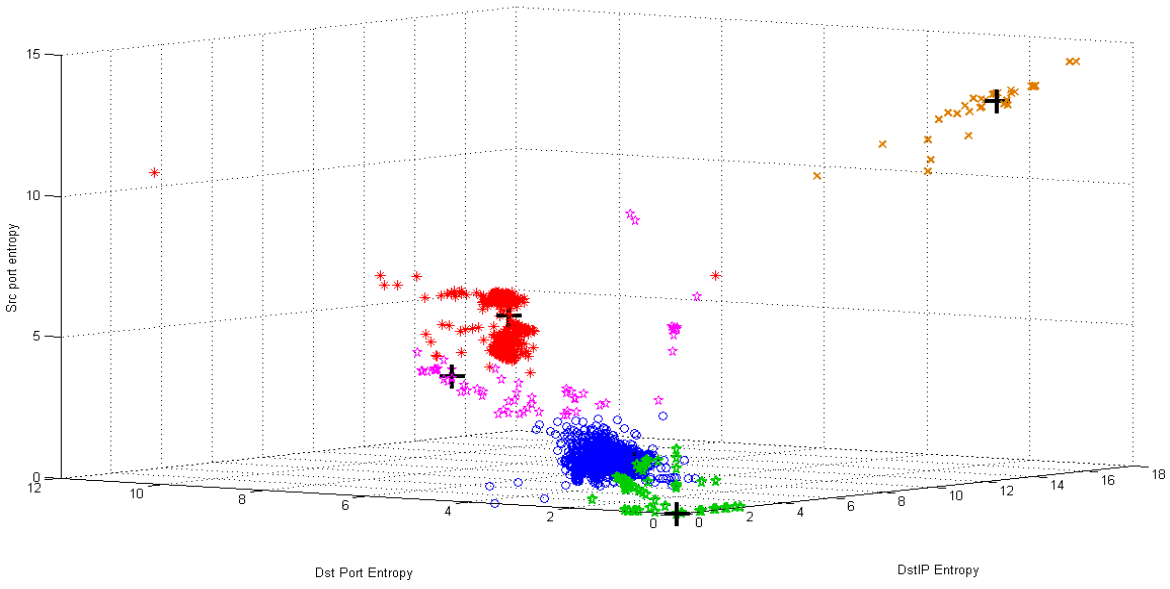


*Figure 4-11: K-means Cluster Plot for SSH based honeypot trace*

The k-means cluster plot in Figure 4-11 shows the different clusters that are available for the entropy values recorded for the SSH based honeypot trace. Figure 4-12 shows the time based view of these clusters and helps in visualizing the sequence of events that took place. Figure 4-12 shows the events based on the time of occurrence and separate markers were used for each of the 12 days of traffic.

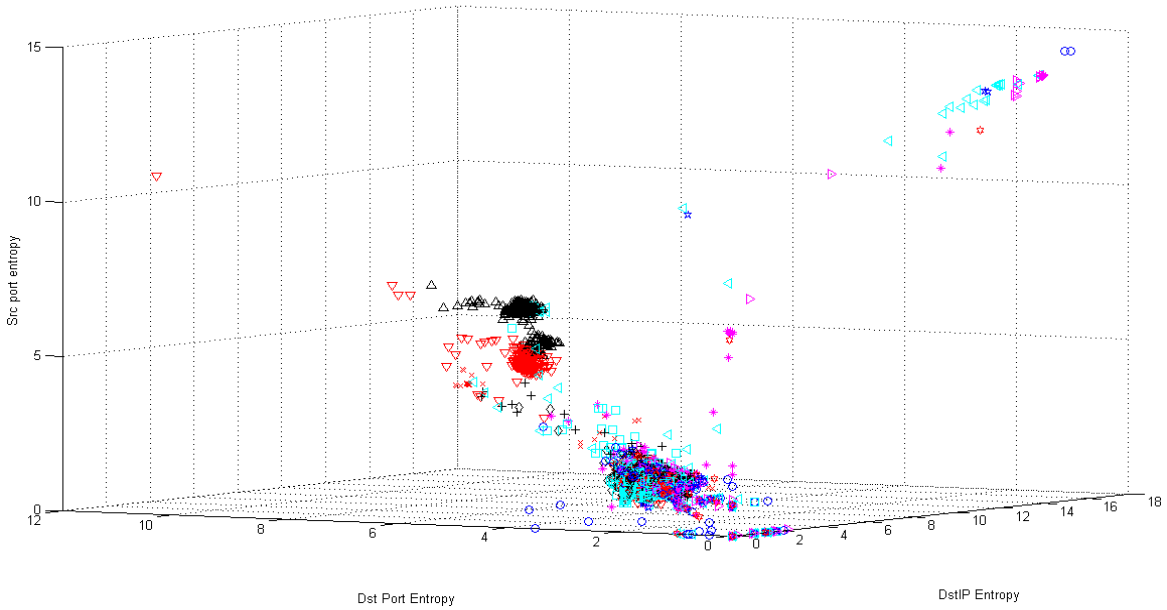


*Figure 4-12: Time based view of network events using different markers and colors for each day*

Percentage Detection Rate: not applicable, because this trace was used in the test data set and the actual number of anomalies in the trace is unknown.

### 4.3.2 Scan 27: Scan of the Month Challenge

The detection results for the scan of the month trace (Scan 27) are presented in this section. The main plots that were used to detect anomalies are the cluster plots and volume-based parameters plots. The three feature cluster plot of destination IP entropy,

destination port entropy and source port entropy is show in Figure 4-13. The honeypot was exposed to a large number of attacks as seen by the wide spread entropy values in Figure 4-13. Many attackers had tried to compromise the honeypot and some worm activity was also detected in the trace.

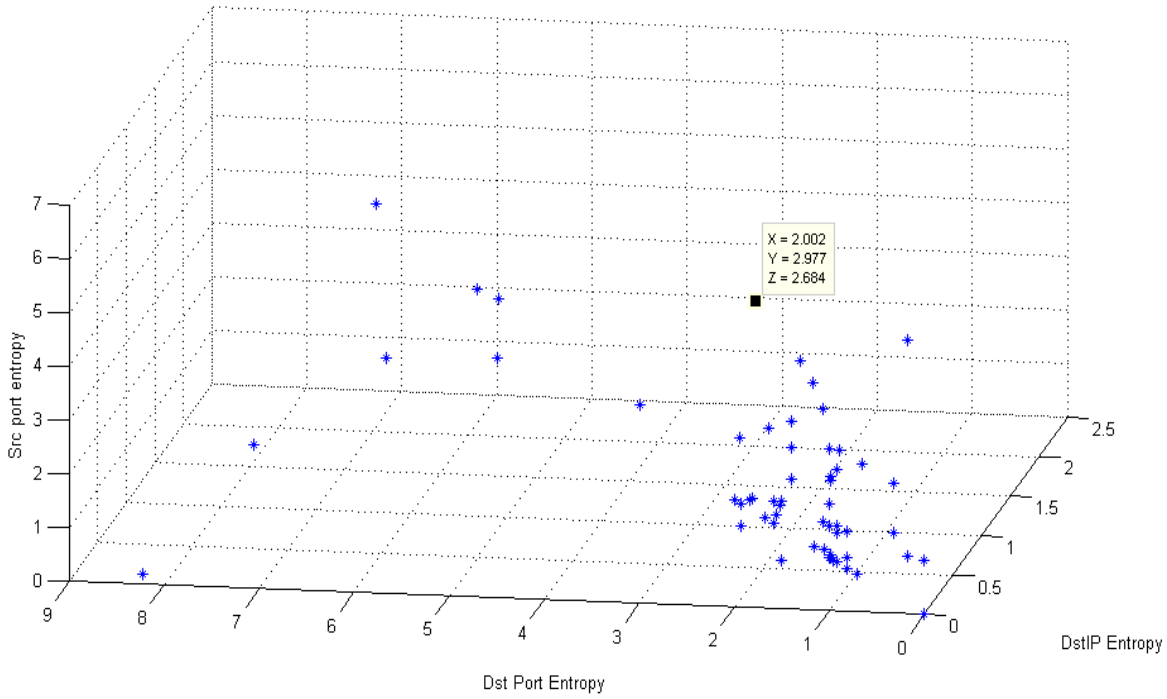


*Figure 4-13: Cluster plot for Scan 27*

The points having high entropy values were due to some port scan activity. The volume plots in Figure 4-14 and Figure 4-15 also confirm the transfer of malicious files to the honeypot. The points with low entropy values close to zero correspond to the IRC traffic that was caused as the honeypot was connected to the IRC network by one of the attackers.

