ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Identification and analysis of DoS attack Using Data Analysis tools

Niharika Sharma, Amit Mahajan, Vibhakar Mansotra

M.Tech Student, Dept. of Computer Science & IT, University of Jammu, Jammu, India

System Analyst, Dept. of Computer Science& IT , University of Jammu, Jammu, India

Professor, Dept. of Computer Science& IT , University of Jammu, Jammu, India

ABSTRACT: : Cyber threats have become more sophisticated and continue to increase in number day by day, thus making it difficult to detect and counter such attacks or intrusions into the network of interest. The framework designed in this paper can be used to capture data in the form of PCAP file to study and analyze the Dos SYN flood attacks using decision tree data mining tool and can be used to identify the threat severity level of the SYN flood attack in the network. With further enhancement this model can be used to prevent network attack (by blocking a particular SRC IP that is sending continuous SYN packets to a particular DST IP) etc.

KEYWORDS: DoS attacks, Decision trees, SYN Flood , Packet Capture, CSV File, WEKA

I. Introduction

Network attacks now-a-days have become more sophisticated and continue to increase in number day by day , making it difficult to detect and counter attacks or intrusions into a network of interest. DoS attacks are one such attack that has become one of the greatest threats to corporations as well as nation-states. According to reports of Kaspersky Lab in the fourth quarter of 2015, resources in almost 69 countries were targeted by Botnet assisted attacks. Also fourth quarter witnessed the longest Botnet based DDos attack which lasted for 371 hours i.e. 15.5 days approximately. Cyber criminals continue to use more sophisticated techniques to illegally gain access to systems and in response companies and employees are adopting new and more sophisticated networked technologies in the workplace to provide solutions to such problems. All of these factors make the task of defending a network more difficult and the trends indicate that these difficulties are likely to increase with time. Fighting against these challenges requires a wide range of tools and techniques to detect and defend such intrusions. Decision tree is one such technique that can assist in task (detection of intrusion). Decision trees can provide unique insights into the problem of identifying malicious activity and can assist in the construction of technology-specific techniques to defend and prevent against attacks.

II. Related work

In recent literature, many methods have been introduced to detect dos attacks. The majority of current detection projects depend upon feature selection from the ip packets captured. Jayveer Singh et al carried out a survey on Machine Learning Techniques for Intrusion Detection Systems. This paper present a rigorous survey study that envisages various soft-computing and machine learning techniques used to build autonomous IDSs. Intrusion detection techniques based on machine learning and soft computing techniques enable autonomous packet detections. These techniques are heavily based on statistical analysis of data. The ability of the algorithms that handle these data-sets can use patterns found in previous data to make decisions for the new evolving data-patterns in the network traffic.[1]

Vipin Das et al used Rough Set Theory (RST) and Support VectorMachine (SVM) to detect network intrusions. First, packets are captured from the network, RST is used to pre-process the data and reduce the dimensions. The features selected by RST is sent to SVM model to learn and test respectively. The method is effective to decrease the space density of data. The experiments compare the results with Principal Component Analysis (PCA) and show RST and SVM schema could reduce the false positive rate and increase the accuracy. The three main approaches we are considering is Paxson’s Bro, Leckie et al’s probabilistic approach and Jung et al’s sequential hypothesis testing for scan detection.[2]

DOI: 10.15680/IJIRCCE.2016. 0406208

Carl Livadas et al author use machine learning techniques to identify the command and control traffic of IRC-based botnets (compromised hosts that are collectively commanded using Internet Relay Chat (IRC)). The author split this task into two stages: (I) distinguishing between IRC and non-IRC traffic, and (II) distinguishing between botnet IRC traffic and real IRC traffic. For Stage I, He compare the performance of J48, naive Bayes, and Bayesian network classifiers, identify the features that achieve good overall classification accuracy, and determine the classification sensitivity to the training set size. A naive Bayes classifier performs best, achieving both low false negative (2.49%) and false positive (15.04%) rates for real-life IRC/non-IRC flows and low false negative (7.89%) rates for our botnet tesbed IRC flows. While some J48 and Bayesian network classifiers perform better for real-life IRC/non-IRC flows, they classified botnet testbed IRC flows poorly. For the feature sets and the traces considered, it was observed that training sets of 10K flows are sufficient and that the benefit of using larger sets is minimal.[3]

