SECURITY AND COMMUNICATION NETWORKS
*Security Comm. Networks* 2013; **6**:567–583
Published online 3 July 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.575

RESEARCH ARTICLE

# Classifying malicious activities in Honeynets using entropy and volume-based thresholds

Mohammed H. Sqalli[1]*, Syed Naeem Firdous[1], Khaled Salah[2] and Marwan Abu-Amara[1]

[1] Computer Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[2] Computer Engineering (Sharjah Campus), Khalifa University of Science, Technology and Research (KUSTAR), PO Box 573, Sharjah, UAE

## ABSTRACT

A Honeynet is a network designed by the Honeynet Project organization to gather information on security threats and attacks. Honeynets are being used by numerous institutions to proactively improve network security by identifying malicious and unauthorized activities in production and private networks. A Honeynet captures a substantial amount of network data and logs. The analysis of these datasets to identify malicious activities is a challenging task. The main aim of the work in this paper is to employ an anomaly detection technique to classify different types of malicious activities present in Honeynet. In particular, we use feature-based and volume-based schemes for Honeynet data classification. A detailed analysis of various traffic features is carried out, and the most appropriate ones for Honeynet traffic are selected. The classification of malicious activities is achieved by applying entropy-based distributions and traffic volume distributions. Entropy-based distributions are used for feature-based parameters, whereas traffic volume distributions are used for volume-based parameters. The behavior of various anomalies or malicious activities is classified using the selected features and their respective threshold values. Finally, we propose a mapping between the various anomalies and their associated behavior, which can be further used to identify similar anomalies in other Honeynet data sets. Copyright © 2012 John Wiley & Sons, Ltd.

**KEYWORDS**
security; honeynet; malicious, entropy; classification

***Correspondence**
Mohammed H. Sqalli, Computer Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia.
E-mail: sqalli@kfupm.edu.sa

## 1. INTRODUCTION

Computer network security is a major area of concern for many people from normal home users to businesses trying to protect their resources from unauthorized access. When a computer is connected to the Internet, it is physically connected to millions of other computers in the network. There is a 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 (DoS) attacks, worms, and Trojans.

### 1.1. Honeynets

A Honeynet is a solution designed to gather information on security threats, and it can be used by 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 are data control, data capture, and data analysis [1]. The Honeynet Project offers awareness, information, and tools to help organizations to set up and implement a Honeynet in their networks. The Honeynet is an effective concept that can be used to understand the threats that exist in the networks. It provides tools such as Honeywall and other data capture and data analysis tools to learn about the vulnerabilities in networks. The Honeynet architecture comprises different honeypots and various other tools. 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: high interaction and low interaction [3]. 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 rootkits to international Internet relay chat (IRC) sessions. Honeywall, Sebek, and CaptureHPC are some of the examples of high-interaction honeypots [2]. On the other hand, 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 hacker's specific activity. Dionaea, Honeyd, Nepenthes, and Google Hack are some of the examples of low-interaction honeypots. Figure 1 shows a sample network layout with different types of honeypot implementations.

Currently, a Honeynet gathers a large amount of network data, and this sometimes makes it difficult to analyze. Various types of data are collected on the basis of which Honeynet tool is used, for example, Honeywall, Nepenthes, HoneyD, and Dionaea, and each tool uses its own format for data representation and storage. For instance, the Honeywall is a high-interaction Honeynet that has a built-in firewall, intrusion detection system (Snort), and Hflow daemon. The Honeywall acts as a layer 2 bridged gateway and is designed using a minimized Linux distribution [4]. It also has a kernel-level module, which collects keystrokes and other activities in the honeypot. Apart from these, the Honeywall also captures the packets and stores them using the PCAP format. The Honeywall runs a daemon known as Hflow, which collects data from different sources and stores them in a MySql database. The information collected in the database includes the following information:

- Five tuples (source and destination addresses, source and destination ports, protocol).
- Snort intrusion detection system responses—gives the relative threat level and also generates alerts.
- Passive OS fingerprinting—identifies the attackers' OS.
- Total bytes transferred.
- Sebek data—data sent by the sebek client, which captures the host activity.

The various honeypot implementations result in the collection of huge data of different types such as packet captures, tcpdump data, malicious binaries, keystroke logs, and URLs of malicious websites [5]. The raw data collected from a Honeynet can be used to provide further insights into the hacker's activities. However, it becomes difficult to analyze the captured data without the use of automated analysis tools. The "needless stack" data overload, that is, too much data and different types of data, is one of the main challenges for Honeynet analysts [6]. Honeynets are now used widely by many researchers and network operators to understand the vulnerabilities in the network. However,

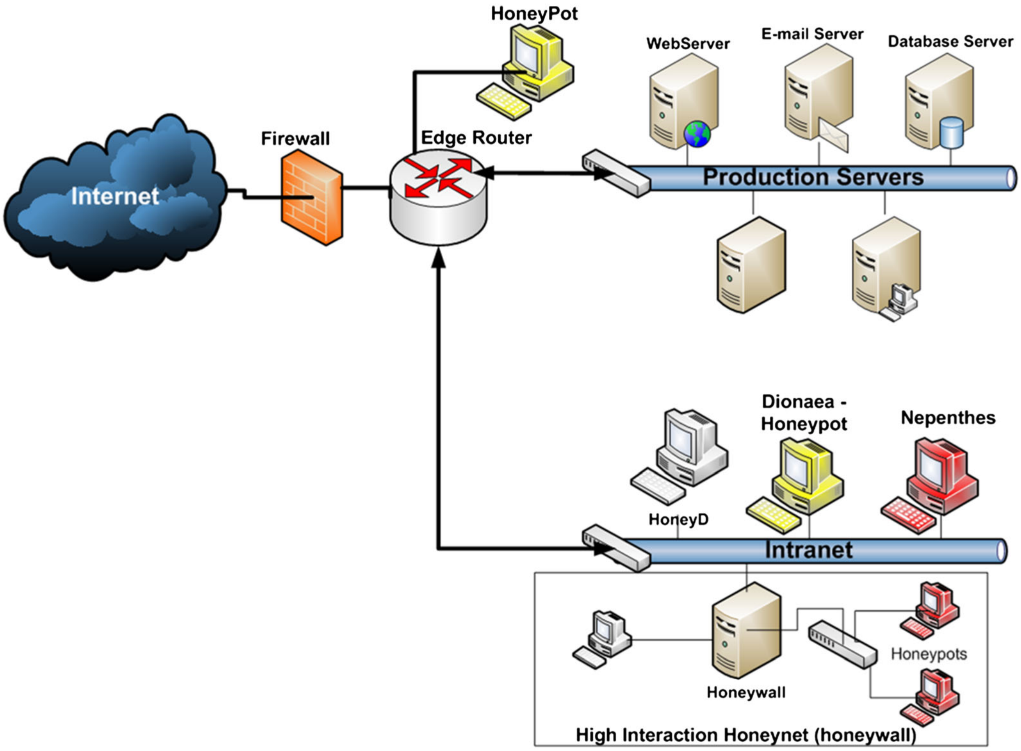

**Figure 1.** Various types of honeypot implementations.

honeypots collect a large amount of data from various data sources, making it difficult to manage honeypots and to understand the collected data [7]. In addition, a Honeynet's real potential will not be realized until organizations can effectively deploy multiple Honeynets and correlate the information they collect.

## 1.2. Anomaly detection

Anomaly Detection refers to a technique for 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 within any data set are referred to as anomalies, outliers, exceptions, peculiarities, and so on [8]. Figure 2 shows the regions that are labeled as normal or outliers.

