IMPLEMENTATION OF EFFECTIVE DYNAMIC CLUSTERING ALGORITHM ON LIVE HONEYPOT DATA SET

Amit Chaudhary
Assistant Professor, CSE-IT DEPT
Modern Institute of Engineering and Technology, Mohri
Kurukshetra, Haryana, India

Heena Goyal
M.Tech. (CSE)
Modern Institute of Engineering and Technology, Mohri
Kurukshetra, Haryana, India

ABSTRACT

A Honeynet is a highly controlled network that interacts with attackers in cyber space to gather the attack data, to collect intelligence on attack techniques and behaviors of the black hat community. Other security devices like Firewalls, IDS etc are usually based on signatures and there has been large amount work done in the field of signature based detection. With the consideration of unknown attack detection, intrusion detection is becoming not useful to detect these kinds of attacks spreading in the network and intrusion detection based attack detection is becoming a very challenging process. Honeynets are enabling us in providing the deep understanding of attacks patterns which are bypassed by the network intrusion detection system. Therefore the Honeynets are becoming very useful to collect the unknown attacks. But before the deployment of Honeynets in the network, one should have a deeper understating of what honeynet can do and what are the risks involved in deployment. There should be a clear understanding of data control mechanism working properly to reduce the risk of compromising the honeynet system to other non-honeynet systems. In this research work, with the consideration of many problems in current traditional security resource applications
and the consideration of the research on Honeynet Technology, We have implemented of port Density Based Dynamic Clustering algorithm on attack data collected on Honeypots which infer the requirement of placing honeypots in any organizational network. A categorization of attack data such as portwise distribution, top IP addresses are being presented which is helpful for any system and network administrators to put control list in the network for blocking of those ports and IP addresses.

Keywords - Data Mining, Clustering, Dynamic Clustering, Outlier

INTRODUCTION

The main motive of a Honeynet is to collect information about threats that exist in a network; those attacks may not be detected by the network intrusion detection system. Based on research on Honeynet technology is a extremely controlled network which is used to analyze attackers in the wild. Honeynets are composed of an administration subnet and many hosts called honeypots. These are basically security resources designed to be probed, attacked or compromised. Such identification is popularly termed as a hacker, blackhat or cracker. The space of computers, connected to a network and Internet is increasing every day and applications running on these computers are growing from small to large which should be highly secured. Generally all the traditional network intrusion detection system (NIDS) is based on some predefined signatures to detect the known attacks in the network. Such systems prevailed as a part of the commercial in-use networks and also used techniques like pattern matching or anomaly detection. There were another type of security systems like system integrity checkers, likely to be host based. The major problem that these systems faced was that they used to run on computers, and went on with daily basis. These systems used to deal with large number of connections and data transfers resulting in huge log files which made it difficult to differentiate between normal traffic and intrusion attempts. There were many such systems which generated false positives or false negatives and gave very little insight to the tools and methods employed by the blackhat community.

In previous literature various methods were applied like Mohammed H. Sqalli et al. proposed a method to identify the best traffic features or parameters that can be used in an anomaly detection technique to identify anomalies in honeynet traffic and a provide detailed analysis of feature-based and volume-based parameters is carried out and the most appropriate features
for honeynet traffic are selected and evaluated feature-based and volume-based parameters that can be used for anomaly detection in honeynet traffic and candidate features were evaluated using real honeynet traces and also it was discovered that the combination of the destination port entropy, the source port entropy, the destination IP entropy, the total payload bytes, and the total packets provide better detection capabilities for various anomalies and said that these features can be used to design a technique to detect anomalies in honeynet traffic. Not only this later on Heng-ru et al. proposed an experimental network attack and defence platform based on virtual Honeynet and he can simulate the real network environment at a relatively low cost and analyze the valuable data that the Honeynet provides, he captured the intruder's intrusion traces effectively and analyze the intruder's tools, technologies and motives.

METHODOLOGY ADOPTED
The implemented work completes through the following steps in order to analyze the network attacks.

