International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 5 147 - 155

Support Vector Machine IDS Rule Extraction Mechanism from Honeypot Data

Saba Farooq [1], Kavita Patel[2]
Sharda School of Engineering and Technology
Sharda University Uttar Pradesh ,Greater Noida
E-mail: Sabaa.faruq@gmail.com[1], kavita.patel@sharda.ac.in[2]

Abstract- As awareness is increasing rapidly, more upto date aggressions are appearing. Security is a key to protection above all these problems. In this work, we will make a real existence scenario, employing honeypots. Honeypot is a well projected arrangement that entices hackers into it. By baiting the hacker into the arrangement, it is probable to monitor the procedures that are commenced and running on the arrangement by hacker. In supplementary words, honeypot is a mislead contraption that looks like a real arrangement in order to appeal the attacker. The target of the honeypot is analyzing, understanding, discerning and pursuing hacker's behaviors in order to craft extra safeguard systems. Honeypot is outstanding method to enhance web protection administrators' vision and discover how to become data from a victim arrangement employing forensic tools. Honeypot is additionally extremely functional for upcoming menaces to retain trail of new knowledge aggressions.

Keywords : *Honeypot, Security, Risks, Attacker, Firewall, SVM, SVM Classifier.*

*****

I. INTRODUCTION

Honeypot is a protection mechanism projected towards bait malicious interest to itself. Seizing such malicious interest permits for studying it to understand procedures and inspiration of attackers, and afterward helps to better maintain computers and webs [1]. A honeypot doesn't have each creation esteem. "It's a protection resource whose worth lies in being probed, assaulted, or compromised". Because Honeypot does not have each creation worth, each new hobbies or web traffic that originates from the honeypot shows that it has been prosperously traded off. Accordingly, a compromise is extremely effortless to see on honeypots. Fake positives as typically built upon established intrusion detection arrangements, don't continue on the honeypots.

Honeypot's beginning can be drew distant support to martial thoughts,custom. However, early materialized in the duration of protection of customers in the 1980s. In order to monitor the impostor on a live arrangement, Stoll and his associates endowed "bait", false martial information, to bait the intruder in a certain span of their system. Because this was not the honeypot that we understand nowadays, it was the early attempt of"catching flies alongside honey". The bulk of these arrangements were categorized according to the taxonomy's industrialized association scheme. The main class recognized of honeypots was the contact level. Probable benefits of the contact level are elevated and low. The elevated contact level denotes that the honeypot arrangement permits for maximum useful interaction. An example of such a honeypot is the Honeynet [2]. A low contact level signifies that the functionality is manipulated, for example by employing emulated services. Many finished they are complementary in nature and permit for extra accuracy, reliant on the conditions of placement and aims of data collection. For example, it could be unnecessary to use a elevated contact honeypot in on a globe scale as globe data is probable to be similar; low contact honeypots are extra suited for this situation.

II. RELATED WORK

Malicious activities present on the web make use of compromised web servers. Over a period of three months, our deployed honeypots , inspite their obscure location on a university network, attracted more than 44,000 attacker visits from close to 6,000 distinct IP addresses[1]. Flooding attack against Internet Threat Monitoring is addressed in which the attackers try to exhaust the network & ITM's resources such as computing power, network bandwidth or operating system data structures by sending the malicious traffic[2]. A no. of syatematic analysis modules are proposed & implemented in time scope which includes transient evidence recover, contamination graph generator, shellcode extractot & break-in reconstructor to facilitate honeypot forensics[3].

ITM is an efficient monitoring system globally used to detect, measure, characterize and track threats like denial of service and distributed Denial of Service attacks and worms. A novel traceback method is proposed for DDOS using Honeypots. IP tracing through honeypot is a single packet tracing method and is efficient than commonly used packet marking techniques[4]. Honeypots are traps that are

designed to resemble easy-to compromise computer systems to deceive postmasters. Postmasters might be able to detect these traps by performing a series of tests, depending on the complexity of services provided by honeypots. The problem of honeypot detection by postmasters is addressed. In particular, A Bayesian game theoretic framework is presented that models the interaction between postmasters and honeypots as a non-zero-sum no cooperative game with uncertainty. The game solution clarifies the optimal response available for the both players[5]. Low-interaction honeypots are able to provide a cost effective security mechanism for a wide range of computer systems. An agent-based optimization system is described that can automatize the generation of emulation programs for honeypots. The system is evaluated in its ability to emulate a server mail. In this evaluation, the system was able to produce correct responses to more than 99% of test data queries[6].