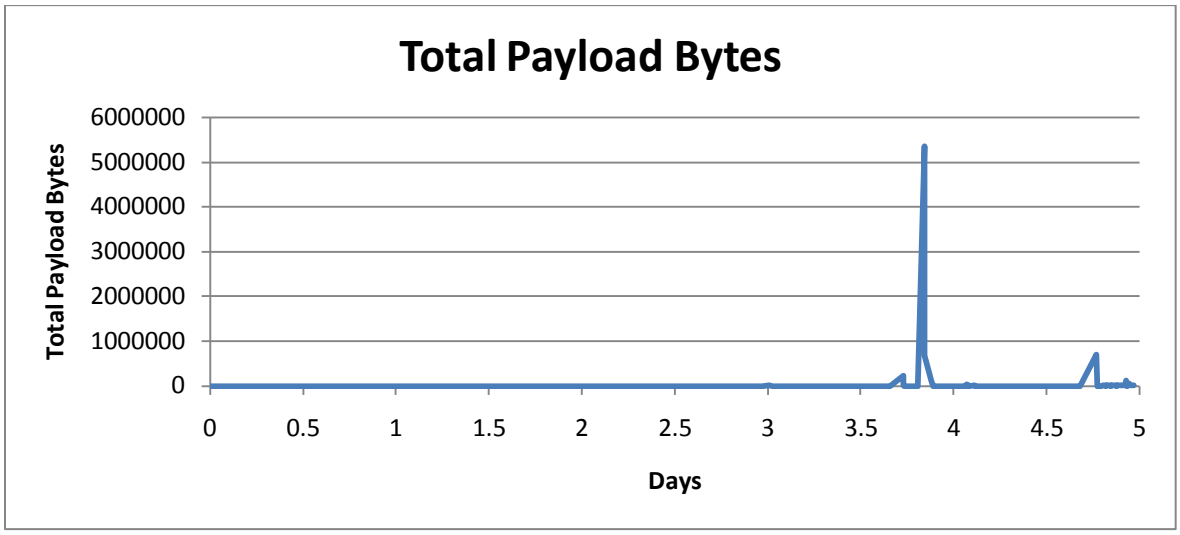


*Figure 4-14: Total Payload Bytes for scan 27 trace*

Most of successful system compromise attacks were seen after the third day which correspond to the peaks shown in Figure 4-15 and the after a successful system compromise, the honeypot was connected to an IRC network.

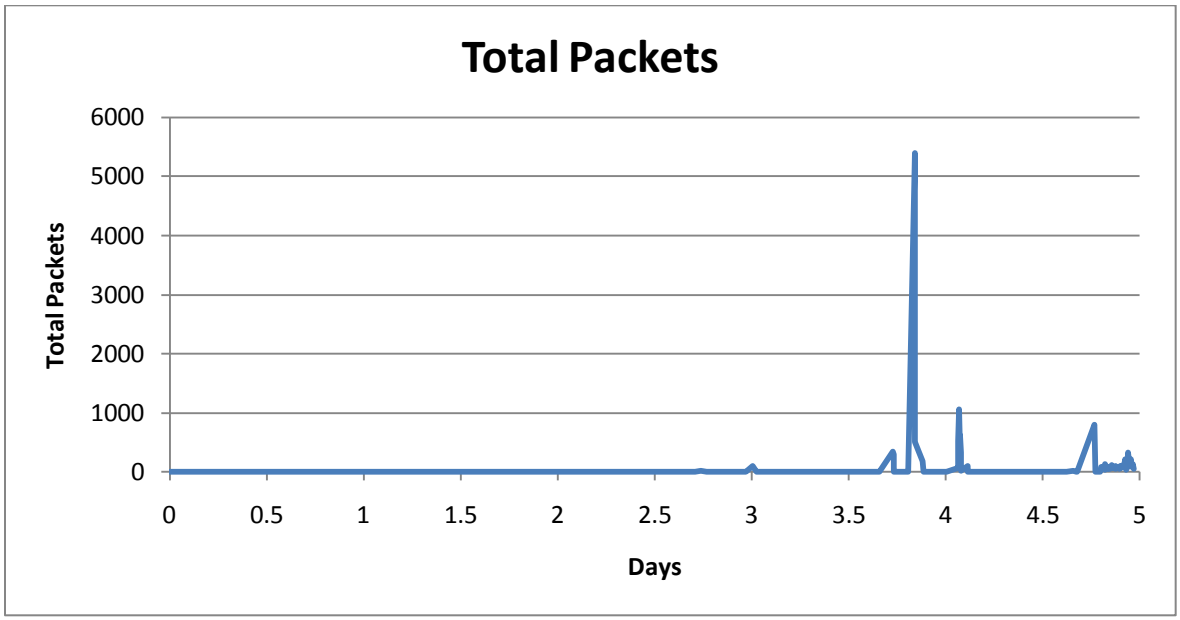


*Figure 4-15: Total Packet Count for Scan 27 trace*

### *4.3.2.1 Types of Anomalies Detected*

*Table 4-7 : Anomalies detected in Scan27 trace*

| Detected Behavior | Type of Anomaly (by Manual Analysis) | Anomaly Behavior Detected by Reverse Mapping | Actual number Occurrences Reported [by Honeynet.org] |
|---|---|---|---|
| TB(M), PC(M) (5 occurrences) | SMB attacks | The detected behavior not available in the known behavior set. [refer to Table 3-18] | 5 occurrence reported |
| DP(M), SP(M), TB(H), PC(M) 1 occurrence | Buffer overflow attempt and System Compromise Many attempts were made to compromise the honeypot. | **System Compromise** | 1 occurrence reported |
| TB(VH), PC(H) 1 occurrence | Malicious file download \System32\\PSEXESVC.EXE , \\System32\\inst.exe | **Malicious File Download** | 1 occurrence |
| DP (VH), SP(VH), TB(M), PC(M) 1 occurrence | **Port Scan** Honeypot was scanned for open ports. | The detected behavior does not match any of the predicted behavior | 1 occurrence |
| DP(M), SP(M), TB(H), PC(H) 1 occurrence | **HTML script kiddies** This attack tries to scan the IIS web server for vulnerabilities and this is done by initiating automated HTML scripts | The detected behavior not available in the known behavior set. | 1 occurrence IIS web server attack |
| TB(M) 1 occurrence | Attempts to exploit buffer overflow in indexing service. | The detected behavior not available in the known behavior set. | 1 occurrence CodeRed II worm |
| DIP (Z) , | **IRC communications** | **IRC communication** | 1 occurrence |

| | | | |
|---|---|---|---|
| DP(Z), SP(Z), TB(M) PC(M) 1 occurrence | IRC communication initiated on Honeypot by the attacker. | | |
| Not Detected | X | X | Slammer Worm, 1 occurrence |
| **Total Detected = 11** | | | **Total Reported = 12** |
| **Anomaly Detection Rate** | | | **91.6%** |

A total of 11 anomalous events which fall in the categories mentioned in Table 4-7, were detected using the proposed technique. Honeynet.org reported 12 anomalous events in this scan of the month challenge. The anomaly that was undetected represent attacks on MS-SQL server UDP port 1434 which resembled the slammer worm. A total of 55 packets were sent to the honeypot targeting the port 1434 but these packets were sent at different times during the five days period. The reason for this not getting detected is that the time gap between these packets was large and did not cause rapid change in entropy. Based on the detection results, a detection rate of 91.6% was achieved using the proposed technique.

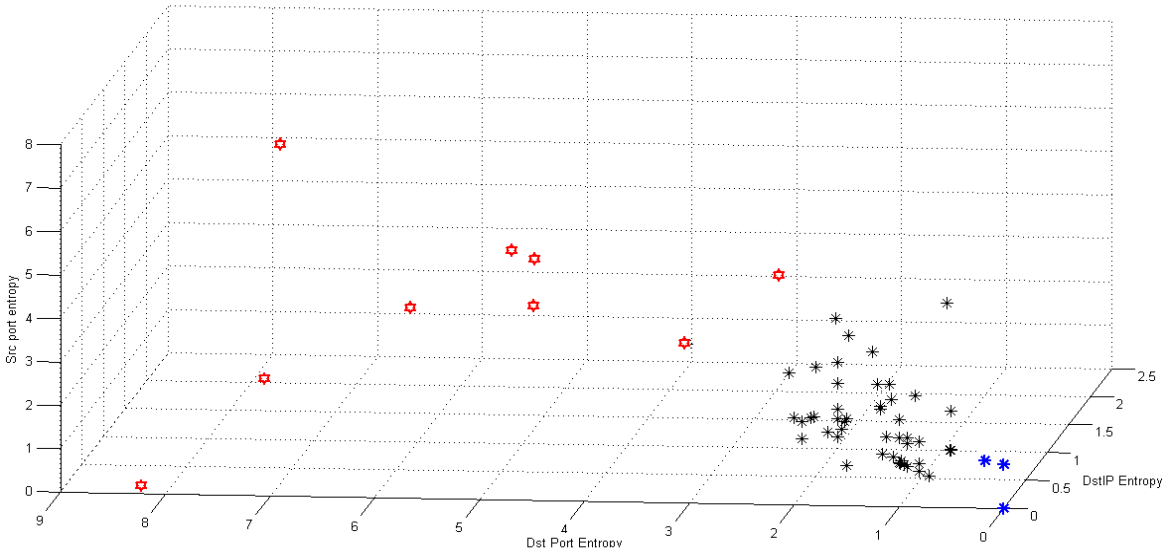


*Figure 4-16: K-Means cluster plot for Scan 27 trace*

The K-means cluster plot shows (refer to Figure 4-16) the main clusters detected in the trace. K-means enabled the detection of three clusters. The cluster marked with * represents the normal traffic region and most of the traffic falls in this cluster. The time view plot in Figure 4-17 displays the spread of events throughout the five day period. It is clear from this plot that system compromise attempt happened on third day and the port scan and other anomalies occurred on the fourth day.

