ATTACKS CLASSIFICATION IN NETWORK

1Shwetambari Ramesh Patil, 2Prof. S.B. Javheri

Department of Computer Engineering, Rajarshi Shahu College of Engineering,
University of Pune, Tathawade, Pune 411033. Maharashtra (India)

ABSTRACT

In Today's internet word the network security is main issue. The intruder can access our data easily if they are not detected. Here we are introducing the Modified Apriory Algorithm which detect attacks and classify them into its attack category and attack type. Modified Apriory Algorithm use the ICMP_TCP.txt file for generation of rules. These rules are given to snort and using snort in real time traffic we can detect and classify ICMP and TCP attacks. Classification of attacks helps to take proper action against attack if it is accurately classified.

Keywords: Artificial Neural Network, Intrusion Detection System, Multilayer Perceptron, Network Security

1. INTRODUCTION

We have designed the intrusion detection and classification system based on the Modified Apriory Algorithm. The system detect and classify attacks. This feature enables the system to suggest proper actions against possible attacks.

Internet is a global public network. With the growth of the Internet and its potential, there has been subsequent change in business model of organizations across the world. More and more people are getting connected to the Internet every day to take advantage of the new business model popularly known as e-Business. Internet work connectivity has therefore become very critical aspect of today’s e-Business. There are two sides of business on the Internet. On one side, the Internet brings in tremendous potential to business in terms of reaching the end users. At the same time it also brings in lot of risk to the business. There are both harmless and harmful users on the Internet. While an organization makes its information system available to harmless Internet users, at the same time the information is available to the malicious users as well. Malicious users or hackers can get access to organizations
internal systems in various reasons. These are: software bugs called vulnerabilities, lapse in administration, leaving systems to default configuration.

2. RELATED WORK

The timely and proper detection of computer and Network system intrusions has always been an important for system administrators and information security researchers. While the complexities of host computers already made intrusion detection a difficult attempt, the increasing in spreading distributed network-based systems and insecure networks such as the Internet has greatly increased the need for intrusion detection. Most current approaches to the process of detecting Intrusions utilize some form of rule-based analysis. Rule-based analysis sets of predefined rules that are provided by an administrator or created by the system. Expert systems are the most common form of rule-based intrusion detection approaches. The use of expert system techniques in intrusion Detection mechanisms were a significant milestone in the development of effective and practical detection-based information security systems. An expert system consists of a set of rules that encode the knowledge of a human “expert”. These rules are used by the system to make conclusions About the security-related data from the intrusion detection system. Unfortunately, expert systems require frequent updates to remain current. This design approach usually results in an inflexible detection system that is unable to detect an attack if the sequence of events is even slightly different from the predefined profile. While increasing the level of abstraction of the rule-base does provide a partial solution to this weakness, it also reduces the granularity of the intrusion detection device. The problem may arise that the intruder or hacker is an intelligent and flexible agent while the rule-based IDSs obey fixed rules. In fact expert systems suffer from the updating, the searching and the Matching of the rule sets.

An intrusion detection system (IDS) [3] is a device or software that observe the network or system activities for malicious activities or policy violations and produces reports to a Management Station. Some systems may attempt to stop an intrusion attempt but this is neither required nor expected of a monitoring system. Intrusion detection and prevention systems (IDPS)[4] are primarily focused on identifying possible incidents, logging information about them, and reporting attempts. In addition, organizations use IDPSes for other purposes, such as identifying problems with security policies, documenting existing threats and deterring individuals from violating security policies. IDPSes have become a necessary addition to the security infrastructure of nearly every organization.

2.1 Existing System
2.1.1 Anomaly based IDS

This is also known as Heuristic or Behaviour based, Anomaly based IDS analyzes the traffic patterns and determine normal activities. After that, it applies statistical or heuristic measures to event to determine if they match with this normal behaviour. Events which do not match with the accepted normal behaviour patterns are considered as attacks [1][7]. By creating patterns of normal behaviour, anomaly based IDS systems can observe when current behaviour deviates statistically from the normal. This capability theoretically gives anomaly-based IDSs abilities to detect new attacks that haven't been seen before or close variants to previously known attacks. It means, these types of IDS may identify any possible attacks [8].