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, and so on. Because of this, most of the existing anomaly detection schemes tackle only a specific problem[8].

In information theory, entropy is defined as a measure of uncertainty or randomness associated with a random variable [6] or in this case data coming to a Honeynet. 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 values. The entropy values of a sample of size $n$ lies in the range $[0\ , \log n]$. The entropy takes the minimum value of 0 when there is no variation in the data items, for example, single IP address or port, and it 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 $\{x_1, x_2, x_3, \ldots, x_n\}$ can be calculated as follows:

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

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 $x_i$. Therefore,

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

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

where $H(X)$ is the entropy, $m_i$ is the number of packets with $x_i$ as the traffic feature, and $m$ is the total number of packets.

The probability of occurrence of a traffic feature value in the observed traffic is computed as follows:

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

Here, the total number of packets $m$ is the number of packets seen within a time window of fixed size $T$. More details about how the time window $T$ is defined and what value is used for it will be provided later in this paper.

The current Honeynet 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 [9]. The main contribution of the work in this paper is to evaluate different candidate features and use the best ones and their corresponding threshold levels to classify the different malicious activities or anomalies seen in Honeynets.

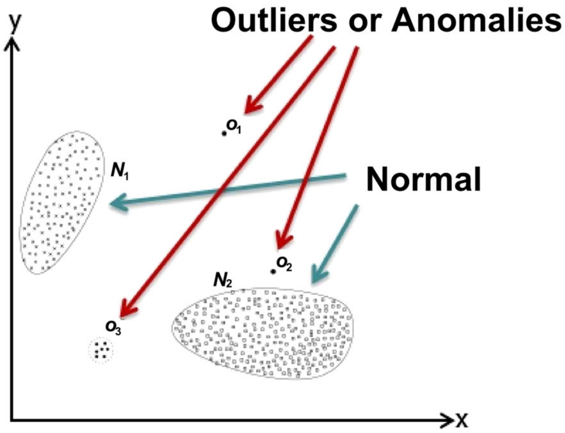

**Figure 2.** Anomalies or outliers [8].

The rest of the paper is organized as follows. Section 2 discusses the related work, in which we briefly summarize the existing research on using feature-based parameters and volume-based parameters for anomaly detection. Section 3 discusses our proposed approach including the data sets analysis and features' evaluations. Section 4 presents the classification of malicious activities on the basis of some defined thresholds for different features. Section 5 presents the conclusions and discusses future work.

## 2. RELATED WORK

The success of a Honeynet mainly depends on the way the data is collected and analyzed to better understand the vulnerabilities in the network. 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 the following [10]:

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

There exist in the literature two main categories of detection techniques applied to network traffic, that is, volume-based detection techniques and feature-based detection techniques.

- *Volume-based detection techniques* [11–14]: 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 techniques.
- *Feature-based detection techniques* [15,16]: The feature-based detection scheme uses the distributional changes of packet header details, such as 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.

Patcha and Park [17] provided a survey of anomaly detection systems that focused on statistical anomaly detection schemes, data mining-based methods, and machine learning-based techniques. As stated by the authors, systems that use the statistical anomaly detection approaches do not require prior knowledge of security flaws and/or the nature of the attacks. Such a feature allows the system to detect "zero day" and the latest attacks. Another feature of the statistical approaches is that they can provide alerts of malicious activities such as DoS attacks, which typically occur over extended periods. On the other hand, statistical anomaly detection approaches can be circumvented by skilled attackers who can train such systems to accept abnormal behavior as normal. Another disadvantage of such approaches is that it can be difficult to determine thresholds that balance the likelihood of false positives with the likelihood of false negatives. Moreover, such approaches need accurate statistical distributions, but not all attacks can be modeled using purely statistical methods. Examples of statistical anomaly detection approaches include Haystack, Next-Generation Intrusion Detection Expert System, and Statistical Packet Anomaly Detection Engine. In contrast, machine learning-based anomaly detection approaches have the ability to change their behavior as a result of acquiring new information. Accordingly, machine learning approaches build a system that improves its performance through the use of already collected results. Examples of such approaches include system call-based sequence analysis, Bayesian networks, principal components analysis, and Markov models. On the other hand, data mining-based anomaly detection approaches focus on identifying bounds for valid network activity so as to help in distinguishing attack activity from normal traffic on the network. Examples of these approaches include classification-based intrusion detection, clustering and outlier detection, and association rule discovery.

Lakhina *et al*. [15] proposed an anomaly detection method using traffic feature distributions in which they argue that distributions of packet features, such as 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, and it also helps in clustering the anomalies into different groups. In their experiment, they used network-wide 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. They pointed out that most of the anomalies affect the distributional aspects of traffic features such as IP addresses and port numbers. The main difference between the method used by Lakhina *et al*. [15] and the previous work is that they used distributions of traffic features, such as IP address and ports,] to detect anomalies compared with using traffic volume. They noted that not all anomalies cause volume changes in traffic but most of them 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 destined only to a Honeynet, and we are using both traffic feature distributions and volume parameters to detect anomalies and classify malicious activities.

Nychis *et al*. [16] 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 IP address and port distributions. The authors calculated the correlation between different feature pairs on the basis of the entropy values to find the correlated feature pairs. They 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 for Honeynet traffic. The traffic features were compared and the best ones were chosen using the test data sets to classify the behavior of different types of malicious activities.

Kind *et al*. [18] proposed a new approach to the feature-based anomaly detection of Lakhina *et al*. [15]. 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 such as source and destination addresses, port numbers, and Transmission Control Protocol flags. In this approach, PCA has been used for dimensionality reduction instead of differentiating between normal and abnormal traffic as performed by Lakhina *et al*. [15]. The main difference of this approach is in the use of histograms to detect anomalies instead of using entropy. In our proposed work, we are using entropy values of different features along with the *k*-means clustering technique to identify anomalies in Honeynet traffic compared with using histogram patterns for clustering.

Ping and Abe [13] proposed an IP packet size entropy (IPSE)-based DoS detection scheme in which changes in the 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. This is because various services have default packet sizes on the basis of the service provided. For example, File Transfer Protocol (FTP) applications have 40-B acknowledgement and full packet data of 1500 B. 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 attacks 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 to identify the anomalous behavior.

Thonnard and Dacier [19] proposed a clustering-based approach to detect attack patterns in Honeynet data. In their approach, they specifically used 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. They conducted experiments on large data sets collected from 44 worldwide distributed honeypots. The attack source is identified as an IP address that targets the honeypot on a given day with a certain port sequence. The network characteristics used by the authors include the following: (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 classify malicious activities in Honeynet data compared with using time signature for clustering Honeynet data.

Al-Haidari *et al*. [20] proposed an entropy-based countermeasure against DoS attacks on firewalls. In their work, they used packet size entropy and the corresponding threshold values to distinguish between normal traffic and attack traffic. They have also 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.

François *et al*. [21] compared two ways of collecting malicious network traffic, by monitoring the activity in large Honeynets and in network telescopes. For instance, they presented results related to the distribution of source addresses, where it was found that a Honeynet is sufficient for learning such information. They have also found that both methods provide similar results about the services/ports that are attacked. They have mainly focused on what can be captured or not in terms of malicious activity, for example, identifying the source IP addresses that are not captured by other honeypots. Another aspect of the authors' work is on detecting misconfigurations such as the analysis of

DOI: 10.1002/sec

whether some IP addresses used on the Internet are not appropriate for such use, for example, IP addresses belonging to ranges reserved for private allocations. In addition, their proposed work allowed finding out which services are the most attacked services. On the other hand, the objective of our work is on classifying the malicious activities in Honeynet traffic on the basis of the distribution of various features. Therefore, it is clear that the work by François *et al*. [21] is different from ours as they are using these distribution results for the purpose of determining the usefulness of having Honeynets as a way to monitor malicious activities. We go a step further by analyzing Honeynet traffic for the purpose of classifying malicious activities within.