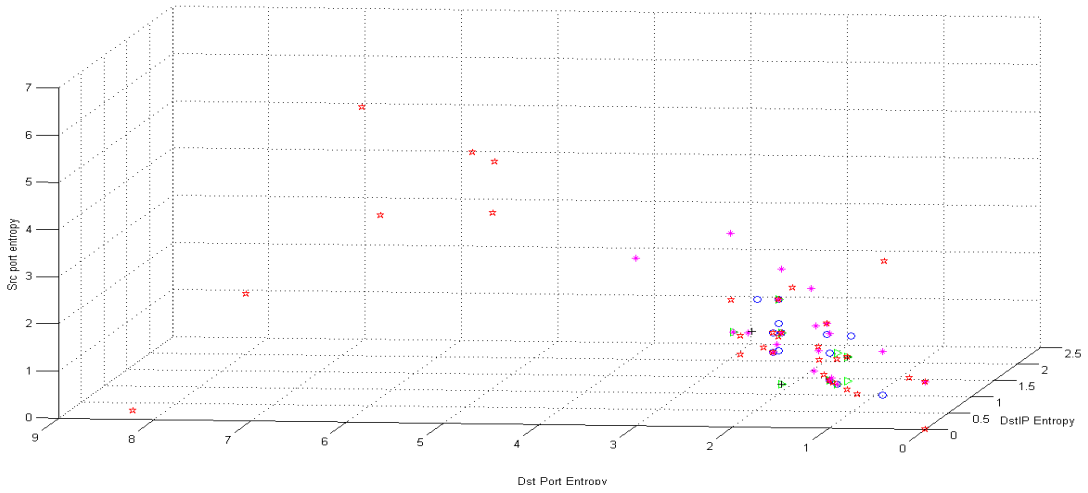


*Figure 4-17: Time based view of events for Scan 27 trace*

### 4.3.3 Lab Trace with Synthetic Anomalies

The lab trace was used to test the effectiveness of the proposed technique. In this trace synthetic anomalies were injected using various tools used for penetration testing. The most popular tools that were used are: NMAP, OpenVAS scanner and Metasploit. The Metasploit tool was used to generate system exploits which target various services on the honeypot. The attacks were generated five days after the honeypot was connected to the Internet. In this lab trace, 27 anomalies were inserted (refer to Table 4-3 for the list of inserted anomalies). But, as this honeypot was connected to the Internet other attacks were also detected. The cluster plot in Figure 4-18 shows various groups of anomalous activities.

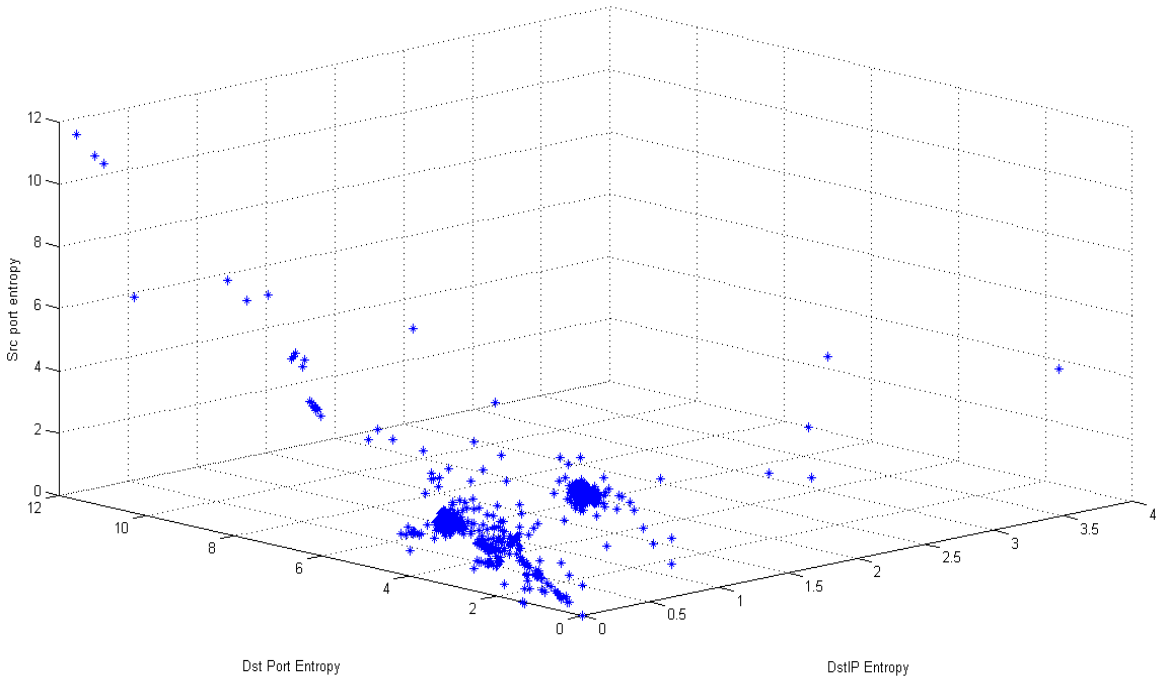


*Figure 4-18: Cluster view of Labtrace*

The volume based parameters also show significant changes during the anomalies as seen in Figure 4-19 and Figure 4-20. The peaks correspond to scanners and malicious data transfers.