Since normal behaviour can be changed over time, this type of system requires frequent retraining of the behaviour profile, lack of which results either in unavailability of the intrusion detection system or in additional false alarms. This type of systems normally
provides comparatively high false alarms as something even slightly different from normal behaviour is considered as an attack by this types of IDS. It also requires expertise to figure out what triggered an alarm.

2.1.2 Host based IDS (HIDS)

Host Based IDS is installed locally on host machines[1]. It works on information collected from within an individual computer system. It utilizes information sources like operating system audit trails, C2 audit logs and system logs. HIDS can be installed on different types of machines namely servers, workstations and notebook computers. Host-based IDS can analyze activities on the host it monitors at a high level of detail. It can often determine which processes and/or users are involved in malicious activities. It can monitor events that are local to a host and can detect successful or failure of attacks that cannot be seen by a network-based IDS. Host-based IDSs can use host-based encryption services to examine encrypted traffic, data, storage and activity. They are unaffected by switched networks and independent of network topology. They can provide thorough information gathered via logs and audit; for example Kernel logs records that the user is.

2.1.3 Network based IDS (NIDS)

Network based IDS monitors the traffic on its entire network segment[3]. This is generally accomplished by placing the network interface card in promiscuous mode to capture all network traffic segments. These network traffic packets are checked network by the IDS to find the attacks. Network based IDS can reassemble packets, look at headers, determine if there are any predefined patterns or signature match. Depending on this pattern or signature matching, IDS decides about the attacks. Network-based IDSs can monitor an entire, large network with only a few well-situated nodes or devices and impose little overhead on a network. This type of IDSs is mostly passive devices that monitor ongoing network activity without adding signature overhead or interfering with network operation. They can monitor network for malicious activity on known ports such as http port 80. They are easy to secure against attack and may even be undetectable to attackers; they also require little effort to install and use on existing networks. Furthermore, they are operating systems independents.

2.1.4 MLP Algorithm

A multilayer perception (MLP)[1][2] is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate output. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP utilizes a supervised learning technique called back propagation for training the network. MLP is a modification of the standard linear perceptron and can distinguish data that is not linearly separable. MLP algorithm is used to detect and classify attacks from KDD cup file. It classify attacks such as smurf, teardrop, etc.

3. PROBLEM STATEMENT

The timely and proper detection of computer and Network system intrusions has always been an important for system administrators and information security researchers. While the complexities of host computers already made intrusion detection a difficult attempt, the increasing in spreading distributed network-based systems and insecure networks
such as the Internet has greatly increased the need for intrusion detection. Most current approaches to the process of detecting intrusion detection mechanisms were a significant milestone in the development of effective and practical detection-based information security systems. An expert system consists of a set of rules that encode the knowledge of a human "expert". The problem may arise that the intruder or hacker is an intelligent and flexible agent. In fact expert systems suffer from the updating, the searching and the Matching of the rule sets.

The proposed system use Modified Apriory Algorithm to detect ICMP and TCP attacks and classify attack into attack category and attack type.

3.1 Goals and objectives
1. Protection against malicious tampering.
2. Interoperability with other network management and security tools.
3. To develop a solution with high detection rate and negligible false alarm.
4. We propose to use several different techniques of anomaly detection to achieve this goal.

3.2 Statement of scope
In this system we are implementing Modified Apriory algorithm and system works within LAN. This project can be useful to students, staff, office employees etc. Implementation of Modified Apriory Algorithm and attack classification is done. We are using ICMP_TCP.txt file in which signature of all TCP and ICMP attack is stored. This dataset is used as the input file for performing classification and generation of rule file.

- We use ICMP_TCP.txt file for calculating the output count in first phase.
- In second phase Modified Apriory Algorithm generate rules from TCP_ICMP.txt to detect and classify attack of TCP and ICMP.
- Modified Apriory algorithm and rules creation enables to perform the real-time operations. The server generates alarm, when attacks sent from the client are detected. The server then kills the process and shutdown client machine.