## 3. ANOMALY DETECTION APPROACH

A Honeynet captures information that can be used by administrators to improve their network security, but the size of the data collected can be overwhelming [22]. Honeynets mainly depend on a signature-based detection scheme, manual analysis, and expertise to identify malicious activities. 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 the different malicious activities in this traffic, manual analysis and expertise are needed.

Honeynet traffic is different from other types of network traffic as every packet that enters or leaves the Honeynet is considered malicious. With this fact, we consider that anomalies that are classified as belonging to a given type are all malicious in nature. Nonetheless, analyzing Honeynet data to identify malicious events is a challenging task and consumes much time. Traffic collected by a Honeynet includes attack traffic, broadcast traffic, probes, and traffic from other local machines, which may not be always malicious, such as network discovery packets coming from windows-based machines. The diversity in the traffic collected by a Honeynet and the real nature of all such traffic (note that an attacker is unaware of the presence of a Honeynet), imply that novelty in analysis of such data is essential to achieve high rates of detection with low false alarms. With our Honeynets traffic analysis, we also found that significant changes in Honeynet traffic occurred only during malicious events, which essentially serves to identify anomalous activities within a given traffic profile.

There are very few anomaly detection techniques addressing the Honeynet systems' needs. Most of the Honeynet 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 set up across the world. Because in a Honeynet, most traffic that enters or leaves is considered malicious, other anomaly detection approaches applied to regular network-wide traffic are not well suited for this type of traffic [19]. In addition, we have used existing Honeynet PCAP traces for testing our proposed anomaly detection technique. Although other PCAP traces exist in the literature, these were for non-Honeynet networks. Hence, these traces cannot be used for testing our proposed anomaly detection technique because the nature of the traffic in non-Honeynet networks is different from that of the traffic in Honeynets. On the other hand, there exist other Honeynet traces but using different format from PCAP and are therefore not suitable for comparison.

In order to address these issues, we propose a simple and easy-to-use anomaly detection technique that can be used to identify malicious activities in Honeynet traffic and, also, to classify the behavior of various malicious activities. This paper addresses specifically the classification part which focuses on mapping the different malicious activities to certain features' behavior using thresholds. The identification of malicious activities in other Honeynet data sets will be the subject of a work to follow.

We propose an anomaly detection technique that uses both feature-based and volume-based parameters to identify anomalies in the Honeynet traffic. The proposed approach uses a combination of packet header parameters entropies and volume changes to identify malicious activities.

Our 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. These features will be taken from both those available in the literature as well as those obtained from a manual data analysis.
(3) Devising and implementing a suitable anomaly detection technique.
(4) Classifying malicious activities in Honeynet data on the basis of the values (or ranges/thresholds) of the different features used by the proposed anomaly detection technique.

Figure 3 gives an overview of the proposed solution to classify malicious activities in Honeynet traffic. The Honeynet project is a useful resource to learn the tools, motives, and tactics of the blackhat community. Using the proposed anomaly detection scheme in Honeynets will greatly improve the data forensics and the detection of unknown and new attacks. Although the focus of this paper is on the classification of malicious activities in Honeynets, the ultimate objective is to be able to identify similar malicious activities in any Honeynet traffic, including large data sets, which can then be filtered out to focus more on new types of attacks (or zero-day attacks).

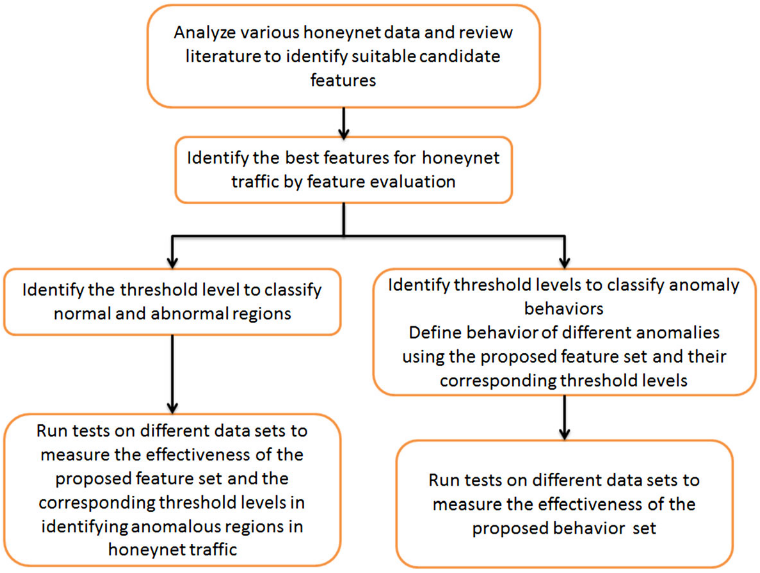
Analyze various honeynet data and review literature to identify suitable candidate features
Identify the best features for honeynet traffic by feature evaluation
Identify the threshold level to classify normal and abnormal regions
Identify threshold levels to classify anomaly behaviors
Define behavior of different anomalies using the proposed feature set and their corresponding threshold levels
Run tests on different data sets to measure the effectiveness of the proposed feature set and the corresponding threshold levels in identifying anomalous regions in honeynet traffic
Run tests on different data sets to measure the effectiveness of the proposed behavior set

**Figure 3.** Proposed solution for classifying malicious activities in Honeynet traffic.

### 3.1. Honeynet test data

In order to identify anomalies in Honeynets, we first need to analyze different Honeynet data sets to understand the difference between normal and abnormal behaviors. Honeynet traces were collected mainly from the honyenet.org site, which includes the scan-of-the-month challenges and forensic challenges released by the Honeynet Project organization [23]. The other source of traces is the hack.lu 2009 Information Security Visualization Contest [24]. The Honeynet traces that were used are listed in Table I.

The traces provided by the Honeynet Project organization are instances of real compromises that were captured by different Honeynet Project chapters. The main reasons for releasing such challenges are to help the network security analysts to hone their forensic and analysis skills to attain 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 carried out using tools such as Wireshark [25] and NetMiner [26].

The lists of features that were recorded from the literature and identified during test data analysis are stated in Tables II and III.

Some of the features that provided redundant information were eliminated such as the application protocol because it is related to the port used. Similarly, instead of using the average packet sizes for different transport protocols, we choose the average payload size. A summary of the traffic features used for further analysis of the Honeynet traffic is presented in Table IV.