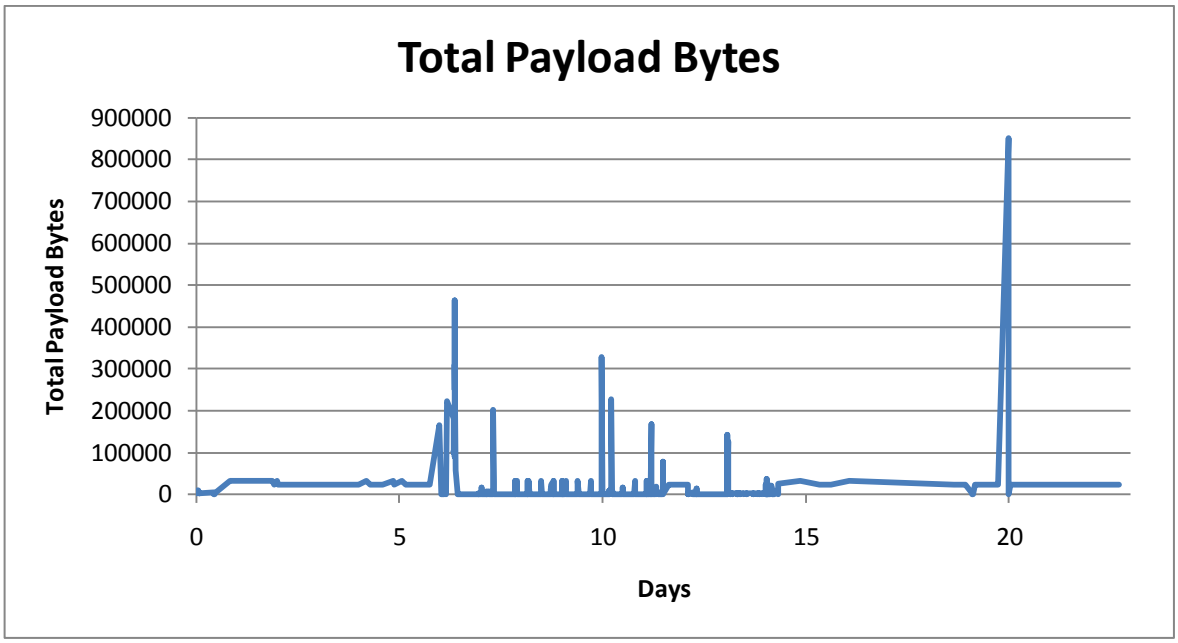


*Figure 4-19: Total Payload Bytes in Labtrace*

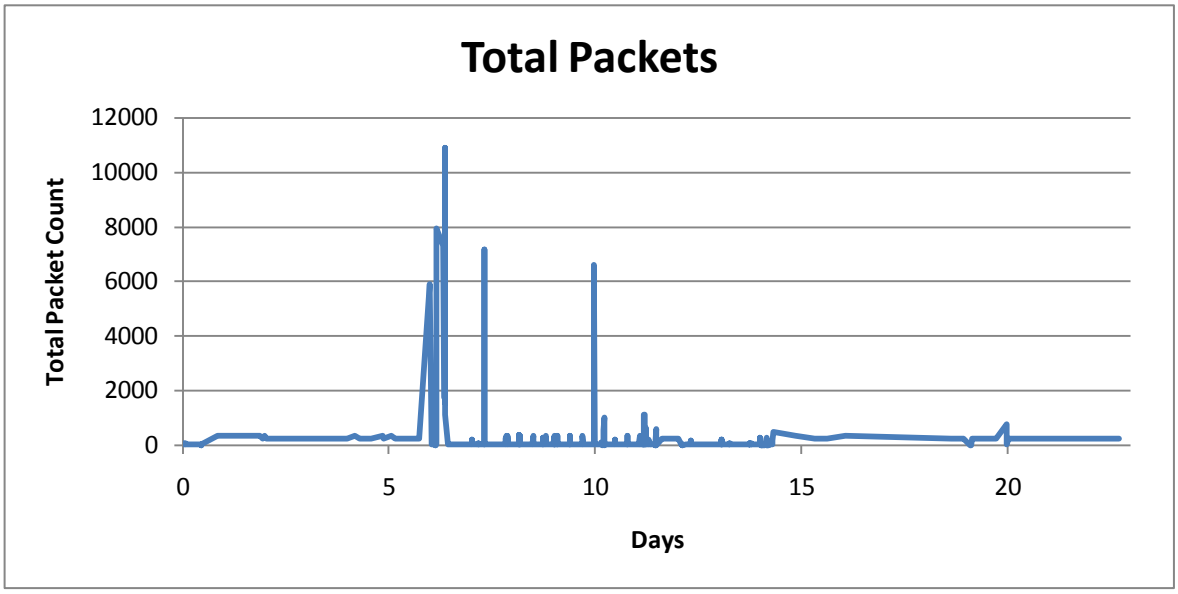


*Figure 4-20: Packet Count for Labtrace*

### *4.3.3.1 Types of Anomalies Detected*

*Table 4-8 : Categories of Anomalies detected in LabTrace*

| Detected Behavior | Type of Anomaly by Manual Analysis | Anomaly Behavior by Reverse Mapping | Reported Number of Occurrences (Refer to Table 4-3) |
|---|---|---|---|
| DP (VH), SP(VH), TB(H), PC(VH) 5 Occurrences | Port Scan Different NMAP port scan types were used to scan the honeypot. The comprehensive scan shows the highest entropy value and the highest number of packets were sent in this type of scan. | Port Scan | 5 occurrences |
| DP(VH), SP(VH), TB(H), PC (VH) 1 occurrence | Vulnerability Scanning Vulnerability scanning using Open VAS Scanner | Port Scan | 1 occurrence |
| DP(M), SP(M), TB(H), PC(M) 12 occurrences | Vulnerability exploits (Metasploit exploits) These are system compromise attempts | System Compromise | 12 occurrences |
| TB (H), PC(M) 3 occurrences | System Compromise | ICMP flood | 3 occurrences |
| DP (M), SP(H), TB(M), PC(H) 4 occurrences | Password Brute force and Fuzzers These attacks used various combinations of username and passwords to guess the account password. The fuzzer tools fall in this category. | Bruteforce | 4 occurrences |

| | | | |
|---|---|---|---|
| DP(M), TB(M), PC(M)<br>24 occurrence | SMB connection attempts<br>Most of these attempts try to connect to the Microsoft-ds port (445) on the remote machine and try to gain access to the system shares. | Detected behavior not available in the known behavior set | X |
| Not Detected | X | X | SSH attack and Microsoft Message Queueing Service Path Overflow exploit<br>2 occurrences |
| **Total Anomalies Detected = 25** | | | **Total Anomalies Reported = 27** |
| **Anomaly Detection Rate** | | | **92.5%** |

The major categories of anomalies that were detected in the Labtrace are listed in Table 4-8. A total of 27 attacks launched against the honeypot using the Backtrack OS. Using the proposed technique, 25 attacks were successfully detected with a detection rate of 92.5%. The undetected anomalies were the SSH attack and the Microsoft Message Queueing Service Path Overflow exploit. The reason for not detecting these two attacks is that they did not cause significant changes in entropy values. In metasploit there was no exploit available for Open SSH (tool that was installed in the honeypot) due this the exploit that was attempted did not succeed and only a few packets were launched during this attack. The second attack, i.e. Microsoft Message Queueing Service Path Overflow exploit was also not successful and hence it did not generate many packets to cause changes to the entropy values. A total number of 49 attacks which include the 25 attacks

generated using metasploit and other attacks caused by other machines in the network were detected.

The K-means clustering was applied on the entropy values and Figure 4-21 shows the different clusters that were detected. Four cluster regions were detected in the LabTrace. The cluster one was the normal traffic and the cluster two and three represents metasploit exploits and SMB attacks, respectively.

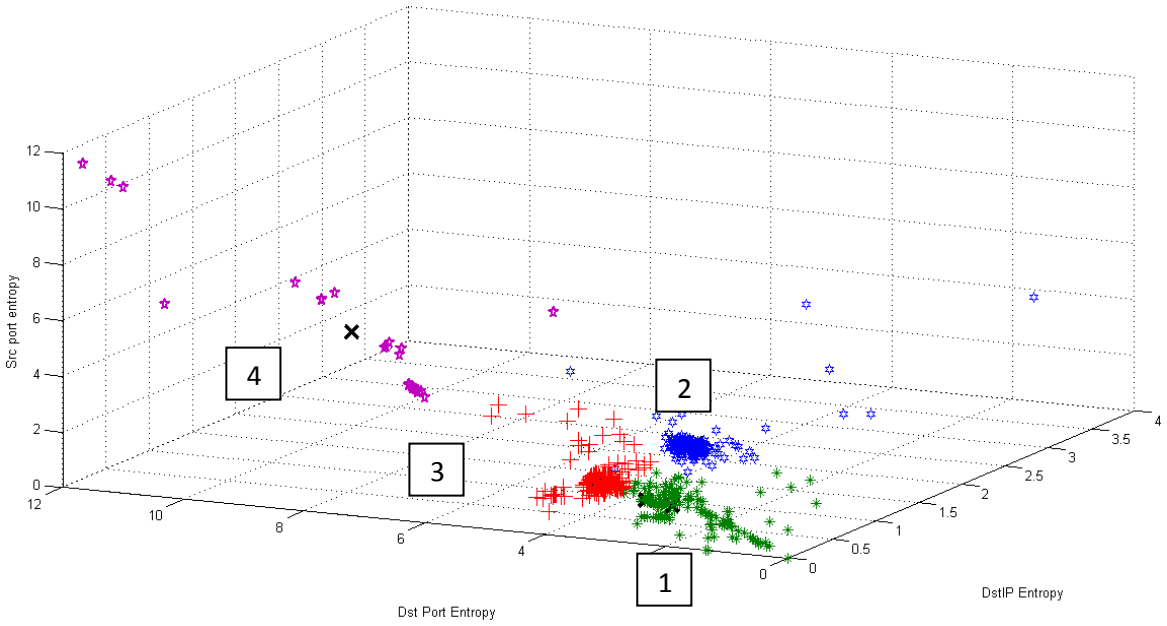


*Figure 4-21: K-Means Cluster view of Labtrace*

The time view trace (refer to Figure 4-22) shows the different times at which the events occurred in the honeypot lab trace during the 25 day period. The attacks were generated five days after the honeypot was connected to the Internet and most of the attacks were generated during the end.