Drive-by-download attacks are the client-side attacks originated from web servers clients visit. High-interaction client honeypots find malicious web pages by directly visiting the web pages. However, they still have the shortcomings that must be addressed : possibility and long inspection time of not detecting certain attacks like time bombs. To address these kind of problems, a new detection method is proposed to identify web pages with time bombs. Experimental results illustrate that our method is more accurate and costs less than conventional methods[7]. Attacks like identity and call fraud theft more often involve sophisticated stateful attack patterns which on top of normal communication, try to harm the systems on a higher semantic level than usual attack scenarios. To this end we propose PRISMA , a method for state machine analysis and protocol inspection , which infers a message format and functional state machine of a protocol from network traffic alone. We demonstrate that PRISMA is capable of simulating correct and complete sessions based on the learned models[8].

Honeypots are very closely monitored decoys employed in a network to study the trail of hackers and to alert the network administrators of a possible intrusion. By analyzing the intrusion information, the content of the newest techniques of the intruder can be obtained and the system vulnerability can be found and the virtual honeypot can prevent the host computer from attacking[9]. Advanced Persistent Threats(APT's) gather data & information on the specific targets, using various kinds of attack techniques to examine the vulnerabilities of the target & then perform the data obtained by hacking. APT's are very intelligent & precise[10]. The botnet attacks are increasing each day & to detect such attacks has become challenging. Bots are having specific characteristics compared to normal malware as they are controlled by the remote master server and usually do not show their behavior like normal malware until they do not receive any command from their master server. Most of time bot malware are inactive, hence it is very difficult to detect them. The experience of Botnet detection in the private network is shared in the private network as well as on the public zone by deploying nepenthes honeypots[11].

During the last few years, Industrial Control Systems have evolved from proprietary systems to open architectures and standard technologies, highly interconnected with other corporate networks and even the Internet. ICS are adopting ICT solutions to promote corporate connectivity and remote access capabilities, and are implemented and designed using industry standard computers, operating systems and network protocols. While this integration introduced new ICT capabilities and tremendous cost optimization opportunities, it also provided less isolation for the ICS, from the outside world[12].

A security gap is always there between the actual level of security needed and ability to secure our networks. A skilled hacker will always find a way. securing networks need good intelligence to direct our efforts and focus on the right spots. Honeypots propose a wide range of possibilities and can also be designed to suit specific needs depending on the intel you want to collect. Cyber intelligence has become more and more important for analysing, tracking and countering of digital security threats within modern society. Situation awareness is important for being able to understand,discover and provide an early warning of new threats which help to prepare to meet a new threat e.g. viruses, hackers and terrorists. Honeypots have proven to offer timely, accurate and concise information for the situational awareness[13].

An investigation of the activity detected on three honeypots that utilize the Kippur SSH honeypot system on VPS servers all on the same C class address. The systems is able to run on identical software bases and hardware configurations. The initial analysis covered in the paper examines patterns nad behaviours detected of the attacking entities[14]. Honeypots which are trap designed to resemble the computers systems that are easy to compromise has become an important tool for security professionals and researchers because of their contribution in disclosing the underworld of cybercirmes. However, several anti-honeypot technologies hav been developed n the recent years. In particular, the

interaction between botmasters and honeypots by a Markov Decision Process (MDP) is modelled and then the honeypots optimal policy for responding to the commands of botmasters is determined. The model is extended using a Partially Observable Markov Decision Process (POMDP) which allows operators of honeypots to model the uncertainty of the honeypot state as determined by the botmasters . Simulation results that show the honeypots optimal response strategies and their expected rewards under different attack scenarios is provided[15].

III. PROPOSED METHODOLOGY

The goal is to find the research honeypot attack analysis of unknown species and not known type of attack. Raw data from the previously deployed honeypots is collected and statistical features are extracted from them. Honeypot alerts are also obtained that were recorded previously by snort, various such data is available online. Then Support Vector Machine SVM is applied to two data sources, Honeypot Data and Honeypot alerts and two honeypot rule extraction models will be obtained consequently.

Support Vector Machine (SVM) is an algorithmic technique that is used for pattern classification that has grown rapidly in recent times and is used in many field including bioinformatics. Support Vector Machine is an alluring method because of its high generalization capability and skill to handle high dimensional input data. . In contrast to neural networks or **decision trees** (previous works for honeypot detection), SVM does'nt suffer from local minima problem, it has some learning parameters to choose, and it produces stable and reproducible results. If perhaps two SVMs are being trained on the same data with same learning parameters, they produce the same results independent of the optimization algorithm they use.