**Table I.** Honeynet traffic test data sets used for analysis.

| Traffic data set name and 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 Security Authority Remote Procedure Call service. | 348 packets<br>Total duration: 16 s |
| Scan 28, Honeynet.org—scan of the month | This trace was collected by the Mexico Honeynet Team—Italian blackhats break into a Solaris server then enable IPv6 tunneling for communications. | Two traces:<br>Day 1: 18 843 packets—24 h<br>Day 3: 123 123 packets—24 h |
| Scan 14, Honeynet.org—scan of the month | This trace is about a successful Windows NT attack. | 6707 packets<br>Total Duration: 20 h |
| Scan 19, Honeynet.org—scan of the month | This is a trace of Redhat Linux 6.2 honeypot compromise. | 24 440 packets<br>Total Duration: 23 h |
| SSH-based honeypot trace—Information Security Visualization Contest—hack.lu 2009 | This dataset was collected from an SSH-based honeypot. It includes anomalies such as network scans, rootkit file transfers, and IRC traffic. | 4 323 191 packets<br>Total Duration: 12 days |

**Table II.** List of feature-based parameters selected from test data analysis and literature.

| Traffic feature-based parameters | Description |
|---|---|
| Source IP address entropy [15] | This parameter indicates the entropy of the unique IP addresses of incoming connections to the honeypot. |
| Destination IP address entropy [15] | The destination IP entropy indicates the number of external connections initiated by the honeypot. |
| Source port entropy [15] | This attribute indicates the number of source ports that are visible during each interval. |
| Destination port entropy [15] | This parameter indicates the number of destination ports visible during each interval. |
| Indegree [16] | Number of distinct Hosts that connect to the observed host. This parameter indicates the number of incoming connections to the honeypot. |
| Outdegree [16] | Number of distinct IP address the observed host connects to. This feature measures the number of outgoing connections from the honeypot. |
| Packet size entropy [13] | Various packet sizes visible in the network traffic. |
| Application protocol used | Application protocol seen during a conversation (e.g., SSH, SMTP, and FTP). |
| Origin of IP address—country | The distribution of countries from which the observed host gets connections. |

**Table III.** List of volume features selected from test data analysis and literature.

| 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 packet size average. |
| Total payload bytes | Total bytes seen in the 5-min interval. |
| Average inter-arrival times | Average inter-arrival time of packets in 5-min interval. |
| Average payload size | Average packet size seen during the 5-min interval. |
| Total packets | Total packets seen during the 5-min interval. |

TCP, Transmission Control Protocol; UDP, User Datagram Protocol; ICMP, Internet Control Message Protocol.

**Table IV.** Traffic features used for a detailed analysis.

| Traffic features | Volume features |
|---|---|
| • Source IP address | • Average packet inter-arrival time |
| • Destination IP address | • Total payload bytes received during the interval |
| • Source port | • Average payload size during the interval |
| • Destination port | • Average number of packets received during the interval |
| • Packet size distribution | |
| • Indegree and outdegree | |

### 3.2. Data sets analysis and features' evaluations

The real Honeynet traces obtained from Honeynet.org were used to test the effectiveness of each individual feature. The candidate features were evaluated on the basis of the traffic distributions seen during the anomalous events. The features were also evaluated on the basis of their ability to differentiate between normal and abnormal traffic. The entropy distributions were obtained by calculating the entropy values of each feature for every 5-min interval. Figure 4 shows the sliding window concept that was used to gather entropy values in overlapping intervals so that any valuable information is not missed in cases where an anomaly overlaps across multiple intervals.

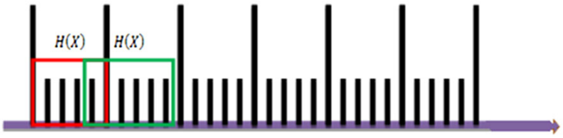

**Figure 4.** Sliding window used for calculating entropy.

The entropy values of each feature were recorded, and further manual analysis of the trace was performed to identify the normal behavior and anomalous behavior. Initially, all the features listed in Table IV were tested, and later, the best features that provide better detection capabilities were selected.

#### 3.2.1. Data set: Scan 28.

This data set was published in the SOM challenges in the Honeynet.org website. The trace was collected by the Mexico Honeynet Team, and it is about Italian blackhats that broke into a Solaris server and then enabled IPv6 tunneling for communication. It is composed of 2 days of collected traffic, that is, Day 1 and Day 3. The Day 1 traffic is about the honeypot being compromised, and the Day 3 traffic consists of the IPv6 tunneling enabled by the blackhats for communication.

#### 3.2.1.1. *Day 1 traffic.*

The destination port entropy (DP) of Day 1 traffic does not show much activity in the first 9 h after which there is a drastic change in the traffic behavior as shown in Figure 5. When we check the volume feature, that is, the total packets in the interval after the ninth hour, it is clear that there was a malicious activity as shown in Figure 6. The manual analysis of the PCAP trace reveals that the honeypot was probed for a specific vulnerability and then compromised during this time. A similar analysis was performed for other features to identify those that had better detection capabilities. The features that gave a clear indication of anomaly are DP, source port entropy (SP), total payload bytes (TB), and total packets. The packet size entropy also showed the change in behavior, but it does not help in understanding the anomaly behavior.

#### 3.2.1.2. *Day 3 traffic.*

The Day 3 traffic shows less activity in the initial hours, but around the sixth 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 SP plotted in Figure 7 shows a drastic increase in the entropy value around the 15th hour.

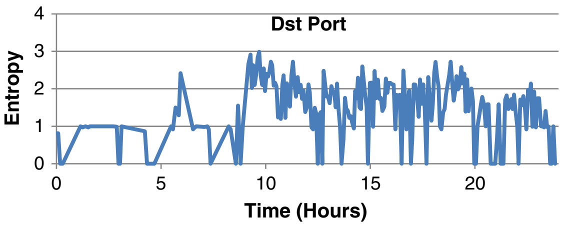

**Figure 5.** Destination port entropy in Day 1 traffic of Scan 28 data set.

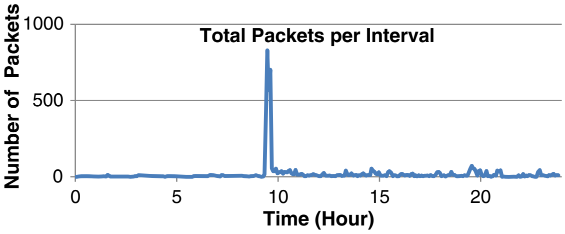

**Figure 6.** Total packets per interval in Day 1 traffic of Scan 28 data set.

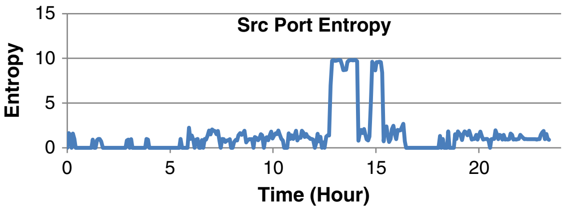

**Figure 7.** Source port entropy in Day 3 traffic of Scan 28 data set.

Also, a port scan activity was recorded, which can be attributed to the peak in the SP.

The destination IP entropy (DIP) and outdegree do not give a very clear picture of the changes in the traffic. The dominant features that were helpful in detecting the malicious events in this trace are the SP, DP, TB, and total packets.

The entropy values are calculated based on the formula mentioned earlier, that is,

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

Because we use a sliding window approach, we calculate the entropy for every 5-min window for the entire trace. Then, we use the entropy versus time plots to visualize the entropy variations for the entire packet capture duration. Two examples of the entropy calculations are presented later. If there is one dominant value during a 5-min window, then the entropy will be close to zero. For example, if there was only HTTP traffic in a 5-min window, then

$$n = 1 \text{ and } probability\ (packets\ with\ port\ 80) = 1$$
$$\Rightarrow entropy = -(1 * \text{‘}\log 1) = 0$$

Similarly, for an interval in which different ports were seen, the entropy will be higher. For instance, if there are 100 packets during a port scanning event with 100 different ports, then:

$$n = 100, probability\ (port\ A) = \frac{1}{100} = 0.01,$$
$$and\ probability\ (port\ B) = \frac{1}{100} = 0.01$$
$$\Rightarrow entropy = -\sum\nolimits_{n=0}^{n=100} 0.01 * \log 0.01 = 2$$

### 3.2.2. Data set: Scan 14.

This trace is about a successful Windows NT machine attack. The attacker exploited a vulnerability in Microsoft® Data Access Components that could allow a web site visitor to take unauthorized actions on a web site hosted using the Internet Information Server. The DP plotted in Figure 8 shows a different behavior during the period when the target machine was being compromised. The volume feature TB plotted in Figure 9 shows the intervals when large data or files were transferred to the target machine. Both total packets and TB show a large variation when some data transfer took place.