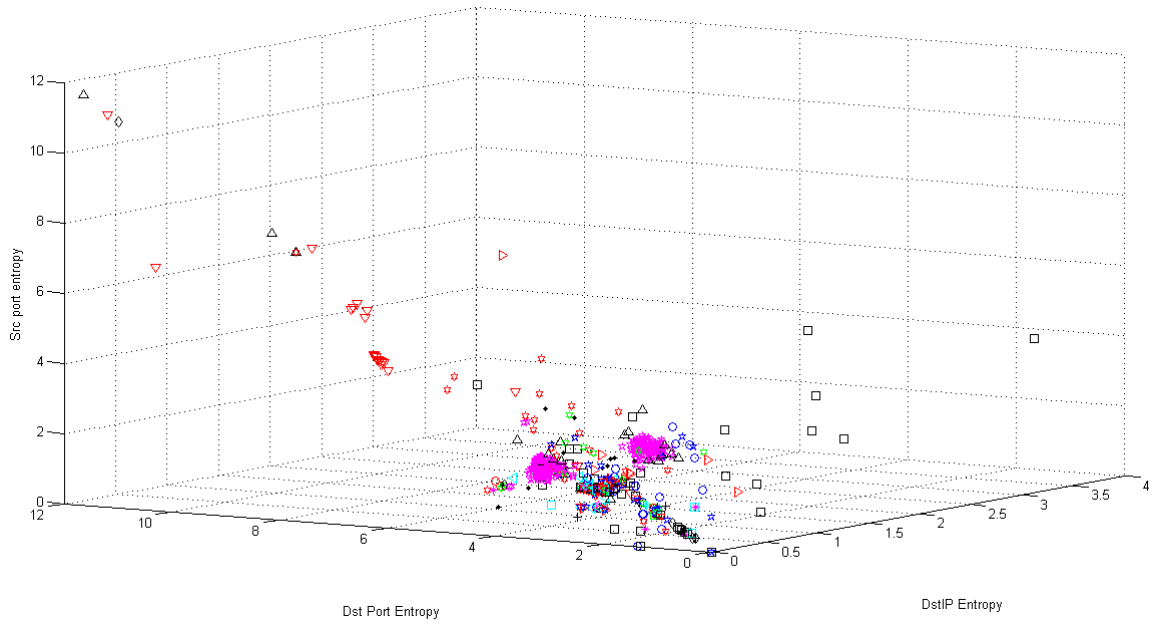


*Figure 4-22: Time view of Labtrace*

## 4.3.4 Dionaea Capture Trace

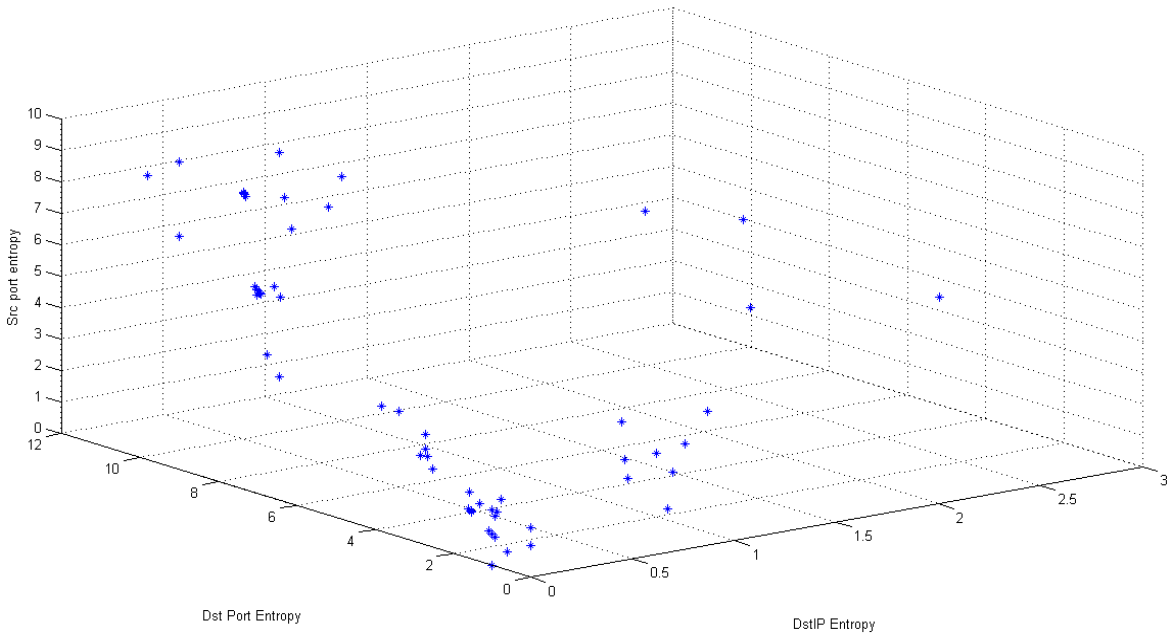


*Figure 4-23: Cluster plot of Dionaea Capture Day1 Trace*

The cluster plot in Figure 4-23 shows the scattered points of the trace. The points having high entropy values are related to the port scan activity seen in the trace. The points in the region of the x-axis from 3 to 6 are related to brute force attempts and connection attempts to FTP and HTTP services.

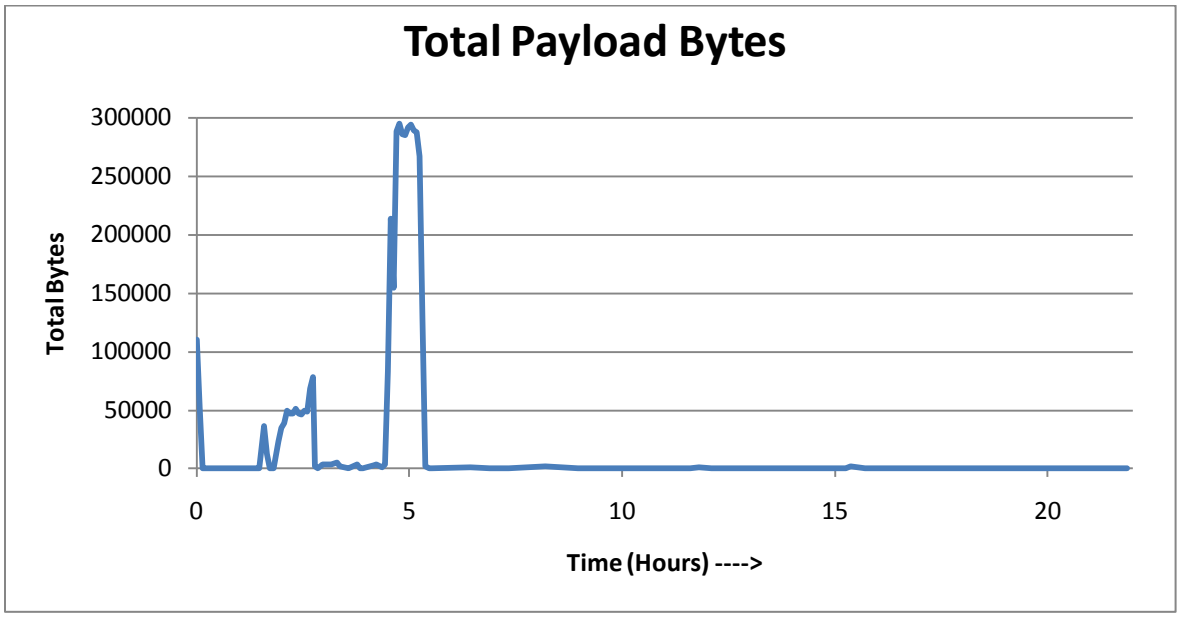


*Figure 4-24: Total Payload Bytes in Dionaea Capture Day1 Trace*

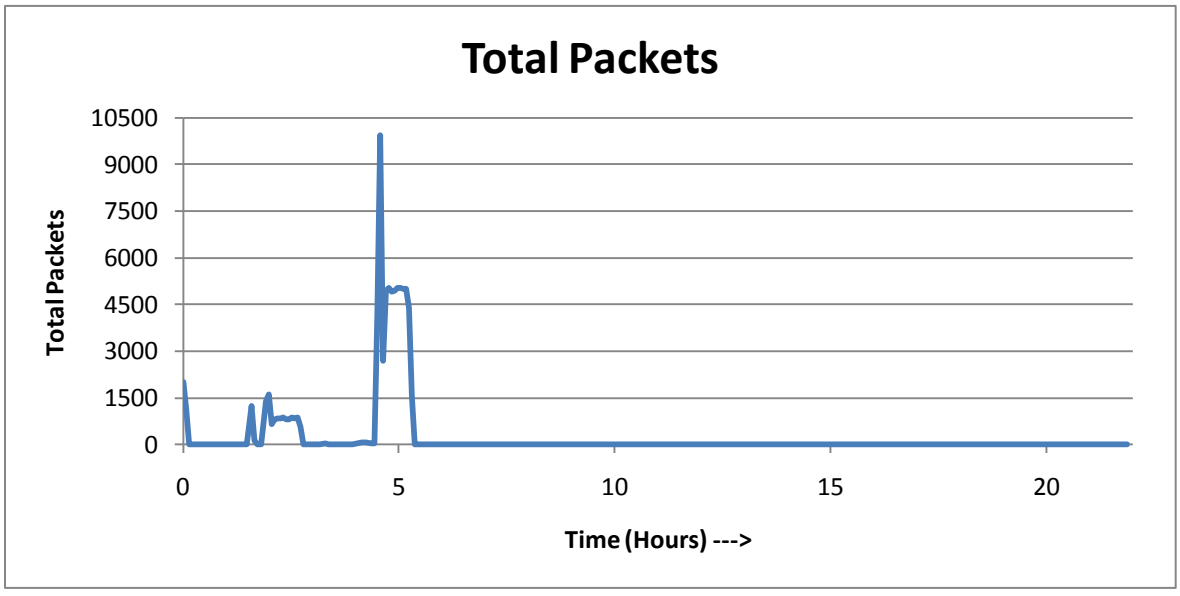


*Figure 4-25: Total Packets in Dionaea Capture Day1 Trace*

The volume plots in Figure 4-24 and Figure 4-25 show the visible peaks that correspond to anomalies in the traffic. The large peaks were seen due to port scan activity.

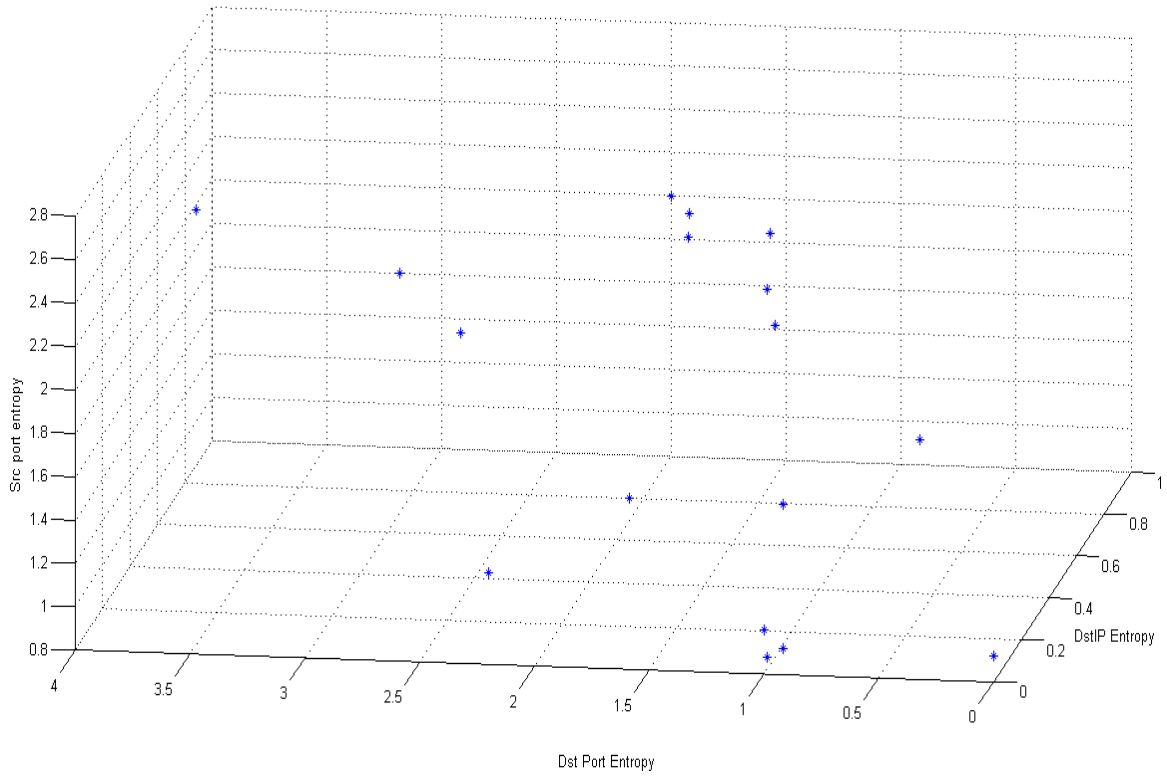


*Figure 4-26: Cluster Plot of Dionaea Capture Day3 trace*

The cluster plot for day 3 is shown in Figure 4-26. The traffic on day 3 was limited and had only few anomalous events compared to the first day. The plot shows only few points as most of the points are overlapping with each other. The points having an entropy value of more than 2 were anomalous. Volume plots in Figure 4-27 and Figure 4-28 show similar activities and the only anomalous event that caused considerable volume change was a web server vulnerability scanning.

