IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

# Rule-Based Decision Tree to Identify Malicious Traffic

**Neha Jain**[*1] **, Dr Naveen Hemrajani**[2]

[*1,2] Suresh Gyan Vihar University, Jaipur, India

njlucky@gmail.com

## Abstract

Intrusion Detection Systems (IDSs) provide an important layer of security for computer systems and networks. An IDS's task is to detect suspicious or unacceptable system and network activity and to alert a systems administrator to this activity. Since data mining is one of the most emerging fields, when we talk about intrusion detection systems. In this paper, decision tree technique is applied on a small set of network data to find out normal and abnormal behavior. The algorithm generates a decision tree model which differentiates the malicious traffic from normal traffic and then generates rules according to that tree, and incorporates the model's logic into snort signatures or firewall rules.

**Keywords**: Data mining, IDS, malicious, intrusion.

## Introduction

An intrusion detection system (IDS) is a component of the computer and information security framework.

One of the main challenges in the security management of large-scale high-speed networks is the detection of suspicious anomalies in network traffic patterns due to Distributed Denial of Service (DDoS) attacks or worm propagation [1][2].The goal of IDS is to detect malicious traffic. In order to accomplish this, the IDS monitor all incoming and outgoing traffic.

When a potential attack is detected the IDS logs the information and sends an alert to the console. IDS try to find data packets that contain any known intrusion-related signatures or anomalies related to Internet protocols.

There are several approaches on the implementation of IDS. Among those, two are the most popular [11]:

***A. Anomaly detection*****:** This technique is based on the detection of traffic anomalies. The deviation of the monitored traffic from the normal profile is measured. On the basis of metrics used for measuring traffic profile deviation, various different implementations of this technique have been proposed.

***B. Misuse/Signature detection:*** This technique looks for patterns and signatures of already known attacks in the network traffic [12]. To store the signatures of known attacks, a constantly updated database is usually used. The way that anti-virus software operates, this technique also deals with intrusion detection on that way. [3][4][5][6][7].

Intrusion Detection Systems (IDS) have become a standard component in security infrastructures as they allow network administrators to detect policy violations.

## Problems with Current IDS

Following are the number of drawbacks of Current IDS:

1. To detect known service level network attacks, Current IDS are usually tuned. This leaves them vulnerable to original and novel malicious attacks.

**2.** ***Data overload:*** The next important aspect is how much data an analyst can proficiently analyze. That amount of data he needs to look at seems to be rising quickly. Depending on the intrusion detection tools employed by a company and its size there is the possibility for logs to reach millions of records per day.

**3.** ***False positives:*** A common complaint is the amount of false positives IDS will produce. A false positive occurs when normal attack is mistakenly classified as malicious and treated accordingly.

4. *False negatives:* This is the case where an IDS does not generate an alert when an intrusion is actually taking place. (Classification of malicious traffic as normal)

## Role of Decision Tree for IDS

The purpose of IDS is to help computer systems with how to discover attacks, and that IDS is collecting information from several different sources within the computer systems and networks and compares this information with preexisting patterns of discrimination as to whether there are attacks or weaknesses [10]. Decision Trees (DT) have also been used for intrusion detection [11]. Decision Tree is very powerful and popular machine learning algorithm for decision-making and classification problems. It has been used in many real life applications like radar signal classification, credit approval, medical diagnosis, weather prediction and fraud detection.

A decision tree is defined as "a predictive modeling technique from the fields of machine learning and statistics that builds a simple tree-like structure to model the underlying pattern".

Decision trees are one example of a classification algorithm. The *Classification* is a data mining technique that assigns objects to one of several predefined categories. From an intrusion detection the steps like 1.) perspective 2.) classification algorithms can characterize network data as malevolent, benign, scanning, or any other category of interest using information like source and destination ports, internet addresses, and total number of bytes sent during session.

Two prerequisites for the analysis are data collection (i.e. identifying and collecting data of interest) and tool acquisition and selection (i.e. identifying and deploying data mining techniques). The acquired data requires a pre-processing phase to move it into the form necessary for decision tree algorithms. Once the data is processed, decision trees can be trained using the processed data and tools. Executing and analyzing the result of this data is an important next step to understand the resulting model and its rule sets. The final step is using the results of the analysis to run the decision rules in real-time.