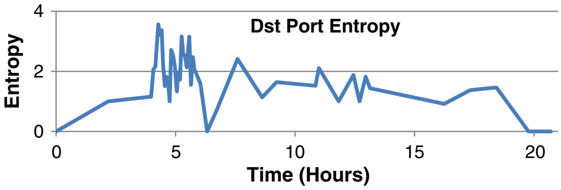

**Figure 8.** Destination port entropy for Scan 14 challenge.

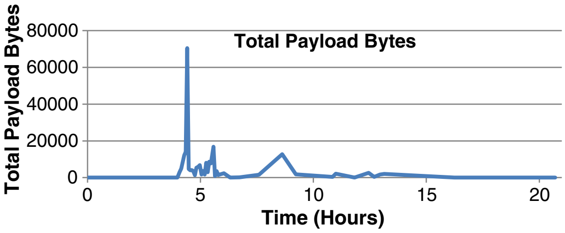

**Figure 9.** Total payload bytes for Scan14 challenge.

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

### 3.2.3. 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 DP plotted in Figure 10 shows that there was not much traffic for nearly 20 h 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 importance of volume features is clear in this trace as they help in understanding the attacker's behavior during a system exploit. The other parameters, such as outdegree shown in Figure 11 and indegree, are not very useful in giving a good understanding of the behavior.

### 3.2.4. Data set: Secure Shell-based honeypot traffic.

The feature analysis tests were also carried out on a large data set collected from a Secure Shell (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, and IRC traffic. The DIP shown in Figure 12 indicates the number of external connections initiated by the honeypot. The peaks indicate that the honeypot initiated a large number of connections during that interval. The high value of DIP indicates that the honeypot was scanning the network.

The indegree shown in Figure 13 does not show all the anomalies, and because of this fact, this feature was not selected for anomaly detection.

Volume-based features such as TB also helped in understanding the behavior and the anomalous events. Figure 14 shows that, before a network scan event begins, 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.

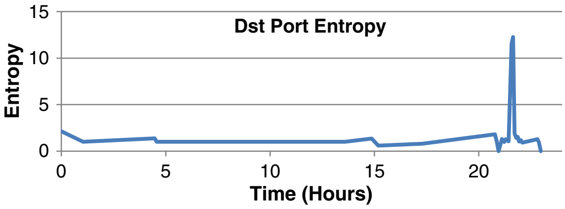

**Figure 10.** Destination port entropy for Scan 19 challenge.

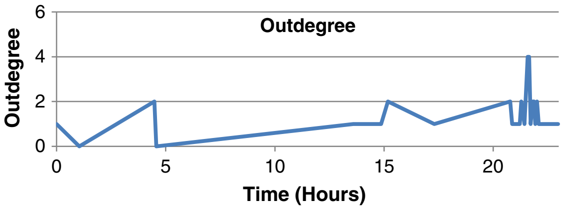

**Figure 11.** Outdegree distribution in Scan 19 Trace.

## 3.3. Combining pairs of 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 have a better understanding of the behavior of the anomaly, we need to look into a combination of features. This is useful to detect

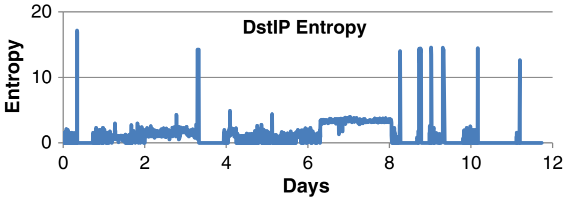

**Figure 12.** Destination IP entropy of SSH-based honeypot trace.

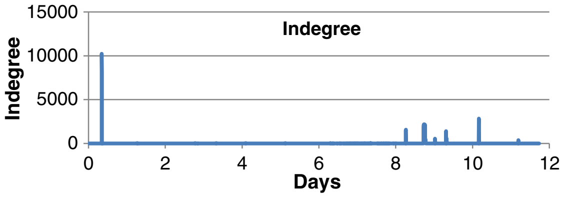

**Figure 13.** Indegree distribution of SSH-based honeypot trace.

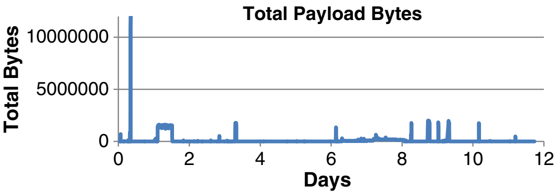

**Figure 14.** Total payload bytes distribution of SSH-based honeypot trace.

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 features' combinations and to have a better understanding of the anomalies.

The DIP and the DP show visible groups, that is, clusters, indicating events with similar behaviors. In Figure 15, the group with high DIP and low DP indicates a network scan where a large number of IP addresses are being scanned for the same port. The cluster with high DP and low DIP is related to a port scan activity.

The combination of source IP entropy and SP plotted in Figure 16 shows that during a network scan, the source IP address entropy value is small because only one IP was scanning the network.

### 3.4. Combining three features to detect visible anomalous groups

When combining different features, we can see different patterns that can help us detect anomalous regions as well as normal regions. Using three features helps in getting a better visualization of the different clusters present in the Honeynet data. We performed various tests using different combinations of the features to identify those features that provide the best distinction between a normal behavior and outliers by showing distinct clusters.

The combination of source IP, destination IP, and destination port is shown in Figure 17. This combination does not show many cluster regions because the source IP entropy and DIP have a similar behavior.

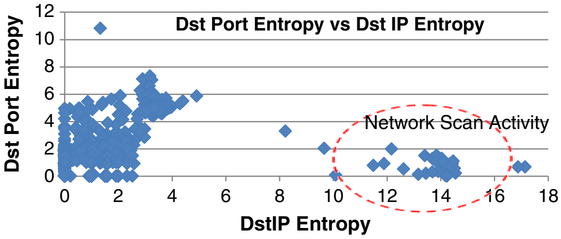

**Figure 15.** Destination port entropy and destination IP entropy combination of SSH honeypot trace.

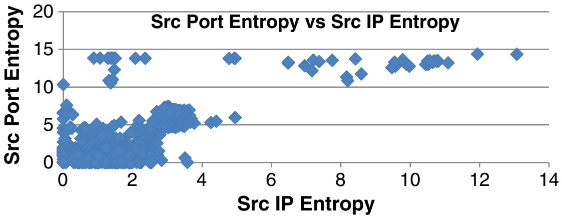

**Figure 16.** Source IP entropy and source port entropy combination of SSH honeypot trace.

The combination of source port, destination port, and destination IP entropies shows visible clusters, which can be attributed to different anomalous events. In Figure 18, cluster 1 includes a region having entropy values of 0–2.8 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, both the SP and the DIP are 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 SP is high and the DP is low as these attacks are targeting the SSH port. The fourth cluster indicates a 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, that is, the fifth cluster, mostly represents the IRC traffic as there are few machines communicating with each other using the IRC ports.