- Platform creation using virtualization and installation of Linux operating system as base operating system.
- Installation of Virtual Box tools to create multiple operating system on a single base machine.
- Installation of unpatched operating system with latest vulnerabilities.
- Configuration development to record the activities of invader.
- Network configuration and accessibility on base machine as well as virtual machines (depend no of honeypots).
- IP address setting and assignments.
- Network port configurations.
- Application bind with specific port (depending upon requirement).
- Network Attack capturing through honeypot and storage of attack data.
- Development of Data Processing Engine (DPE).
- Signature based classification through data processing engine.
- Packet analysis tools use for PCAP packet reading.
- Port wise clustering using port Density Based Dynamic Clustering algorithm.
MATHEMATICAL MODEL

In the proposed formulation, we have taken the parameter TTL also. Without TTL, we cannot get the full efficiency. You can check the formula in Excel sheet in which this parameter is used. Time-to-live (TTL) is a value in an Internet Protocol (IP) packet that tells a network router whether or not the packet has been in the network too long and should be discarded. For a number of reasons, packets may not get delivered to their destination in a reasonable length of time. For example, a combination of incorrect routing tables could cause a packet to loop endlessly.

Formula: IP packets carrying TCP segments:
\[ \text{payload} = \text{dgmlen} - \text{iplen} - \text{TTL} \]  
(SIR PLEASE EXPLAIN THIS SECTION IN OWN TERMINOLOGY WHICH HAVE USED IN OUR PROJECT).

In the proposed approach, the specific parameter of TTL (Time to Live) is taken as it is important in the evaluation of payload. The payload is directly associated with TTL. In existing approach, it is not considered.

MATHEMATICAL FORMULATION

At the very first level, there is a Data Items Repository (D). In the Data Repository, there will be number of transactions or data sets or transactional data. Each transaction or record or data set is termed as R(Ti). The Data Set regardless of the value and associated parameters will be eligible and move forward the fitness function modelling. FLD refers to the fitness level of the data items. Once the fitness level of the data items is applied successfully in the unbiased and unsupervised learning based measurement, it will generate the novel criteria for the addition in the dynamic clusters. At the final level, the set of clusters are generated with effective results and optimal parameters in terms of time.

\[ \sum (D_{ii}, \{TR(Ti)\} \rightarrow FLD) \Rightarrow CR \in R(TC (R): TP(R)) \Rightarrow FQm \ (m \leq i) \] where \( D_i = \) Data Items Repository

Terminology

TR(Ti)=Record / Data Set of the Transaction

FLD= Fitness Level of Data Items
CR=Cluster Eligibility of the Transaction Record R
FQm=Final Qualification for the Dynamic Cluster m where m <=i
TP = Percentage Equivalent of R
TC = Percentile Equivalent Value of R

And finally we want to achieve following

**OUTPUT: ( ∑ DCi ) ∞ (EXTi – EXTi-1)**

This section of the output generation and analysis is performing the calculations based on the difference between the times at two different instances. We have initialized the time at two levels. First: At the beginning of implementation. Second: At the final stage of implementation. At last the difference is measured to calculate the final execution time from start to end

Here, the Repository is complete back end database of honeypot.

Categorical Information Table – Complete Schema of the honeypot data

Third module is the algorithmic approach that finds out the similarity of IP Address with the specific port.

After this the Cluster Structure is identified and formation takes places

At the final level, the integrity of clusters is validated so that the record match with the back end table

\[ \sum (DII.\{TR(Ti)\}+FLD) => CR \ni R(TC (R):TP (R)) => FQm (m <=i) \text{ where } Di= \text{ Data Items Repository} \]

**OUTPUT: ( ∑ DCi ) ∞ (EXTi – EXTi-1)**

where DCi = Dynamic Cluster set

EXTi = Execution Time in ith level
• DL is the dynamic cluster set. Initially the cluster set is empty because the data items (ip address and hits) shall be added only after matching the fitness value. Fitness value means the closest percentile / association with the cluster base property. The cluster base property means: every cluster is broadly having the port with its specific value. The ip and hits shall be associated with the respective port and aggregation/grouping based on the matching is performed. EXT stands - execution time. it means we have calculated the execution time of cluster formation from EXTi (final time set measure at last) and EXTi-1 (initial time set) algorithm