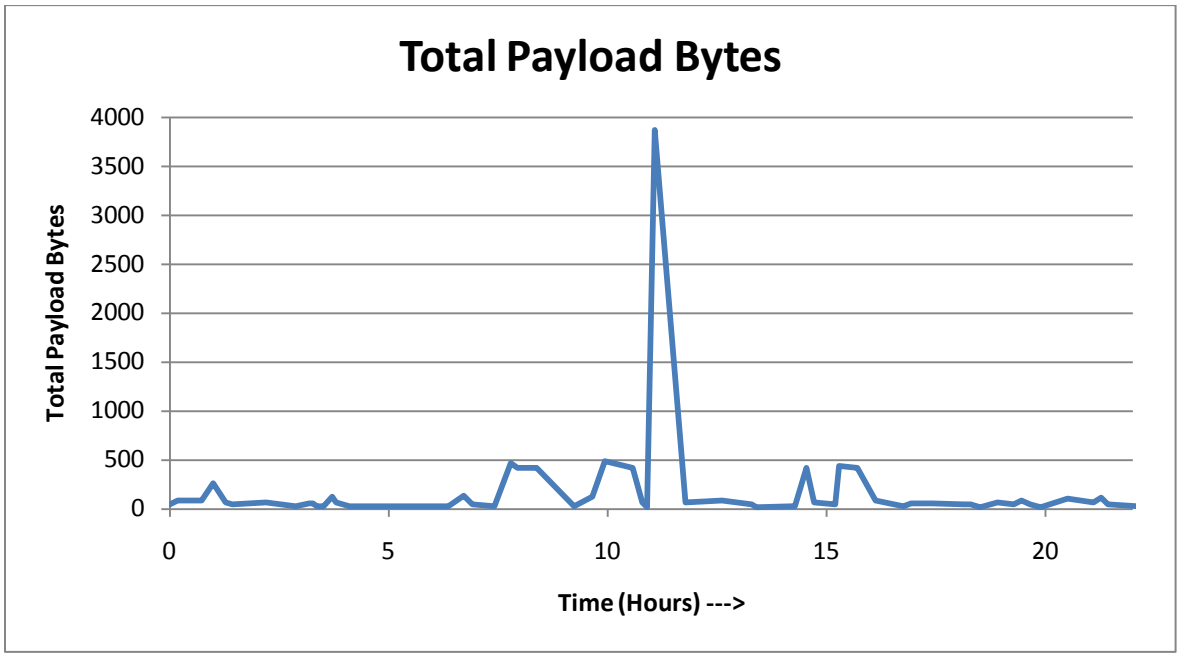


*Figure 4-27: Total Payload Bytes for Dionaea Day3 Capture*

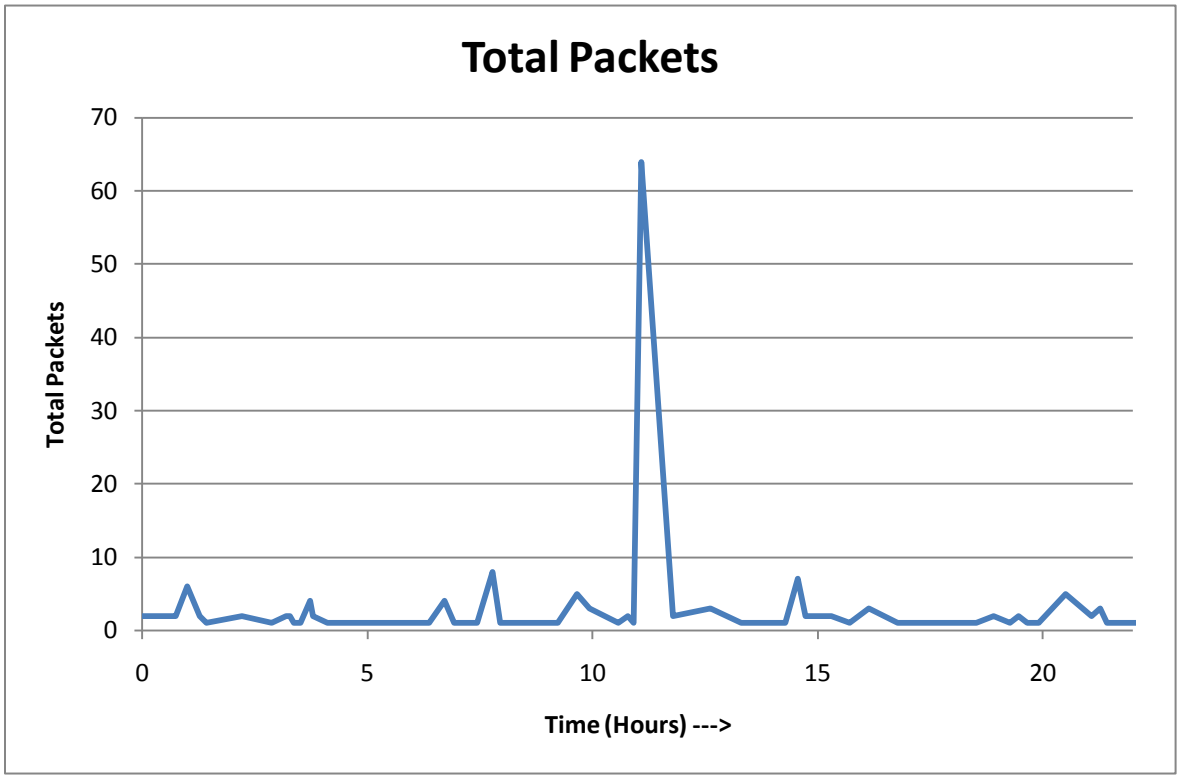


*Figure 4-28: Total Packets for Dionaea Day3 Capture*

### *4.3.4.1 Types of Anomalies Detected*

The list of anomalies that were detected in day1 traffic is explained in the Table 4-9.

*Table 4-9 : Anomalies detected in Dionaea Day1 capture trace*

| Detected Behavior | Type of Anomaly by Manual Analysis | Anomaly Behavior by Reverse Mapping | Reported Anomalies |
|---|---|---|---|
| DP(VH), SP(VH), TB(H), PC(VH)<br>3 occurrences | **Port Scan**<br>Different NMAP port scan types were used to scan the honeypot. | Port Scan | 3 occurrences |
| DP(M), SP(M), TB(H), PC(M)<br>1 occurrence | **MS-SQL Brute force attempts**<br>Multiple login attempts were made to break into the MSSQL server using different passwords. | System Compromise | 1 occurrence |
| DP(M), SP(H)<br>2 occurrences | **Web Robots (also known as Web Wanderers, Crawlers, or Spiders)**<br>These probes can either be malicious if they are coming from malicious users or it could be from search engine websites like Google which try to index the web sites on the network. | Detected behavior is not available in the known behavior set | 2 occurrences |
| DP(M), SP(M)<br>4 occurrences | Connection attempts on popular ports (HTTP, FTP, MSSQL etc.)<br>These could be potential attacks as they are | Detected behavior is not available in the known behavior set. | 4 occurrences |

| | | | |
|---|---|---|---|
| | unauthorized users trying to connect to services available on the honeypot. | | |
| Not Detected | X | X | SIP worm reported 1 occurrence |
| **Total Anomalies Detected = 10** | | | **Total Anomalies Reported = 11** |
| **Anomaly Detection Rate** | | | 90.9% |

A total of 10 anomalies were detected with a detection rate of 90.9% in the day1 dionaea traces which fall in the categories mentioned in Table 4-9.

List of anomalies detected in day 3 traffic are given in Table 4-10. In this trace, the maximum entropy value detected is 3.5. The destination IP entropy is equal to zero for most of the duration of the trace as there were only one or two IP addresses seen during most of five minute intervals.

*Table 4-10: Anomalies detected in Dionaea day3 capture trace*

| Detected Behavior | Type of Anomaly (by Manual Analysis) | Anomaly Behavior by Reverse Mapping | Reported Anomalies |
|---|---|---|---|
| SP(M), TB(M) 1 occurrence | Phpmydmin Attack A number of packets were sent to the port 80 and from different ports to connect to PhpMyadmin | Detected behavior is not available in the known behavior set. | 1 occurrence |
| DP(M), SP(M) 1 occurrence | FTP connection attempts Connections attempts to the FTP server made from different ports. | Detected behavior is not available in the known behavior set. | 1 occurrence |

| | | | |
|---|---|---|---|
| DP(M), SP(H), TB(M), PC(M)<br>1 occurrence | Web Server Vulnerability Scanning<br>The web server was scanned for vulnerabilities | Detected behavior is not available in the known behavior set. | 1 occurrence |
| Not Detected | X | X | SIP Worm reported<br>1 occurrence |
| **Total Anomalies Detected = 3** | | | **Total Anomalies Reported = 4** |
| **Anomaly Detection Rate** | | | **75%** |

The K-means cluster plot in Figure 4-29 shows the visible clusters in the dionaea capture. The cluster three represents the port scanning activity and the cluster two represents the other attacks discussed in Table 4-10. A total of 3 anomalies with were detected with an anomaly detection rate of 75% in the day3 trace. Similar events happened at different times of the day during the dionaea capture as seen in the time view plot of Figure 4-30. The SIP scanning worm was not detected using the proposed technique. The SIP worm was not detected because; during each connection attempt it was sending only two packets to the honeypot. Due this the anomaly did not cause significant changes in the selected features.