Yet, SVMs suffer from sluggish training mostly with large input data size and linear kernels. SVMs are binary classifiers primarily. Plug-ins to multi-class problems are mostly made by combining various binary machines in order to produce final multi- classification results.

With the rapid proliferation and development of the web infrastructure and local networks, more security threats, e.g., ddos , computer viruses, Internet worms, sywares, adwares, trojan horses and bots, for computer systems and networks are also constantly emerging. Various efforts have been considered inorder to fight against these security threats in the last decade, which includes cryptography, firewalls, honeypot rule extraction systems, and so on. Honeypot Rule extraction is becoming increasingly significant to keep up high-level network security. fig. below describes whwre to place honeypots in the network.

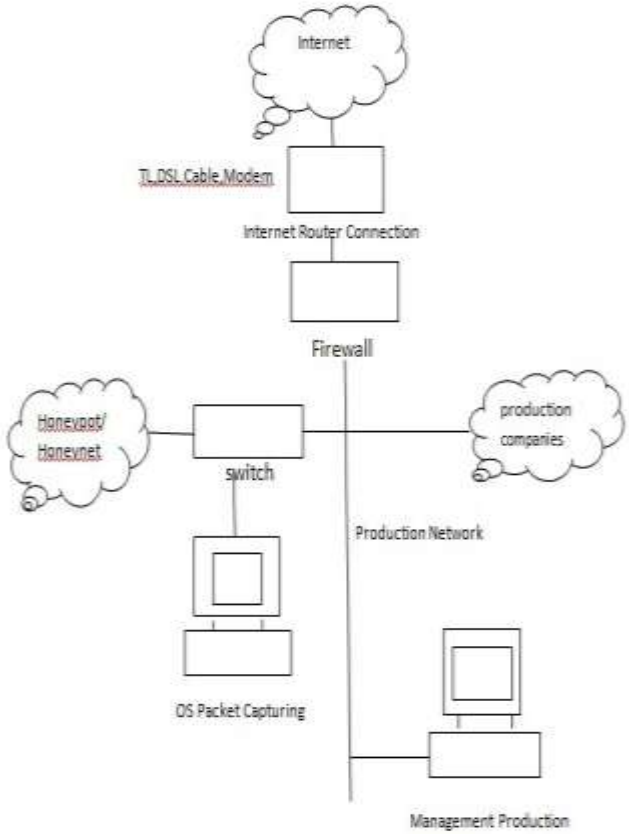

Fig 3.1: Honeypot Placement In a network

Honeypot Data

In the honeypot extraction field , KDD Cup1999 data would be taken as benchmark data to calculate performance of Honeypot Rule Extraction procedure using SVM. But KDD Cup1999 data set has a fatal drawback that it is not able to reflect most recent attack trends & current network situations, as it was made by simulation. Threfore, its attack types are old-fashioned. But researchers have used it as their analysis data inspite of this drawback because it actually is very difficult to obtain high-quality analysis data because of privacy and competitive issues. Most organizations hardly share their data with other researchers and institutions.

To provide more practical and useful analysis results, it is actually required to carry out the experiments by using real traffic data. Several types of honeypots would be deployed over different systems which are outside and inside. For eg., **Wireshark**, and other traffic record Devices .

All traffic data has been collected from/to honeypots and it has been observed that almost all of them consists of attack data. In fact, for the collected traffic data, a deep inspection has been carried out for each and every connection if there was a buffer overflow assault or not. Inorder to to identify an exploit code and shellcode from traffic data, dedicated detection software was used. Honeypot alerts extracted from Snortand malware information extracted from Online resources as some extra information for checking traffic data. By using these varieties of diverse information, we inspected the collected traffic data, and discovered what has occured on the networks.

Despite of inspecting real attacks on the campus networks, there is a certainity that unidentified attacks are being contained in the honeypot traffic data. However, in the analysis, It was observed that almost all of honeypot traffic data captured in honeypots are composed of attack data and there were some unidentified traffic data. Therefore, all the original honeypot traffic data are considered as attack data in the benchmark data, because the performance of one-class Support Vector Machine is almost unaffected by a tiny amount of unidentified attack data or they can be treated as noisy data.

However, As almost all of the honeypot traffic data contains attack data, large amount of normal data should be prepared in order to judge the performance of SVM effectively. In order to generate normal traffic data, a mail server was deployed on the about the same network with honeypots, and regarded the traffic data as normal data. Your ailbox server was operated with several communication protocols as well, which include, ssh, http and https, for their management and also received several attacks. Although all of these activities were present with the traffic data, they do not affect the performance of machine learning techniques present in our experiments because of their small amount.