In this paper, the authors Samaneh Rastegari et al introduces an intrusion detection system for Denial of Service (DoS) attacks against Domain Name System (DNS). The system architecture consists of two most important parts: a statistical preprocessor and a neural network classifier. The preprocessor extracts required statistical features in a shorttime frame from traffic received by the target name server. The author compared three different neural networks for detecting and classifying different types of DoS attacks. The proposed system was evaluated in a simulated network and showed that the best performed neural network is a feed-forward back propagation with an accuracy of 99%.[4]

Research paper by Martin J Reed et al. presents an introduction to intrusion detection systems (IDS) and survey of different DoS/DDoS detection techniques. An overview and broad classification IDS are presented. The difficulties and characteristics of DoS/DDoS attacks are discussed in the DoS detection section. Furthermore, a classification of DoS attacks is explained. Three different classifications have been chosen and divided in two groups: general DoS classification and network flooding DoS-based. In each classification, many different proposed techniques are introduced and reviewed to point out the limitations. The key observation of this survey paper is that a CUSUM-based detection technique has many advantages over other statistical instruments in that it is nonparametric; consequently, it does not require training and is more robust to variations in the attack profile. [5]

Research paper by Prajakta Solankar et al shows various techniques for classification of attack. K-Nearest Neighbor (KNN), support vector machine (SVM), decision tree and naïve bayes are described and experimental results by using weka tools are determined. In this paper various denial of service attack types and review of various classification techniques like support vector machine, k-NN, naïve bayes and decision tree are given. From weka tool, the author analyzed that support vector machine and k-NN having more accuracy than all other however k-NN requires more time.[6]

In this paper, the authors Bayu Adhi Tama et al attempts to classify papers concerning DoS/DDoS attack detection using data mining techniques. 35 papers were selected and carefully reviewed by authors from two online journal databases. Each of selected paper was classified based on the function of data mining such as association, classification, clustering, and hybrid methods. The findings of this work indicate that classification and hybrid techniques received a great deal of attention from researchers. Our literature review provides a state of the art analysis concerning DoS/DDoS attack detection using data mining techniques.[7]

In this paper, the authors Majed Tabash et al proposes an approach merging methods from data mining to detect and prevent DoS attacks, by using multi classification techniques to achieve a sufficient level of accuracy and reduce false alert alarm. And secondly, the author evaluates his approach in comparison with other existing approaches. Author’s work is based on EGH Dataset to detect DoS attacks, in addition, he implemented his approach using Rapidminer, the experimental results show that the proposed approach is effective in identifying DoS attacks, the designed approach achieves significant results. In the best case, accuracy is up to 99.96%, author used two component of security; Snort tool and PfSense firewall, and compared his approach with other approaches, and also his approach achieves best accuracy results in most cases.[8]

Research paper by V. Hema et al presents a traffic classification scheme to improve classification performance when few training data are available is used. The traffic flows are described using the statistical features and traffic flow information is extracted. A traffic classification method is proposed to aggregate the naïve bayes predictions of the traffic flows. Since classification scheme is based on the posterior conditional probabilities, it can identify attacks occurring in an uncertain situation The experimental results show that the proposed scheme can efficiently classify packets than existing traffic classification methods and achieved 92.34% accuracy.[9]

III. UNDERSTANDING DoS ATTACKS

Denial-of-service attacks popularly known as ‘DoS’, is an attempt by attacker to deprive legitimate user of resources/services in a network. In other words, A Denial of Service (DoS) attack can be described as an attack designed to render a computer or network incapable of providing normal services [10]. TCP SYN Flood is one such type of attack. In this process, a synchronize (SYN) request is sent to the required target by the node requesting Connectivity, to begin the handshake process. The target responds to this request by sending an acknowledgement (SYN/ACK) package back to node initiating the connection. At last a connection is established when the node requesting connectivity sends an acknowledgement (ACK) back to the target to finish the handshake process. In a TCP SYN attack scenario an attacker (one or more) sends SYN connection requests to a specific target over and over again, without complete handshake. As a result due to pending connections the connection buffer of the target will be filled, and thus preventing it from answering new requests that may be valid.