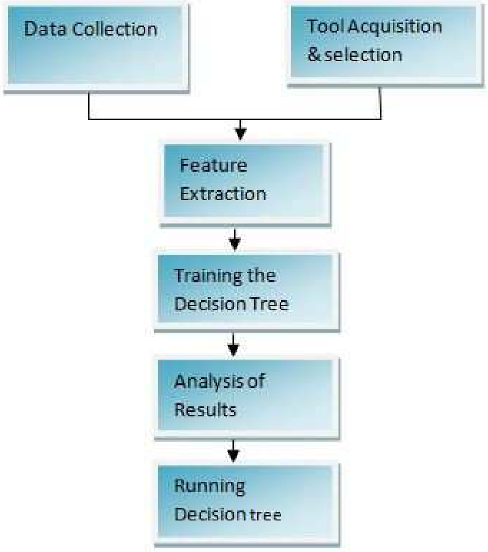


**Fig 1: Process to implement decision tree for Intrusion Detection**

## Implementation

A. *Procedure to classify normal traffic from malicious traffic:*

1. Implementing decision trees can require some network data. Download the following files from www.openpacket.org
   example.com-1.pcap
   example.com-3.pcap
   example.com-4.pcap
   example.com-5.pcap
   example.com-6.pcap
   example.com-7.pcap
   zeus-sample-3.pcap
   12b0c78f05f33fe25e08addc60bd9b7c.pcap
2. Implementing decision trees can require various tools:

- *Feature extraction tools*: used during the data pre-processing phase. For this we use tcptrace tool to perform feature extraction from pcap files.
  Download and install tcptrace from www.tcptrace.org
- *Data mining analysis tool*: Weka is used for this purpose. Of all the open-source tools, Weka has been described as "perhaps the best-known open-source machine learning and data mining environment"
  Download and install Weka from www.cs.waikato.ac.nz/ml/weka

3. Now the Feature Extraction tool is used to collect and structure the features from a dataset in a format that can be used for training the decision tree.

Steps:

i. For each pcap file, run the command:
   *tcptrace --csv -l filename1.pcap > filename1.csv*
   (here filename is the name of the pcap file)

ii. From each csv file, remove rows 1-8 (the row before conn #)

iii. From each csv file, delete the following columns EXCEPT
   - port_a
   - port_b
   - total_packets_a2b
   - total_packets_b2a
   - unique_bytes_sent_a2b
   - unique_bytes_sent_b2a

iv. Add new column called "class" to each spreadsheet. Fill in each cell of the new column with either "normal" or "malicious".

v. Copy and paste all cells from the spreadsheets into a single csv file called My_implementation

4. Run Weka

i. From the Weka GUI Chooser, click on the Explorer button

ii. From the Weka Explorer GUI, click on Open File.

iii. Using the explorer, open the My_implementation.csv file

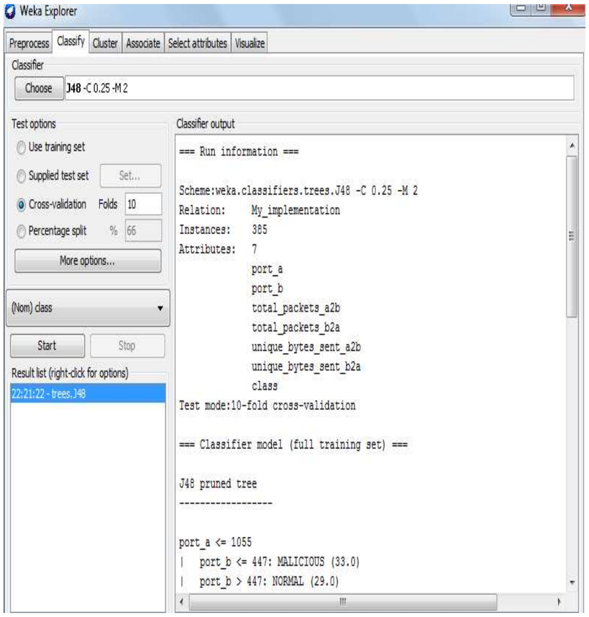
iv. Click on the Classify tab at the top of the Weka Explorer GUI
Click on the "choose" button to select a classifier
From the menu, expand the trees icon
Click on the J48 Tree classifier

v. On the Classify GUI, click on the Start button to start the classifier

5. Weka Output

The output of the decision tree in the above example states that connections with port_a <= 1055 and port_b <= 447 are malicious, otherwise the connection is normal.