The actual behavior of different anomalies is explained in the following section. Table V summarizes the findings of feature analysis by providing the detection capabilities of various features.

After testing various combinations of traffic features, we conclude that the combination of DP, SP, and DIP provide better detection capabilities. On the other hand, the volume features, that is, TB 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 feature-based parameters, whereas these events were detected by the volume-based parameters. Therefore, instead of just using the feature-based techniques, we also need to use the volume-based techniques in order to detect most types of anomalies in a Honeynet.

## 4. MALICIOUS ACTIVITIES CLASSIFICATION

This section describes the method used to classify different malicious activities by using the selected features. The first step is to define the threshold levels for the selected features and then use these threshold levels to identify the behavior pattern of different anomalies.

### 4.1. Defining thresholds for different features

In our proposed approach, anomalies are identified using the five top-ranked features: DIP, DP, SP, TB, and total packet count (PC). The classification between normal and abnormal traffic is performed using the entropy and volume variations of the corresponding features. For example, the sample instances (which represent rows in Tables VI and VII) taken from Honeynet data collected from different sources indicate that during normal behavior, very small variations in either entropy or volume values are seen, as shown in Table VI. However, there are significant entropy and volume changes

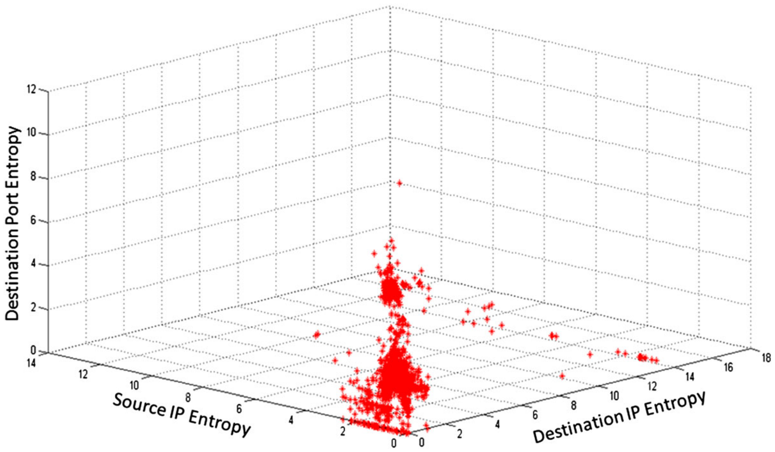

**Figure 17.** Combination of destination port, source IP, and destination IP entropy values.

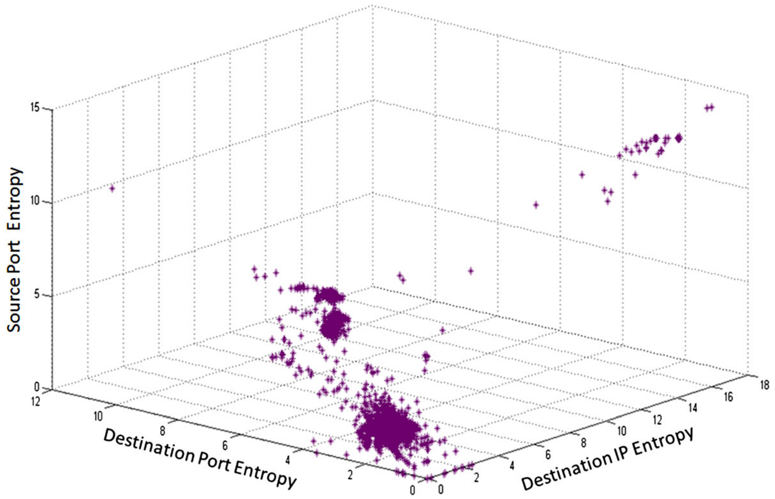

**Figure 18.** Combination of destination IP, destination Port, source port entropy values.

during the presence of anomalies or malicious activities, as shown in Table VII. With a thorough manual analysis of the training data sets, we found that during normal traffic, that is, traffic that is not part of malicious activities, the entropy-based features had an entropy variation in the range of 0–3. Similarly, for volume-based features, variations of normal traffic were in the range of 0–3000 B for the TB and 0–50 packets for the PC.

With the various entropy and volume variations seen in the Honeynet data, both during normal and anomalous traffic, threshold levels can be defined to distinguish between normal and abnormal traffic regions. The threshold levels are selected on the basis of the entropy values recorded during normal and abnormal events as indicated in Tables VI and VII. The data collected also shows that for different anomaly types, different entropy values have been recorded. For instance, in a port scan event, higher port entropy values have been recorded in comparison with other events. A thorough comparison of all entropy values recorded for the test datasets described in Table I was used to identify threshold levels in Honeynet traffic. The analysis of the entropy and volume changes recorded for the anomalies present in these datasets shows that entropy values greater than 3 are considered anomalous as indicated by the values reported in Tables VI and VII. Similarly, a volume change of the total bytes that is greater than 3 kB or that of the total packets that is greater than 50 kB is considered malicious. These values are initially used to classify Honeynet traffic into normal and abnormal regions, that is, detection of an anomaly.

In addition, various other threshold levels are defined on the basis of the entropy values and volume changes to identify the different types of malicious activities. These levels were obtained by analyzing the entropy and volume values of anomalous traffic in many traces, including those presented in Table VII. The behavior of different types of malicious activities can then be identified by the selected features and the associated threshold levels. For the purpose of easy mapping between types of malicious activities and threshold values, we define the following threshold levels:

**Table V.** 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 a good indication of an anomaly. |
| Source IP entropy | Shows less variations in the traffic compared with the 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. |
| 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. |

**Table VI.** Entropy and volume values for normal traffic.

| DIP | DP | SP | TB | PC |
|---|---|---|---|---|
| 0 | 1.310 | 1.31 | 228 | 6 |
| 1 | 1.520 | 0.98 | 444 | 4 |
| 0 | 1.870 | 2.04 | 2631 | 20 |
| 1 | 0.918 | 1.58 | 3 | 1 |
| 0 | 1.620 | 0.33 | 168 | 8 |

DIP, destination IP entropy; DP, destination port entropy; SP, source port entropy; TB, total payload bytes; PC, total packet count.

**Table VII.** Entropy and volume values for abnormal traffic.

| DIP | DP | SP | TB | PC |
|---|---|---|---|---|
| 0.52 | 3.56 | 4.46 | 12 118 | 152 |
| 0 | 3.22 | 4.20 | 13 971 | 138 |
| 0 | 3.37 | 3.61 | 70 497 | 185 |
| 17.14 | 0.67 | 14.33 | 141 048 | 5702 |
| 16.87 | 0.677 | 14.36 | 181 988 | 7023 |
| 0.419 | 11.55 | 11.53 | 374 099 | 4152 |
| 0.218 | 12.26 | 12.26 | 214 096 | 5374 |

DIP, destination IP entropy; DP, destination port entropy; SP, source port entropy; TB, total payload bytes; PC, total packet count.

*Very high entropy or very high volume*: This level is used for high entropy values and high volume of data. With the tests made on the traces, only few anomalies, that is, network scan and port scan, had high entropy values. The entropy values greater than 7 are considered as very high. Volume changes greater than 500 kB and packet count greater than 2000 packets are also considered very high.