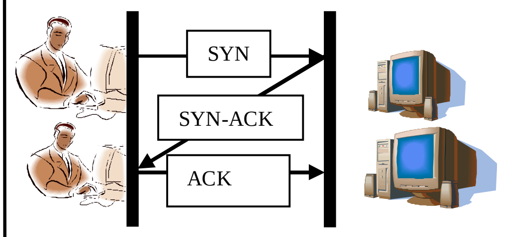


Fig: Normal Scenario

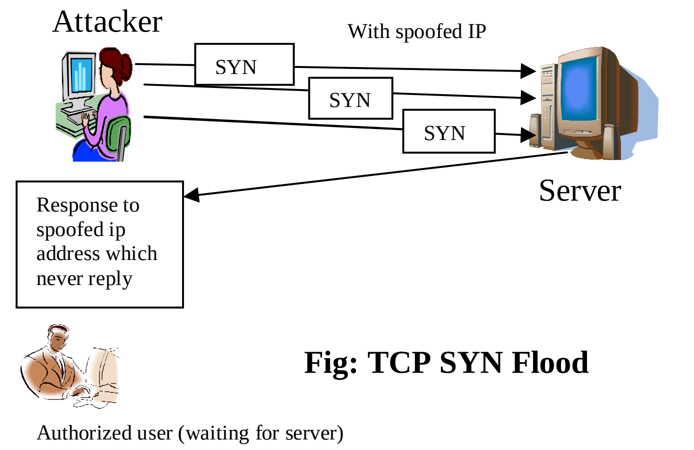

Attacker
With spoofed IP
SYN
SYN
SYN
Server
Response to spoofed ip address which never reply
Authorized user (waiting for server)

Fig: TCP SYN Flood

DECISION TREES

Decision Trees are one of the most widely use data mining tool for classification purposes. A decision tree is used as a classifier for determining an appropriate action (among a predetermined set of actions) for a given case [11]. They are a non-parametric supervised learning method used for classification and regression purposes. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

Structure of Decision Tree

A decision tree has three types of nodes:

- Root node: Root node is the top most node. It has no incoming edge but zero or more outgoing edge.
- Internal node: internal node has exactly one incoming edge and two or more outgoing edges.
- Leaf node: Leaf node has exactly one incoming node and no outgoing edge.

Decision trees play an important role in the process of intrusion detection. From an intrusion detection perspective, decision trees can classify incoming packet as malicious, normal or any other category using information like source port, destination port, no. of SYN flags from a particular source to destination port(in case of SYN Flood) etc. This can be illustrated with the help of an example, If the no. of SYN flags from same source to same destination are >20 then consider the packets as malicious else normal, as illustrated in the figure:

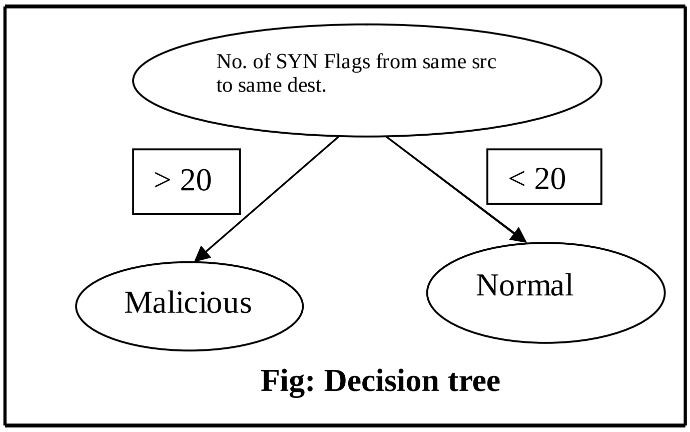


Fig: Decision tree

IV. PROPOSED ALGORITHM

In our proposed system, data is captured in the form of packets from the network. Our system uses a pcap file, a binary file accommodating network packet data, typically gathered via a network sniffer or a network packet inspection tool such as tcpdump. Since the packet capture file is binary file, here's where Wireshark tools come in very handy. Wireshark is a network packet analyzer. A network packet analyzer tries to display captured packet data in as much detail as possible. Wireshark package offers two main interfaces for processing pcap files: command-line and graphical; tshark and wireshark, respectively. Next step is preparing the data to be consumed by tool to be used for the data intake process. Tools often used supports a lot of different data and file types (referenced as data inputs), ranging from syslog to XML files. One of the simplest types is known as CSV, which stands for comma separated values. In order to turn a pcap file into csv, which is basically binary packed data, tshark command comes into play by allowing specific fields to be dumped from the network capture file. This process is also known as feature selection. Then the obtained csv file is used to perfoms classification as to whether a packet is normal or an attack.