4. MODIFIED APRIORY ALGORITHM

By the observation of the attack signatures, we find that there are some attack signatures dependent on other previous attack signatures. This is due to the new attack is a derivative from the previous attack. So far as we know, there are at least two kinds of attacks have this property. For the first case, a new attack is variation of an existing attack. The steps of how to find out frequent k-item sets will be as follow. At the first step, all of the frequent items will be found. And then we use a simple way to scan the database in order to find the frequency of occurrence of each item, and decide which one meets the minimum support. Secondly, we generate the candidate n-item sets by checking all of the possible combinations of the frequent items with already known signatures, if they meet the minimum support requirement. Then, append this n item sets from right. We can first append the backward, until the minimum support is unsatisfied. Then, we append forward, and stop when the same condition occurred. Finally, the maximum length of frequent-item set can be mined by our method. When the minimum support decreases, the processing times of algorithms increase because of the total number of candidate item sets increases. Our algorithm is faster than the Signature Apriory no matter what the minimum support is. The reason is that the number of candidate 1-itemsets is not very large. Therefore, in the real environment, there are not too
much candidate item sets to be generated during each pass of finding signatures. One of the approaches of developing a network safety is to describe network behaviour structure that point out offensive use of the network and also look for the occurrence of those patterns. While such an approach may be accomplished of detecting different types of known intrusive actions, it would allow new or undocumented types of attacks to go invisible. As a result, this leads to a system which monitors and learns normal network behaviour and then detects deviations from the normal network behaviour.

Association Rules find all sets of items (itemsets) that have support greater than the minimum support and then using the large item sets to generate the desired rules that have confidence greater than the minimum confidence.

4.1 RELEVANT MATHEMATICS

- **Steps find_frequent_n-itemsets**
  Input: D, min_sup, L1, known_signature.
  Output: Ln
  Algorithm:
  Step 1: Ln = ∅;
  Step 2: for each item in L1 do
  Step 3: add known_signature + item into Cn
  Step 4: Ln = {c ∈ Cn | support (c) ≥ min_sup};

- **Algorithm candidates_gen**
  Input: D, min_sup, L1
  Output: Lk
  Algorithm:
  Step 1: Lk = ∅;
  Step 2: for each item in L1 do begin
  Step 3: add sig+ item into Ck
  Step 4: Lk = {c ∈ Ck | support (c) ≥ min_sup};
  Step 5: if there is no any s(sig+ item) ≥ min_sup then do step 6,7 until find maximum length of frequent itemset.
  Step 6: add item+ sig into Ck
  Step 7: Lk = {c ∈ Ck | support (c) ≥ min_sup};
  Step 8: if there is no any s(item+sig) ≥ min_sup. Then stop this procedure.

- **Proposed Algorithm**
  Input: D, min_sup.
  Output: Lk
  Algorithm:
  Step 1: L1 = find_frequent_1_itemsets(D, min_sup);
  Step 2: Ln = find_frequent_n_itemsets(D, L1, min_sup, known_signature);
  Step 3: for k=n+1 to max_len do
  Step 4: Ck = candidates_gen (D, L1, Lk-1, min_sup);
  Step 5: Answer = Lk
• **Steps find_frequent_n-itemsets**
  Input: D, min_sup, L1, known_signature.
  Output: Ln
  Algorithm:
  Step 1: Ln =\emptyset ;
  Step 2: for each item in L1 do
  Step 3: add known_signature + item into Cn
  Step 4: Ln =\{c \in Cn | \text{support (c)} \geq \text{min_sup}\};

• **Algorithm candidates_gen**
  Input: D, min_sup, L1
  Output: Lk
  Algorithm:
  Step 1: Lk = \emptyset ;
  Step 2: for each item in L1 do begin
  Step 3: add sig+ item into Ck
  Step 4: Lk =\{c \in Ck | \text{support (c)} \geq \text{min_sup}\};
  Step 5: if there is no any s(sig+ item) \geq \text{min_sup}
  then do step 6,7 until find maximum length of frequent itemset.
  Step 6: add item+ sig into Ck
  Step 7: Lk =\{c \in Ck | \text{support (c)} \geq \text{min_sup}\};
  Step 8: if there is no any s(item+sig)\geq\text{min_sup}. Then stop this procedure.

4.2 ARCHITECTURAL DESIGN

This represent the early stage of the software design process. Here a description of the program architecture is presented.