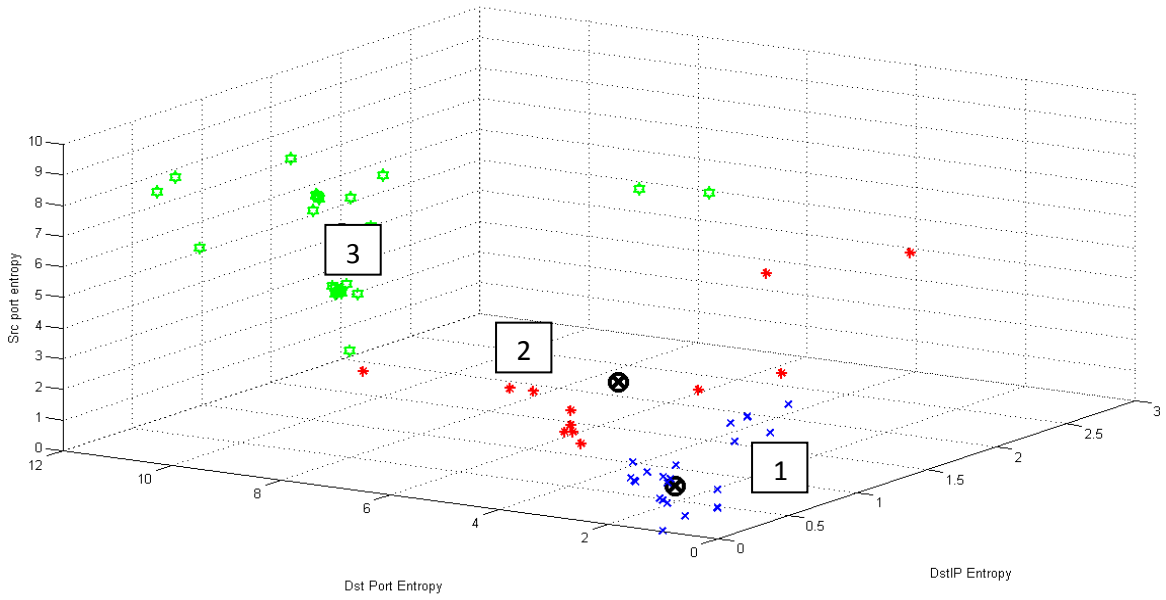


*Figure 4-29: K-Means Cluster plot for Dionaea Capture*

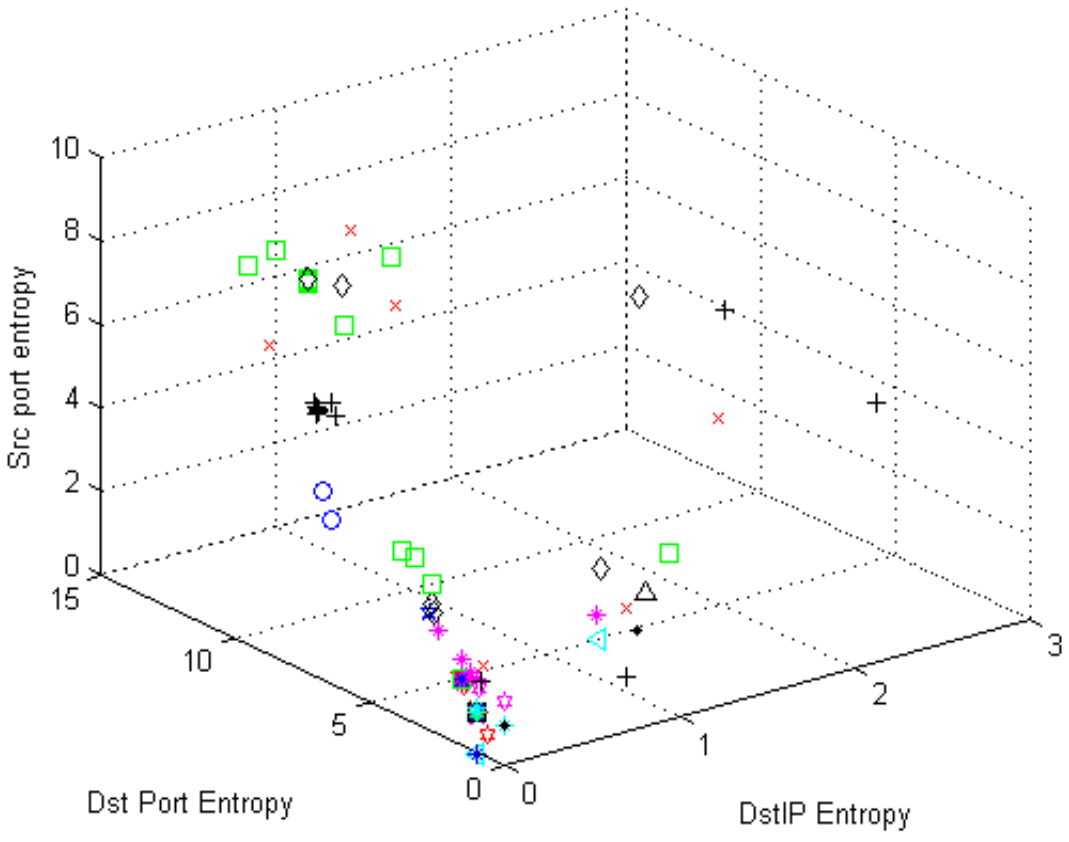


*Figure 4-30: Time View of Dionaea Capture Trace*

## 4.4 Recall and Precision of Anomaly Behavior Detection

Recall and precision metrics were used to evaluate the effectiveness of the reverse mapping applied to detect the anomaly behavior from the predicted behavior.Table 4-11 summarizes the anomaly detection rate as well as precision and recall percentages for the proposed technique. The metrics used are [36] :

$$Recall = \frac{True\ Posivites}{True\ Positives + False\ Negatives}$$

$$Precision = \frac{True\ Posivites}{True\ Positives + False\ Positives}$$

*Table 4-11: Recall and Precision of Anomaly Behavior Detection*

| **Data Set** | **Anomaly Detection Rate** | **Anomaly Behavior Detection** | | | | |
|---|---|---|---|---|---|---|
| | | **R** | **D** | **I** | **Precision % (I/D)** | **Recall % (I/R)** |
| Scan 27 Trace | 91.6% | 10 | 4 | 3 | 75% | 30% |
| LabTrace | 92.5% | 43 | 23 | 19 | 82.6% | 44.18% |
| Dionaea Trace | 86.6% | 9 | 4 | 3 | 75% | 33.33% |
| **Total** | **Anomaly detection Rate: 90.25%** | **62** | **31** | **25** | **80.65%** | **40.32%** |

*Table 4-12: Recall and Precision without considering multiple occurrences*

| **Data Set** | **R** | **D** | **I** | **Precision % (I/D)** | **Recall % (I/R)** |
|---|---|---|---|---|---|
| Scan 27 Trace | 6 | 4 | 3 | 75% | 50% |
| LabTrace | 4 | 5 | 3 | 60% | 75% |
| Dionaea Trace | 3 | 2 | 1 | 50% | 33.33% |
| **Total** | **13** | **11** | **7** | **63.63%** | **53.84%** |

The anomaly detection rate measures the percentage of anomalies detected based on the reported behavior out of all anomalies reported for the trace. A detection rate of 90.25% was achieved based on the threshold levels (discussed in section 3.7**Error! Reference source not found.**) used in the proposed technique. The effectiveness of mapping the detected anomaly behavior to the appropriate anomaly was found using precision and recall metrics. A precision of 80.65% and a recall percentage of 40.32% were achieved using the proposed technique when all the occurrences of different anomalies are taken into account. The precision and recall percentages for detecting the anomaly type i.e. without considering multiple occurrences for each type (refer to Table 4-12) show better recall percentages compared to previous values in Table 4-11. The values obtained in Table 4-11 are based on the fact that certain anomalies occurred many times and did not match the predicted behavior set (refer to Table 3-18). This led to decrease of the recall value. The reason for getting low values for the recall is that we have only few number of anomaly behaviors mapped in the predicted behavior set. Therefore, we expect the recall to improve if more traces are used to update the mapping table (Table 3-18) obtained in the previous chapter.

# 4.5 Summary

This chapter described in detail the different results that were obtained using the anomaly detection technique. The feature-based parameters i.e., Destination IP entropy, Destination Port entropy and Source Port entropy were used to obtain the cluster plots. Two volume-based features i.e., Total Payload Bytes and Total Packet Count were used to obtain the volume plots. The details of traces that were used in testing our proposed

technique were discussed in this chapter. The total number of anomalies detected using this technique as well as the types of anomalies detected were presented.