*High entropy and high volume*: This level is used for entropy values that lie between 5 and 7. With the experimental results, it can be understood that certain anomaly types such as bruteforce attacks or fuzzers result in high entropy values. Certain anomalies have high entropy because they initiate too many connections from different ports to crack the passwords or exploit the vulnerabilities of different applications. Volume changes between 50 and 500 kB as well as packet count between 500 and 2000 packets are considered high.

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

*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 5-min interval, then an entropy value of zero is recorded. This level is used only for feature-based parameters and is not applicable to volume-based parameters. Also, the situation in which this level is considered an anomaly is when there is zero entropy for the three feature-based parameters and a medium volume change.

Table VIII summarizes the various levels used to identify the malicious activities' behaviors in the Honeynet traffic.

## 4.2. Classifying malicious activities based on features' thresholds

It is essential to analyze the behavior of the various malicious activities detected in the training data sets after having already defined the required features to detect such anomalous activities. This analysis of the behavior of the various malicious activities detected will help in recommending a classification of such activities on the basis of their behavior pattern. Hence, this section presents the various entropy and volume ranges, that is, threshold levels, that were recorded for different types of malicious activities found in the training data sets. These ranges were used to learn the behavior pattern of different malicious activities and to recommend a mapping between the features' thresholds and the types of such activities, hence

**Table VIII.** Threshold levels used for identifying malicious activities in Honeynets.

| Threshold level | Range |
|---|---|
| Very high entropy/volume | Entropy > 7<br>Bytes > 500 kB<br>Packet count > 2000 |
| High entropy/volume | 7 > entropy > 5<br>500 kB > bytes > 50 Kb<br>2000 > packet count > 500 |
| Medium entropy/volume | 5 > entropy > 3<br>50 kB > bytes > 3 kB<br>500 > packet count > 50 |
| Zero entropy | Zero entropy (also, there should be medium volume changes) |

providing a classification of malicious activities on the basis of the different features' threshold levels. The behavior pattern of each type of a malicious activity is defined using the five features that were selected earlier for anomaly detection.

With the analysis of the various training data sets that are presented later, it was found that not all the features are required to define the behavior of all the malicious activities. Certain malicious activities can be defined using just two or three features, whereas others require all the features. The reason for this is that certain malicious activities such as Internet Control Message Protocol (ICMP) flood are independent of specific features such as port entropies that do not pertain to such malicious activities. Accordingly, certain features have values in the normal range in all instances of the same malicious activity in different training data sets because they do not aid in identifying such malicious activity.

The following set of tables summarizes the analysis of the behavior of all the malicious activities on the basis of the various training data sets. It should be noted that the feature that was considered less important to define the behavior of the malicious activity 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 IX.

With the recorded values, it can be concluded that the behavior of the system compromise malicious activity is Medium DP, Medium SP, High TB, and Medium PC. In this case, the DIP 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 DIP values.

Table X shows the values recorded for malicious file downloads in different training data sets. With these values, the behavior of malicious file download can be defined as Very High Total Packet Bytes and High Packet Count. Note that the entropy values are omitted from Table X 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 a malicious file download, there is no significant change in the entropy values because most of the communication occurs between two machines using specific ports, that is, FTP, HTTP, and so on.

Table XI shows the values recorded during the IRC communications that were noticed in the different training data sets. With these values, the behavior of the IRC

**Table IX.** Malicious activity type: system compromise.

| Training data set | Dst IP entropy | Dst port entropy | Source port entropy | Total payload bytes | Total packet count |
|---|---|---|---|---|---|
| Scan 28 | 0–2.2 | 2.02–2.988 | 2.02–3.11 | 4547–742 346 | 22–1491 |
| Scan 14 | 0–1.84 | 3.15–3.56 | 3.065–4.465 | 12 118–70 497 | 138–185 |
| Scan 19 | 0.9893 | 1.8078 | 2.159 | 1191–13 145 | 33–102 |
| SSH-based honeypot | 1.2223 | 2.0894 | 2.0773 | 343 184 | 385 |

**Table X.** Malicious activity type: malicious file download.

| Training data set | Dst IP entropy | Dst port entropy | Source port entropy | Total payload bytes | Total packet count |
|---|---|---|---|---|---|
| Scan 28 | X | X | X | 392 336–42 346 | 753–1491 |
| Scan 14 | X | X | X | 16 805–70 497 | 145–185 |
| Scan 19 | X | X | X | 374 099 | 4152 |
| SSH-based honeypot | X | X | X | 103 512–1 271 603 | 1727 |

**Table XI.** Malicious activity type: IRC communication.

| Training data set | Dst IP entropy | Dst port entropy | Source port entropy | Total payload bytes | Total packet count |
|---|---|---|---|---|---|
| Scan 28 Day 1 | 0–2.5 | Many 0 points<br>1–2.5 | Many 0 points<br>1–2.6 | 6200–19 048 | 10–97 |
| Scan 28 Day 3 | 0 | 0 | 0 | 1657–8652 | 15–75 |
| SSH-based honeypot | 1.58 | 1.79 | 1.78 | 26 263–10 660 | 229–249 |

communications can be defined as Zero DIP, Zero DP, Zero SP, Medium TB, and Medium PC.

Table XII shows the various values recorded during the ICMP flood anomaly. The values indicate this malicious activity behavior as High TB and Medium PC. The reason that this malicious activity 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 the layer 4 protocols.

Table XIII shows the values recorded during the port scan malicious activity. With these values, the behavior of port scan malicious activity can be defined as Very High DP, Very High SP, High TB, and Very High Packet Count. Because this malicious activity basically scans the ports on the target machine, it is independent from the DIP.

Table XIV shows the variation of different features during the network scan malicious activity. With the recorded values, the network scan behavior can be defined as Very High DIP, Very High SP, High TB, and Very High PC. We should note that the network scan involves the scanning of a large number of IP addresses, and therefore, it is independent of the DP.

Table XV shows the variation of the different parameters recorded during a bruteforce malicious activity. With these values, the behavior of a bruteforce malicious activity can be defined as Medium DP, High SP, Medium TB, and High PC. During bruteforce attempts, most of the communication occurs between two machines, and hence, it does not cause significant changes in the DIP.

**Table XII.** Malicious activity type: ICMP flood.

| Training data set | Dst IP entropy | Dst port entropy | Source port entropy | Total payload bytes | Total packet count |
|---|---|---|---|---|---|
| Scan 28 | 0.721–3.4 | 0–1.38 | 0–1.63 | 6348–16 177 | 6–58 |
| SSH-based honeypot | 1.584 | 0 | 0 | 14 372–56 636 | 14–55 |

**Table XIII.** Malicious activity type: port scan.

| Training data set | Dst IP entropy | Dst port entropy | Source port entropy | Total payload bytes | Total packet count |
|---|---|---|---|---|---|
| Scan 28 Day 1 | 0–0.91 | 7.09–8.685 | 6.95–9.81 | 153 112–764 302 | 406–3197 |
| Scan 28 Day 3 | 0–0.39 | 4.99–7.424 | 5.29–9.61 | 56 066–169 238 | 674–2773 |
| Scan 19 | 0.218–0.419 | 11.5–12.263 | 11.53–12.26 | 214 096 | 5374 |
| SSH-based honeypot | 2.00–4.91 | 4.51–5.877 | 3.66–5.94 | 15 289 | 154 |

**Table XIV.** Malicious activity type: network scan.

| Training data set | Dst IP entropy | Dst port entropy | Source port entropy | Total payload bytes | Total packet count |
|---|---|---|---|---|---|
| SSH-based honeypot | 16.87–17.14 | 0.037–0.67 | 10.97–14.3 | 117 906–10 677 114 | 1603–163 519 |

**Table XV.** Malicious activity type: bruteforce.

| Training data set | Dst IP entropy | Dst port entropy | Source port entropy | Total payload bytes | Total packet count |
|---|---|---|---|---|---|
| SSH-based honeypot | 0–1.4 | 0–4.39 | 3.98–6.53 | 29 680–76 402 | 494–1947 |

**Table XVI.** Classification of the behavior of different malicious activities.

| Malicious activity | Dst IP entropy | Dst port entropy | Src 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 |

VH, very high; H, high; M, medium; Z, zero.