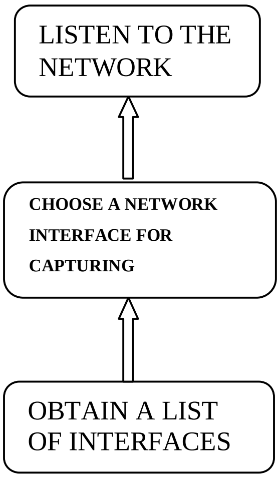

LISTEN TO THE NETWORK
CHOOSE A NETWORK INTERFACE FOR CAPTURING
OBTAIN A LIST OF INTERFACES

Fig: Packet capturing process

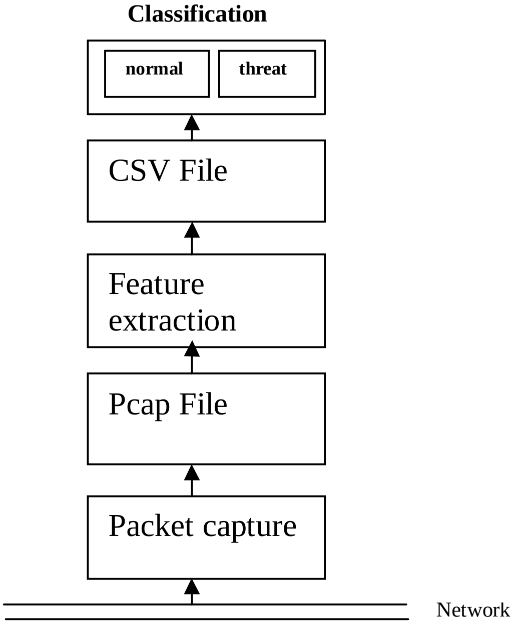


Fig: Proposed Architecture

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

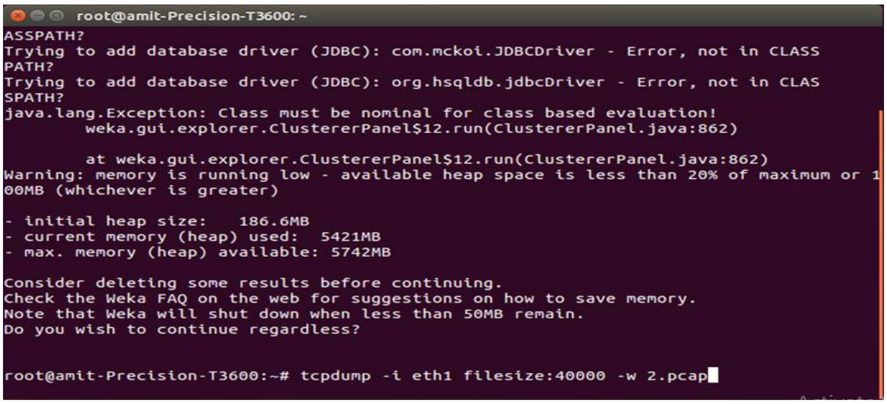
(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Packet Capture

Packet capture in computer networking is a term for intercepting a data packet that is moving over a specific computer network. A packet once is captured; it is stored temporarily to be analysed later on. The packet is examined to help diagnose and solve network problems and to ensure whether network security policies are being properly followed. Packet capturing techniques can be used by hackers to steal data that is being sent over a network.

We have used tcpdump as packet sniffer, the following screenshot provides the command to capture a ‘2.pcap’ file of size 4000 bytes.



```
root@amit-Precision-T3600: ~
ASSPATH?
Trying to add database driver (JDBC): com.mckoi.JDBCDriver - Error, not in CLASS
PATH?
Trying to add database driver (JDBC): org.hsqldb.jdbcDriver - Error, not in CLAS
SPATH?
java.lang.Exception: Class must be nominal for class based evaluation!
        weka.gui.explorer.ClustererPanel$12.run(ClustererPanel.java:862)

        at weka.gui.explorer.ClustererPanel$12.run(ClustererPanel.java:862)
Warning: memory is running low - available heap space is less than 20% of maximum or 1
00MB (whichever is greater)

- initial heap size:   186.6MB
- current memory (heap) used:  5421MB
- max. memory (heap) available: 5742MB

Consider deleting some results before continuing.
Check the Weka FAQ on the web for suggestions on how to save memory.
Note that Weka will shut down when less than 50MB remain.
Do you wish to continue regardless?


root@amit-Precision-T3600:~# tcpdump -i eth1 filesize:40000 -w 2.pcap
```