*Tcptrace :* defines port_a as the port of the machine initiating the connection and port_b as the port of the machine receiving the connection.



Classifier: J48 -C 0.25 -M 2

Cross-validation Folds: 10

Percentage split %: 66

(Nom) class

22:21:22 - trees.J48

```
=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     My_implementation
Instances:    385
Attributes:   7
              port_a
              port_b
              total_packets_a2b
              total_packets_b2a
              unique_bytes_sent_a2b
              unique_bytes_sent_b2a
              class
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

port_a <= 1055
|   port_b <= 447: MALICIOUS (33.0)
|   port_b > 447: NORMAL (29.0)
```

**Fig 2: Weka output**

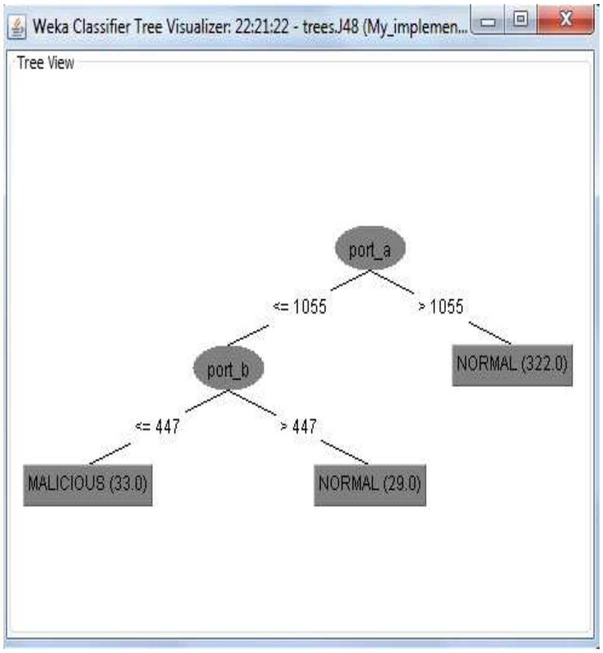


**Fig 3: Tree view**

B. *Rule-Based Classification Using Decision Tree:*

The decision tree classification algorithm detects activity based on data elements that are not captured by IDS like Snort in real-time, using features like the total packet counts sent from one host to another.

1. For implementing this, all the tools and data files remain same as of Part A.
2. The additional file "Tcp-scan.pcap" is required which is available at www.openpacket.org
3. Steps:

i. For the tcp-scan.pcap file, run the command: *tcptrace --csv -l filename1.pcap > filename1.csv*
(Here filename is the pcap file's name)

ii. Remove rows 1-8 (the row before conn #)

iii. For the file, delete the following columns EXCEPT

- port_a
- port_b
- total_packets_a2b
- total_packets_b2a
- unique_bytes_sent_a2b
- unique_bytes_sent_b2a

iv. Add a new column called "Class" to the resulting spreadsheet. Fill in each cell of the new column with "SCANNING".

v. Copy and paste all cells from the spreadsheets into a single csv file called"My_implementation2.csv"

4. Model Creation

i. Run Weka

ii. From the Weka GUI Chooser, click on the Explorer button

iii. From the Weka Explorer GUI, click on Open File My_implementation2.csv

iv. Using the explorer, open the My_implementation2.csv file

v. Click on the Classify tab at the top of the Weka Explorer GUI

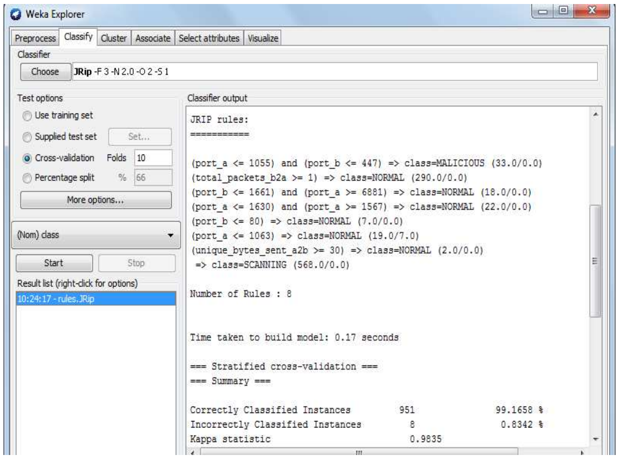
vi. Click on the “Choose” button to select a classifier
From the menu, expand the rules icon
Click on the JRIP classifier

vii. On the Classify GUI, click on the Start button to start the classifier

5. Weka Output

List of spruced version of the output from Weka from the rule-based classifier:

JRIP rules:
===========
(port_a <= 1055) and (port_b <= 447) => class=MALICIOUS (33.0/0.0)
(total_packets_b2a >= 1) => class =NORMAL (290.0/0.0)
(port_b <= 1661) and (port_a >=6881) => class=NORMAL (18.0/0.0)
(port_a <= 1630) and (port_a >= 1567) => class=NORMAL (22.0/0.0)
(port_b <= 80) => class= NORMAL (7.0/0.0)
(port_a <= 1063) => class= NORMAL (19.0/7.0)
(unique_bytes_sent_a2b >=30) => class= NORMAL (2.0/0.0) => class= SCANNING (568.0/0.0)
Number of Rules: 8



JRip -F 3 -N 2.0 -O 2 -S 1

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

10:24:17 - rules.JRip

```
JRIP rules:
===========


(port_a <= 1055) and (port_b <= 447) => class=MALICIOUS (33.0/0.0)
(total_packets_b2a >= 1) => class=NORMAL (290.0/0.0)
(port_b <= 1661) and (port_a >= 6881) => class=NORMAL (18.0/0.0)
(port_a <= 1630) and (port_a >= 1567) => class=NORMAL (22.0/0.0)
(port_b <= 80) => class=NORMAL (7.0/0.0)
(port_a <= 1063) => class=NORMAL (19.0/7.0)
(unique_bytes_sent_a2b >= 30) => class=NORMAL (2.0/0.0)
 => class=SCANNING (568.0/0.0)

Number of Rules : 8


Time taken to build model: 0.17 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         951               99.1658 %
Incorrectly Classified Instances         8                0.8342 %
Kappa statistic                          0.9835
```

**Fig 4: Weka output**

## Conclusions

This paper has presented a decision tree technique that categorize new piece of information into a number of predefined categories. Decision tree uses a pre-classified dataset to learn to categorize data based on existing trends and patterns. After the creation of tree, the decision tree’s logic can be incorporated into number of intrusion detection technologies including firewalls and IDS signatures.

Building an effective intrusion detection models with good accuracy and real-time performance are essential due to the increasing incidents of cyber attacks. Data mining is relatively new approach for intrusion detection. More data mining techniques should be investigated and their efficiency should be evaluated as intrusion detection models.

## References

[1] Christos Douligeris, Aikaterini Mitrokotsa, ”DDoS attacks and defense mechanisms: classification and state-of-the-art” ,Computer Networks: The International Journal of Computer and Telecommunications Networking,Vol. 44, Issue 5 , pp: 643 - 666, 2004.

[2] Z. Chen, L. Gao, K. Kwiat, Modeling the spread of active worms,Twenty- Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Vol. 3, pp. 1890 1900, 2003.

[3] Mithcell Rowton, Introduction to Network SecurityIntrusion Detection,December 2005.

[4] Biswanath Mukherjee, L.Todd Heberlein, Karl N.Levitt, ”Network Intrusion Detection”,IEEE, June 1994.

[5] Presentation on Intrusion Detection Systems, Arian Mavriqi.

[6] Intrusion Detection Methodologies Demystified, Enterasys Networks TM.

[7] Protocol Analysis VS Pattern matching in Network and Host IDS, 3rd Generation Intrusion Detection Technology from Network ICE

[8] Han, J. and Kamber, M. (2000). Data Mining: Concepts and Techniques, Morgan Kaufmann Publisher.

[9] Mannila, H., Smyth, P., and Hand, D. J. (2001). Principles of Data Mining. MIT Press. Mannila, H., Toivonen, H., and Verkamo, A. I. (1997)

[10] Berry, M. J. A. and Lino_,G. (1997). Data Mining Techniques. John Wiley and Sons, Inc

[11] Mithcell Rowton, Introduction to Network Security Intrusion Detection, December 2005.

[12] Mounji, A. (1997). Languages and Tools for Rule-Based Distributed Intrusion Detection. PhD thesis, Faculties Universitaires Notre-Dame dela Paix Namur (Belgium).