1. Generation of the Dataset (Sequentially / Randomly) / Tuple series (fetched from large and big data based warehouse)

\[|DT| = \{ DTm | m \in (1,N) \}\]

Initially we have number of records in the honeypot. that is called the dataset DT

2. Association of the Qualification Value (QVi) to each data item that is based on the Acceptability / Avoidance of the Data Item for joining the Dynamic Cluster Formation Modules

\[DT = \{ DTi[QVi] | i \in (1,N) \}\]

After that we will find out the occurrence and association of each data item (ip, hits) with the cluster base property (port). DT (data items of honeypots) are mapped with QV (fitness value or qualification value)

3. Generation of the random clusters sets (if already exists)

\[DYC = \{ DYC i | i \in (1,N) \}\] and assign Threshold/Qualification Thrust

We will start generation of the clusters (all ports). Initially all ports are considered as independent clusters. The IP and hits (data items) will be mapped based on their association and classification

4. Compare DT with DYC based on Qualification value and Joined Parameters

We will compare every ip and hit value with the corresponding cluster (port) and join operation will be applied for association

5. If (DYCi == NULL) AND QTfitness==NULL GoTo Step 6
   Else
      If there is no cluster (port) for that ip or hit, go to end
If (DYCi = Initial Cluster) 
Assignment of the first thrust value based on the application 
If there is any port (cluster) for the ip and hit, it will join that cluster 
Else 
    GoTo Step 1 
Generate Final Results from both algorithms and calculate complexity 

This process will go till all the records are not completed. Finally we will find out the execution time 
6. End

EXPLANATION TO THE MATHEMATICAL FORMULATION
First of all describe in detail step by step procedure. Afterwards, this procedure is tested on database to check the validation and efficiency of the proposed technique 

- Data Repository Formation 
- Set of Data Records with multiple tuples and degree 
- Generation and Application of Cumulative Function 
- Application of Fitness Value to each record arbitrarily 
- Generation of the Clusters and Reading Records inside each Cluster 
- The Detailed Investigation of each record and cluster in terms of execution time and associated cost factors 

Existing Algorithm repeatedly reads tuples one by one from the dataset. When the first tuple arrives, a new cluster is formed. The consequent tuples are either put in the existing clusters or rejected by the existing clusters to form a new cluster based on the similarity measure between a tuple and a cluster. In existing approach each tuple belongs to one cluster only. 

where DCi = Dynamic Cluster set 
EXTi = Execution Time in ith level
IMPLEMENTATION SCENARIOS

Port Number: 23
IP Addresses Identified: 1
IP1: Hits (5)

Port Number: 1433
IP Addresses Identified: 4
IP2: Hits (2)
IP3: Hits (54)
IP99: Hits (2)
IP100: Hits (2)

Port Number: 3306
IP Addresses Identified: 1
IP5: Hits (2)

Port Number: 1998
IP Addresses Identified: 1
IP6: Hits (5)
GRAPHICAL REPRESENTATION OF EXECUTION TIME ON HONEYPOT RECORDS

It is clear in the graph that the execution time of the clustering algorithm implementation is very less and performing clustering in the efficient time. The clustering algorithm is implemented dynamically on the honeypot data and it is found that the results are effective in terms of cluster formation as well as turnaround time.
CONCLUSION AND FUTURE WORK
In this research work, we have presented the Honeynet based attack data collection and analysis of attack data based Intrusion Detection System placed inline in the organizational network. We have explored honeypot technology in depth which can be useful for tighten the security of the organization. We believe our solutions, if widely deployed, could significantly ease the sharing of collected data. We are having large amount of malicious PCAP data which is further useful for research perspective and can serve as a excellent environment for development of and automated IDS signatures. Our solution is completely automated but lack of automated correlation of attacker source IP address to Sebek Keystrokes remains a major problem. Our database schema is presently only for centralized botnets; no support for P2P botnets and encrypted botnets. We plan to add some basic support for these kinds of botnets also.

REFERENCES