Fig: Command to capture packets of size 4000 bytes

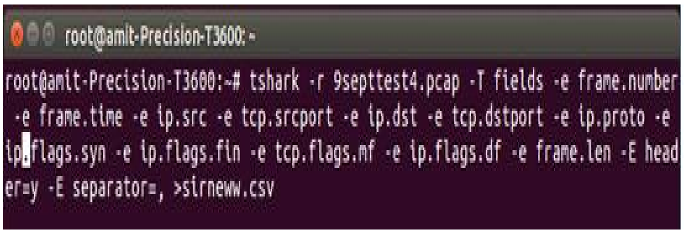
PCAP File

A pcap file is a binary file accommodating network packet data, typically gathered via a network sniffer or a network packet inspection tool such as tcpdump. From the above command we get a pcap file ‘2.pcap’ file of size 4000 bytes.

Feature Extraction

Feature extraction is a process where features such as tcp destination port, tcp sync flags, tcp source port, no. of ICMP etc. are drawn out from a file to aid the process of data analysing for eg: detection of anomalies, detect attacks etc.

We have used tshark to extract features from the captured pcap file using the command shown in the following screenshot.



```
root@amit-Precision-T3600: ~
root@amit-Precision-T3600:~# tshark -r 9septtest4.pcap -T fields -e frame.number
 -e frame.time -e ip.src -e tcp.srcport -e ip.dst -e tcp.dstport -e ip.proto -e
ip.flags.syn -e ip.flags.fin -e tcp.flags.mf -e ip.flags.df -e frame.len -E head
er=y -E separator=, >sirneww.csv
```

Fig: Command to extract feature set from pcap file

Feature set extracted

The above picture depicts the command for extracting using wireshark. Our feature set consists of 10 features which are :

Frame number	Tcp syn flags
IP Src	Tcp fin flags
TCP src port	IP mf flag
IP dst	IP df flag
IP dst port	class

Fig: Feature set extracted Using tshark

DOI: 10.15680/IJIRCCE.2016. 0406208

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

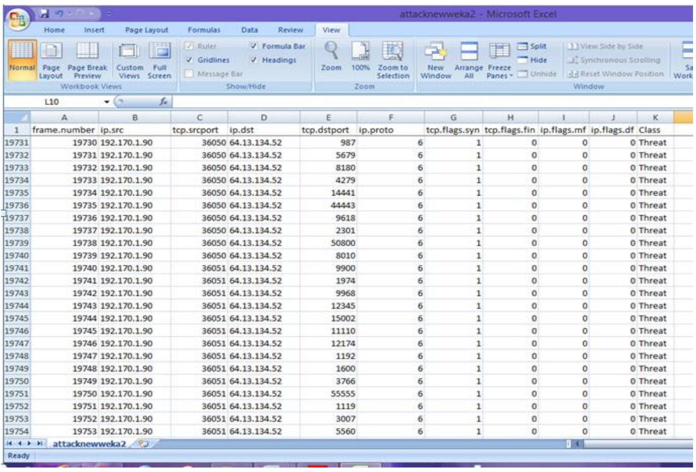
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

CSV File

Comma separated values store tabular data in plain text.. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. In order to turn a pcap file to CSV file, which is basically binary packed data, tshark command comes into play by allowing specific fields to be dumped from the network capture file. The following screenshot provides a glimpse of the CSV file used.