The results mainly focus on the anomaly detection rate obtained using the proposed technique. In Honeynet systems, all traffic coming to the honeypots is considered malicious. Based on this fact, we are not presenting the false alarm rate and we consider that anomalies that are detected are all malicious activities. The results show that the proposed technique has a detection rate of 90% and enabled the detection of most of the anomalies in the network traffic. In addition the proposed technique achieved a recall of 40% and precision of 80.65%. This technique will be very useful for anomaly detection in Honeynet systems, as it will ease the data analysis process. The use of both volume-based and feature-based parameters is very important in detecting the anomalies in any network traffic.

# *Chapter 5*

# Conclusion and Future Work

Data analysis is one of the key aspects in understanding the behavior of network attacks. The success of Honeynet systems depends in successfully detecting the attack and understanding the behavior of the attack. The main focus of this thesis work is to use an anomaly detection technique to detect anomalies and determine the anomaly type. In this thesis work an anomaly detection approach based on feature-based parameters and volume-based parameters is addressed. The entropy distributions of feature-based parameters and volume change of volume-based parameters were used to identify the best feature sets and to define the anomaly behavior.

A number of tests were carried out on various test datasets and the best features suitable for honeynet traffic were selected. The selected features were:

- Destination IP Entropy (DIP)
- Destination Port Entropy (DP)
- Source Port Entropy (SP)
- Total Payload Bytes (TB)

- Total Packet Count (PC)

Using the test data sets and the selected features the anomaly behavior was proposed. The threshold levels of the selected features were used to detect anomalies in live traces. Also K-means clustering plot was used to identify the different clusters available in the traces. A detection rate of 90.25% was achieved using the proposed anomaly detection technique. The predicted behavior set was used to identify the types of anomalies in the traces. A recall percentage of 40.32% and precision percentage of 80.65% was achieved using the proposed behavior detection approach.

The main Contributions of this thesis work are:

- Identifying best features suitable for Honeynet traffic.
- Using packet header details and volume changes to identify anomalies
- Identifying anomaly behaviors using reverse mapping or pattern matching.
- Identifying anomalies in Honeynet Traffic with a detection rate of 90%.

# 5.1 Future Research

The proposed anomaly detection technique can be extended to cover the following topics:

- A large database of the different anomaly behaviors can be created by using many test data sets. This will increase the recall percentage and will enhance the anomaly behavior detection capabilities.
- The anomaly points in the trace could be used along with some reverse engineering techniques to identify the actives of the anomaly. For example if bruteforce anomaly occurred then using some reverse engineering technique the type of service that was attacked could be determined.

- The proposed anomaly detection technique can be used with some payload analysis techniques to determine the exact vulnerabilities that were exploited during the attack.

# Bibliography


1. Levine, J., et al. *The use of Honeynets to detect exploited systems across large enterprise networks*. in *Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society*. 2003.
2. Spitzner, L., *Honeypots: Tracking Hackers, Addison-Wesley, http://www.tracking-hackers.com/book/*. 2003.
3. Spitzner, L., *Honeypots, definitions and value of honeypots. http://www.spitzner.net/honeypots.html*. May 2003.
4. Watson, D. and J. Riden, *The Honeynet Project: Data Collection Tools, Infrastructure, Archives and Analysis*, in *2008 WOMBAT Workshop on Information Security Threats Data Collection and Sharing*. 2008.
5. Viecco, C. *Improving Honeynet Data Analysis*. in *Information Assurance and Security Workshop, 2007. IAW '07. IEEE SMC*. 2007.
6. Chandola, V., A. Banerjee, and V. Kumar, *Anomaly detection: A survey*. ACM Comput. Surv., 2009. **41**(3): p. 1-58.
7. *Information Entropy, [Online] http://www.absoluteastronomy.com/topics/Information_entropy*.
8. Honeynet.org. *Honeynet Project, Honeynet Definitions, Requirements, and Standards Documentation, Honeynet Project website (http://old.honeynet.org/alliance/requirements.html)*.
9. Talabis, R. *The Philippine Honeynet Project, The Gen II & Gen III Honeynet Architecture*.
10. Balas, E. and C. Viecco, *Towards a third generation data capture architecture for honeynets*, in *Information Assurance Workshop, 2005. IAW '05. Proceedings from the Sixth Annual IEEE SMC* 2005. p. p 21 - 28.
11. Honeynet.org, *Know your enemy sebek. http://project.honeynet.org/papers/sebek.pdf,*. 2003.
12. Gong, D.F., *Deciphering Detection Techniques: Part II Anomaly-Based Intrusion Detection* 2003.
13. Dainotti, A., A. Pescape, and G. Ventre. *NIS04-1: Wavelet-based Detection of DoS Attacks*. in *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*. 2006.
14. Haggerty, J., et al. *DiDDeM: a system for early detection of TCP SYN flood attacks*. in *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*. 2004.
15. Ping, D. and S. Abe. *Detecting DoS attacks using packet size distribution*. in *Bio-Inspired Models of Network, Information and Computing Systems, 2007. Bionetics 2007. 2nd*. 2007.
16. Barford, P., et al., *A signal analysis of network traffic anomalies*, in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. 2002, ACM: Marseille, France. p. 71-82.
17. Lakhina, A., M. Crovella, and C. Diot, *Mining anomalies using traffic feature distributions*, in *Proceedings of the 2005 conference on Applications, technologies,*

*architectures, and protocols for computer communications*. 2005, ACM: Philadelphia, Pennsylvania, USA. p. 217-228.
18. Nychis, G., et al., *An empirical evaluation of entropy-based traffic anomaly detection*, in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*. 2008, ACM: Vouliagmeni, Greece. p. 151-156.
19. Kind, A., M.P. Stoecklin, and X. Dimitropoulos, *Histogram-based traffic anomaly detection.* Network and Service Management, IEEE Transactions on, 2009. **6**(2): p. 110-121.
20. Thonnard, O. and M. Dacier, *A framework for attack patterns' discovery in honeynet data*, in *Digital Investigation*. 2008. p. S128-S139.
21. Al-Haidari, F. *An Entropy-Based Countermeasure against Intelligent DoS Attacks Targeting Firewalls*. 2009.
22. Spathoulas, G.P. and S.K. Katsikas, *Reducing false positives in intrusion detection systems.* Computers & Security, 2010. **29**(1): p. 35-44.
23. Barford, P., et al., *Employing Honeynets For Network Situational Awareness*, in *Cyber Situational Awareness*, S. Jajodia, et al., Editors. 2010, Springer US. p. 71-102.
24. García-Teodoro, P., et al., *Anomaly-based network intrusion detection: Techniques, systems and challenges.* Computers & Security. **28**(1-2): p. 18-28.
25. Honeynet.org. *http://www.honeynet.org/challenges*.
26. hack.lu, *Information Security Visualization Contest, hack.lu 2009,* *http://2009.hack.lu/index.php/InfoVisContest*. 2009.
27. Lu, W. and A.A. Ghorbani, *Network anomaly detection based on wavelet analysis.* EURASIP J. Adv. Signal Process, 2009. **2009**: p. 1-16.
28. *DARPA 99* *http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html*. 1999.
29. Lakhina, A., M. Crovella, and C. Diot, *Diagnosing network-wide traffic anomalies.* SIGCOMM Comput. Commun. Rev., 2004. **34**(4): p. 219-230.
30. BackTrack. *Penetration Testing Distribution (http://www.backtrack-linux.org/ ).*
31. Metasploit. *Metsploit Framework - Metasploit Project web page,* *http://www.metasploit.com*.
32. Fyodor.: (2007) *Top 100 Network Security Tools (last visited, July 25, 2007), available on line on http://sectools.org*.
33. Laskov, P. and M. Kloft, *A framework for quantitative security analysis of machine learning*, in *Proceedings of the 2nd ACM workshop on Security and artificial intelligence*. 2009, ACM: Chicago, Illinois, USA. p. 1-4.
34. Rieck, K. and P. Laskov, *Language models for detection of unknown attacks in network traffic*. Journal in Computer Virology, 2007. **2**(4): p. 243-256.
35. Düssel, P., et al., *Cyber-Critical Infrastructure Protection Using Real-Time Payload-Based Anomaly Detection*, in *Critical Information Infrastructures Security*, E. Rome and R. Bloomfield, Editors. 2010, Springer Berlin / Heidelberg. p. 85-97.
36. Olson, D.L. and D. Delen, *Advanced Data Mining Techniques*. 1st ed. 2008: Springer Publishing Company, Incorporated.

# Curriculum Vita

## Syed Naeem Firdous


- **Email**: naeemfirdous@gmail.com
- **DOB**: 23/11/1982
- **Nationality**: Indian
- **Parmament Address**: No.2 Anna Street, Hari Homes, Gandhi Nagar, Saligramam, Chennai, INDIA – 600093
- **Contact No**: 0552538519


**Education**

- Masters (M.S) in Computer Networks, Computer Engineering department, KFUPM, Dhahran, Saudi Arabia - 2008 to 2010.
- Bachelors of Engineering (B.E) in Electronics and Communication Engineering, ECE department, Dr. M.G.R Engineering College, Chennai, India – 2001 to 2005.

**Experience**

- Programmer Analyst – Cognizant Technology Solutions, Chennai, India, Feb 2007 to Oct 2008

**Research Interests**

- Computer Network Security and Intrusion Detection systems
- Anomaly Detection Techniques
- Network Management
- Network Design