As we can see from the previous analysis, we have used different number of instances to identify the different types of malicious activities. It can also be stated that the malicious activities' behaviors that were identified on the basis of more than one training data set have more significance compared with the behaviors identified on the basis of only one training data set. Hence, the presence of more instances of the same malicious activity in different traces will be useful in accurately predicting other similar types of malicious activities' behaviors.

Table XVI lists a classification of the various malicious activities and their associated behavior in terms of different features. Identifying the behavior of different malicious activities will help in detecting similar activities in other data sets. Using a large number of data sets will help in defining the behavior of the malicious activities better. This information can then be used to detect similar malicious activities by comparing the detected behavior with the proposed classification.

## 5. CONCLUSION AND FUTURE WORK

In this paper, we have employed an anomaly detection technique to classify different types of malicious activities present in Honeynet traffic. The classification of malicious activities is performed by using entropy distributions for feature-based parameters and traffic volume distributions for volume-based parameters. Using a number of collected Honeynet data sets, we have identified the parameters that provide sound detection capabilities of malicious activities in Honeynet traffic. As it has been demonstrated in the paper, the combination of DP, SP, DIP, TB, and TC provide a sound classification of the various anomalies. The behavior of various anomalies or malicious activities has been defined using the selected parameters and their respective threshold values, leading to a classification of various malicious activities.

As a future work, the proposed anomaly behavior classification can be used for identifying similar anomalies in other Honeynet data sets. In addition, the recommended behavior of various anomalies can be further tested for accuracy by using other Honeynet data sets. Testing the detection rate of the proposed anomaly detection technique will also be the focus of future work. In addition, a database of malicious activities' behaviors can be created on the basis of the proposed classification. Such a database can be used to accurately identify the type of malicious activity that occurred in a Honeynet data set.

## ACKNOWLEDGEMENTS

The authors would like to acknowledge the support provided by the King Abdul Aziz City for Science and Technology (KACST) through the Science and Technology Unit at King Fahd University of Petroleum and Minerals (KFUPM) for funding this work through project No. 08-INF101-4 as part of the National Science, Technology, and Innovation Plan. The authors would like to also thank all Saudi Honeynet Project team members for their feedback, especially Zubair Baig, Farag Azzedin, and Hakim Adiche.

## REFERENCES

1. Levine J, LaBella R, Owen H, Contis D, Culver B. The use of Honeynets to detect exploited systems across large enterprise networks. In the Proceedings of Information Assurance Workshop, IEEE Systems, Man and Cybernetics Society, 2003; pp. 92–99.
2. Spitzner L. 2003. Honeypots: tracking hackers. Addison-Wesley. Available at: http://www.tracking-hackers.com/book/
3. Spitzner L. May 2003. Honeypots, definitions and value of honeypots. Available at: http://www.spitzner.net/honeypots.html
4. Honeynet. 2005. *Know Your Enemy: GenII Honeynets*. Naperville, IL 60563 USA. Available at: http://old.honeynet.org/papers/gen2/index.html
5. Honeynet.org. Honeynet Project. 2004. Honeynet definitions, requirements, and standards documentation. Honeynet Project website. Available at: http://old.honeynet.org/alliance/requirements.html [Acessed: December 2010].
6. Information entropy. Available at: http://www.absolute-astronomy.com/topics/Information_entropy [Accessed: December 2010].
7. Balas E, Viecco C. Towards a third generation data capture architecture for honeynets. In the Proceedings of Information Assurance Workshop (IAW '05), the Sixth Annual IEEE SMC, 2005.
8. Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. *ACM Computer Survey* 2009; **41**:1–58.
9. Spathoulas GP, Katsikas SK. Reducing false positives in intrusion detection systems. *Computers & Security* 2010; **29**:35–44.
10. Gong DF. 2003. *Deciphering Detection Techniques: Part II Anomaly-Based Intrusion Detection*. Network Associates: Santa Clara, California.
11. Dainotti A, Pescape A, Ventre G. NIS04-1: wavelet-based detection of DoS attacks. In the Proceedings of Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE, 2006; pp. 1–6.
12. Haggerty J, Berry T, Shi Q, Merabti M. DiDDeM: a system for early detection of TCP SYN flood attacks. In the Proceedings of Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE, Vol.4, 2004; pp. 2037–2042.
13. Ping D, Abe S. Detecting DoS attacks using packet size distribution. In the Proceedings of Bio-Inspired Models of Network, Information and Computing Systems (Bionetics), 2007; pp. 93–96.

14. Barford P, Kline J, Plonka D, Ron A. A signal analysis of network traffic anomalies. In the Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment, Marseille, France, 2002.
15. Lakhina A, Crovella M, Diot C. Mining anomalies using traffic feature distributions. In the Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Philadelphia, Pennsylvania, USA, 2005.
16. Nychis G, Sekar V, Andersen DG, Kim H, Zhang H. An empirical evaluation of entropy-based traffic anomaly detection. In the Proceedings of the 8th ACM SIGCOMM conference on Internet measurement, Vouliagmeni, Greece, 2008.
17. Patcha A, Park J-M. An overview of anomaly detection techniques: existing solutions and latest technological trends. *Computer Networks* 2007; **51**:3448–3470.
18. Kind A, Stoecklin MP, Dimitropoulos X. Histogram-based traffic anomaly detection. *Network and Service Management, IEEE Transactions* 2009; **6**:110–121.
19. Thonnard O, Dacier M. A framework for attack patterns' discovery in Honeynet data. Presented at the Digital Investigation, 2008.
20. Al-Haidari F, Sqalli M, Salah K, Hamodi J. An entropy-based countermeasure against intelligent dos attacks targeting firewalls. Presented at the Proceedings of the 10th IEEE International Conference on Policies for Distributed Systems and Networks, London, United Kingdom, 2009.
21. François J, State R, Festor O. Activity monitoring for large Honeynets and network telescopes. *International Journal on Advances in Systems and Measurements* 2008; **1.1**:1–13.
22. Barford P, Chen Y, Goyal A, Li Z, Paxson V, Yegneswaran V. Employing Honeynets for network situational awareness. In *Cyber Situational Awareness*, Vol. **46**, Jajodia S, Liu P, Swarup V, Wang C (eds). Springer: US, 2010; 71–102.
23. Honeynet.org. Honeynet Project Challenges. Available at: http://www.honeynet.org/challenges [Accessed: December 2011].
24. hack.lu. Information security visualization contest, hack.lu 2009. Available at: http://2009.hack.lu/index.php/InfoVisContest, 2009.
25. Wireshark. Available at: http://www.wireshark.org/ [Accessed: December 2011].
26. Network Miner. Network Monitor. Available at: http://www.netresec.com/?page=NetworkMiner [Accessed: December 2011].
27. Lu W, Ghorbani AA. Network anomaly detection based on wavelet analysis. *EURASIP Journal on Advances in Signal Processing* 2009; **2009**:1–16.