	A	B	C	D	E	F	G	H	I	J	K
1	frame.number	ip.src	tcp.srcport	ip.dst	tcp.dstport	ip.proto	tcp.flags.syn	tcp.flags.fin	ip.flags.mf	ip.flags.df	Class
19731	19730	192.170.1.90	36050	64.13.134.52	987	6	1	0	0	0	Threat
19732	19731	192.170.1.90	36050	64.13.134.52	5679	6	1	0	0	0	Threat
19733	19732	192.170.1.90	36050	64.13.134.52	8180	6	1	0	0	0	Threat
19734	19733	192.170.1.90	36050	64.13.134.52	4279	6	1	0	0	0	Threat
19735	19734	192.170.1.90	36050	64.13.134.52	14441	6	1	0	0	0	Threat
19736	19735	192.170.1.90	36050	64.13.134.52	44443	6	1	0	0	0	Threat
19737	19736	192.170.1.90	36050	64.13.134.52	9618	6	1	0	0	0	Threat
19738	19737	192.170.1.90	36050	64.13.134.52	2301	6	1	0	0	0	Threat
19739	19738	192.170.1.90	36050	64.13.134.52	50800	6	1	0	0	0	Threat
19740	19739	192.170.1.90	36050	64.13.134.52	8010	6	1	0	0	0	Threat
19741	19740	192.170.1.90	36051	64.13.134.52	9900	6	1	0	0	0	Threat
19742	19741	192.170.1.90	36051	64.13.134.52	1974	6	1	0	0	0	Threat
19743	19742	192.170.1.90	36051	64.13.134.52	9968	6	1	0	0	0	Threat
19744	19743	192.170.1.90	36051	64.13.134.52	12345	6	1	0	0	0	Threat
19745	19744	192.170.1.90	36051	64.13.134.52	15002	6	1	0	0	0	Threat
19746	19745	192.170.1.90	36051	64.13.134.52	11110	6	1	0	0	0	Threat
19747	19746	192.170.1.90	36051	64.13.134.52	12174	6	1	0	0	0	Threat
19748	19747	192.170.1.90	36051	64.13.134.52	1192	6	1	0	0	0	Threat
19749	19748	192.170.1.90	36051	64.13.134.52	1600	6	1	0	0	0	Threat
19750	19749	192.170.1.90	36051	64.13.134.52	3766	6	1	0	0	0	Threat
19751	19750	192.170.1.90	36051	64.13.134.52	55555	6	1	0	0	0	Threat
19752	19751	192.170.1.90	36051	64.13.134.52	1119	6	1	0	0	0	Threat
19753	19752	192.170.1.90	36051	64.13.134.52	3007	6	1	0	0	0	Threat
19754	19753	192.170.1.90	36051	64.13.134.52	5560	6	1	0	0	0	Threat

Fig: CSV file showing SYN Flooding

Classification

The classification process detects whether a packet is normal or is an attack. For the classification process we can use a data analyzing tool that may be licensed (SPSS modular) or open source(WEKA). We are using weka since it is one of the best open source Machine- Learning and data mining environment. We have used J48 tree for classification purposes . Our CSV file consist of 37160 instances and 11 attributes out of which 3 attributes namely : tcp syn flags, IP df flags and class are selected through attribute selection process.

WEKA Output:

```
=== Run information ===
Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: Weka1
Instances: 37160
Attributes: 3
            Tcp.flags.syn
            Ip.flags.df
            class
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===
J48 pruned tree
------------------
Tcp.flags.syn<=0: Normal(34370.51/942.51)
Tcp.flags.syn>0: Threat(2789.49/5.1)
Number of Leaves : 2
Size of tree: 3
Time taken to build model: 0.09 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances    36141    97.2578%
Incorrectly Classified Instances   1019     2.7422%
Kappa statistic                    0.8274
Mean absolute error                0.0331
Root mean squared error            0.123
Relative absolute error            27.4427%
Root relative squared error         50.4578%
Total Number of Instances           37160
=== Detailed Accuracy By Class ===
TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
 1         0.273     0.97        1        0.985       1          Normal
0.74        0         1           0.74     0.851       1            Threat
   0     0          0             0        0            0.493
Weighted Avg.
   0.973     0.246     0.972      0.973    0.97        0.999
=== Confusion Matrix ===
      a      b     c    <-- classified as
33428        0     0 |   a = Normal
   951   2713     0 |   b = Threat
    68       0     0 |   c = error
```

V. Analysis of Results of Weka

The weka output is broken into various parts like Run information, classifier model, Stratified cross-validation. The run information in general provides information about the model including the algorithm used, the examined data file, the number of instances in the CSV file, the features used in the classification process, and the testing method.