![System Diagram](image-url)

**Fig.4.1 Sytem Diagram**
Fig. 4.1 shows the architecture diagram. There are several modules that introduce the network intrusion detection system based on the Apriory algorithm. Now we are about to describe this layering structure step by step. The Modified Apriory Algorithm contains following steps:

- On server side Modified Apriory Algorithm is chosen. We find that there are some attack signatures dependent on other previous attack signatures. This is because of the new attack is a derivative from the previous attack.
- Modified Apriory Algorithm is used to classify attacks from ICMP_TCP.txt file.
- The input_count and output_count for ICMP_TCP.txt file is calculated. This gives the accuracy of attack classification.
- By using Modified Apriory Algorithm the Candidate set is generated. Then scan reduction technique is used.
- Using candidate set rules are generated in Modified Apriory algorithm. These rules are created using the combination of attacks which are stored in ICMP_TCP.txt file. These rules are stored in rules file.
- Rule file is given input to the snort here the real traffic is given to snort and using rule file it detect the TCP and ICMP attack and gives accuracy of detection and detection rate. It also detect the attack category and attack type.

4.3 DATA DESIGN

The system does not involve any use of database or entity tables. The data storage require is done in flat comma separated files.

4.3.1 Internal software data structure:
1. For modified Apriory algorithm we use ICMP and TCP packet and create rule file ICMP_TCP File in txt format

4.3.2 Database description

Modified Apriory Algorithm classify attacks of TCP and ICMP from TCP_ICMP.txt file. Modified Apriory Algorithm is also used for real time attack detection. Here the system detect the ICMP and TCP packet and attack using the rule file which is created using Modified Apriory Algorithm.

5. RESULT

In system we design Modified Apriory Algorithm. We compare the processing time of the proposed algorithm with or without the scan-reduction. It is clear that the processing time of the proposed algorithm with scan-reduction method rise sharply when minimum support is smaller than 0.4. The reason is that smaller minimum support causes very large candidates be generated after three passes. Therefore, we can know that the proposed algorithm with scan reduction method is good for the high minimum support.

Here, we use two different spans as for test. The result shows that by using longer string as signature for SNORT will get more accuracy than shorter string. Analysis show that the detection rate of the shorter span curve is higher than longer span curve. The reason is that the probability of a shorter signature in normal traffic (not attack) is high, and this will cause the high false positive rate.
Table 5.1 Classification rate-ICMP

Table 5.1 shows the Classification rate of ICMP attack type from ICMP_TCP.txt file. It uses the Modified Apriory algorithm for attack classification. ICMP_TCP.txt file contains all ICMP attack. The algorithm classifies the ICMP attack into shows the attack type and classification of attack type.

Table 5.2 Classification rate-TCP
Table 5.2 shows the classification rate for TCP attack. Table 5.2 shows the table of classification rate for TCP packet.

![Table 5.2 Classification rate for TCP attack](image1)

Table 5.3 Classification rate - real time

Table 11.3 shows the classification rate for real-time traffic. Table 11.3 shows the classification rate for real-time traffic containing TCP and ICMP attack.

![Table 11.3 Classification rate for real-time traffic](image2)

Fig 5.4 Detection rate - real time

Fig 11.1 shows the graph of detection of attack in real-time traffic using Modified Apriori Algorithm. The algorithm detects attacks using rule file which is given to snort.

![Fig 5.4 Detection rate - real time](image3)
6. FUTURE ENHANCEMENT

- Future work can be done by using the experimental setup and by using different number of nodes and with different traffic conditions.
- Real time implementation of the system for detecting more types of attack not only ICMP and TCP attacks

7. CONCLUSIONS

The project "Classification of Attacks In Intrusion Detection System" describes how we can classify TCP and ICMP attacks in real time traffic. The system use the Modified Apriory Algorithm which is a modified version of Signature Apriory Algorithm. Modified Apriory Algorithm generate rule file from ICMP_TCP.txt file. The rule file is given to snort for real time attack detection and classification. Here attack type and category is detected which is helpful for taking proper action against proper attack category and attack type.

8. REFERENCES


[6] James Cannady, Artificial Neural Networks for Misuse Detection, School of Computer and Information Sciences, Nova Southeastern University, Fort Lauderdale, FL 33314.