The classifier model in the WEKA output is the most relevant part for intrusion detection. For Decision tree algorithms this section shows a set of rules that determine whether or not SYN flooding exists. The decision tree in the output states that if no. of SYN packets from same source to same destination is greater than one than it is considered as threat otherwise it is normal i.e. Tcp.flags.syn <=0: Normal and Tcp.flags.syn >0: Threat. For reference, Tcp.flags.syn defines no. of SYN packets from same source to same destination.

The stratified cross-validation results are the last section of the Weka output. This section answers some question about the model like how well did this model perform? 10-fold cross-validation was used used to evaluate the accuracy of the model in the above experiment.

For the above data set, the summary shows that the model had 97.2578% accuracy in differentiating packets as normal or threat as well as some other error statistics. The detailed accuracy by class section presents a number of statistics for use in data mining; however these statistics can be difficult to interpret. The confusion matrix presents how the decision tree algorithm classified the data as compared to the actual category of the data.

VI. Conclusion

Decision tree analysis has the potential and can be used to support an intrusion detection team with the challenges of defending networks especially ‘campus networks’. Due to the huge volume of network data, decision trees can help in saving time for security experts and can also assist them in the analysis of malicious data in the campus network. Organizations can try implementing decision trees with existing network data. While performing this analysis, the decision tree algorithm learns the characteristics of the network and provides tailored feedback to support intrusion detection process, thus these algorithms can be implemented in the network to block the malicious traffic which affects the normal flow of campus network traffic.

References

1. Jayveer Singh, Manisha J. Nene ,” A Survey on Machine Learning Techniques forIntrusion Detection Systems,” International Journal of Advanced Research in Computer and communication Engineering Vol. 2, Issue 11, pp. 4349-4355, November 2013.
2. Vipin Das , Vijaya Pathak, Sattvik Sharma, Sreevathsan, MVVNS.Srikanth, Gireesh Kumar T,” NETWORK INTRUSION DETECTION SYSTEM BASED ON MACHINE LEARNING ALGORITHMS” International Journal of Computer Science & Information Technology (IJCSIT), Vol 2, No 6, pp 138-150, December 2010.

3. Carl Livadas, Bob Walsh, David Lapsley, Tim Strayer," Using Machine Learning Techniques to Identify Botnet Traffic," Internetwork Research Department BBN Technologies
4. Samaneh Rastegari, M. Iqbal Saripan and Mohd Fadlee A. Rasid," Detection of Denial of Service Attacks against Domain Name System Using Neural Networks," IJCSI International Journal of Computer Science Issues, Vol. 6, pp 23-27 No. 1, 2009.
5. Martin J Reed, Mohammed Alenezi," Methodologies for detecting DoS/DDoS attacks against network servers," The Seventh International Conference on Systems and Networks Communications, pp 92-98, 2012,.
6. Prajakta Solankar1, Prof. Subhash Pingale2, Prof. Ranjeetsingh Parihar3,"Denial of Service Attack and Classification Techniques for Attack Detection," (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (2) , pp 1096-1099, 2015.
7. Bayu Adhi Tama, Kyung-Hyune Rhee," Data Mining Techniques in DoS/DDoS Attack Detection: A Literature Review," The 3rd International Conference on Computer Applications and Information Processing Technology (CAIPT 2015), Yangon, Myanmar, June 23-24, 2015.
8. Majed Tabash, Tawfiq Barhoom," An Approach for Detecting and Preventing DoS Attacks in LAN," International Journal of Computer Trends and Technology (IJCTT) – Volume 18 Number 6, pp 265-27, Dec 2014.
9. V. Hema and C. Emilin Shyni," DoS Attack Detection Based on Naive Bayes Classifier, "Middle-East Journal of Scientific Research 23 (Sensing, Signal Processing and Security), pp 398-405, 2015.
10. Lee W and Stolfo S. Data Mining Approaches for Intrusion Detection. In Proceedings of the Seventh USENIX Security Symposium (SECURITY '98), San Antonio, TX. 1998.
11. Almuallim, Shigeo Kaneda, and Yasuhiro Akiba. ,Development and applications of decision trees. In Expert Systems The Technology of Knowledge Management and Decision Making for the 21st Century Six-Volume Set , pp. 53-77, Academic Press, 10 2001 .

DOI: 10.15680/IJIRCCE.2016. 0406208