Features Extraction

Extraction of only vital and essential features from traffic data using honeypots and Wireshark and a mail server, and continuous features excluding one categorical feature for the evaluation data.

Duration: the length of connection

Service : the service type of the connection, e.g., http, telnet

Source bytes : Its the data bytes sent by the source IP address

Destination bytes : Its the number of the data bytes that are sent by destination IP address

Count: the no. of connections whose source IP address and destination IP address are same to that of the present connection in the past two seconds.

Same srv rate : % of the connections to same services in Count feature

Serror rate : % of the connections that have got the "SYN" errors that are in the Count feature

Srv serror rate : % of the connections which have the "SYN" errors in Srv_count feature.

Dst host count : among the past connections whose destination IP address is same to that of present connection, the number of connections which has source IP address is the same to that of present connection.

Dst host srv count: Among the past connections whose destination IP address is the same to that of the present connection, the number of connections whose service type is also the same to that of the present connection

Dst host same src port rate : % of the connections whose source port is the same to that of the present connection in Dst host count feature

Dst host serror rate : % of connections that have the "SYN" errors in Dst host count feature

Dst host srv serror rate : % of connections which the "SYN" errors in Dst host srv count feature

Flag: state of connection at the time when the summary was written . The different types of states are summarized in the section given below.

IV. RESULTS & ANALYSIS

Evaluation process

Figure below shows the overall process of Honeypot Rule Extraction using SVM's and evaluation data: Honeypot data and Honeypot alerts. The evaluation process is comprised of two phases: Training phase and esting phase. The training phase is summarized as follows.

 ISSN: 2321-8169
147 - 155

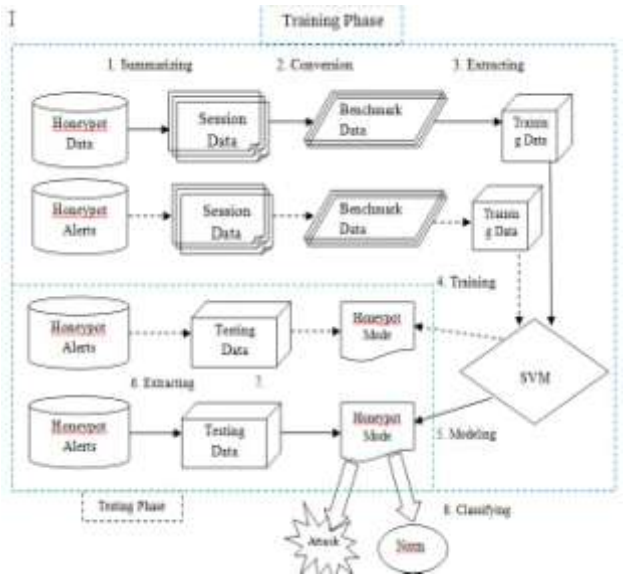

Fig 3.2: Architecture

While performing Honeypot learning we need to perform the subsequent steps:

Training phase: to present the honeypot data and train with SVM model, by pairing the input by all of the expected output.

Validation/Test phase: to calculate approximately how well the model has been trained (that is dependent upon the volume of the data, the value we need to predict, input etc) and to estimate model properties (mean error for numeric predictors, classification errors for classifiers, recall and precision for IR-models etc.)

Execution Steps

Capturing Packets

Select an interface within the interface list to start out packet capturing on the same interface. For example., Click wireless interface if the traffic is captured on the wireless network. By clicking capture choices advanced options are often confugured i.e., promiscous mode. Once the packet starts appearing within the real time Wireshark captures each packet sent to from system. If you have got promiscous mode & capturing is done on a wireless interface enabled in capture choices, alternative packets may also be seen on the network.

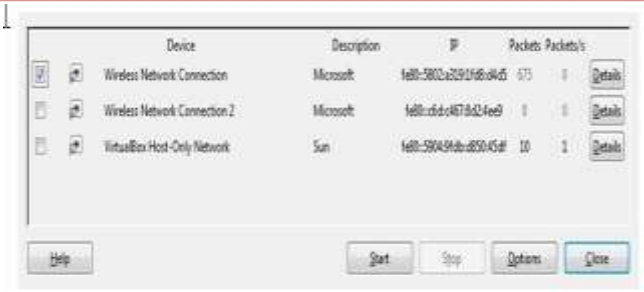

Fig 4.1: Selecting Packet Interface in wireshark

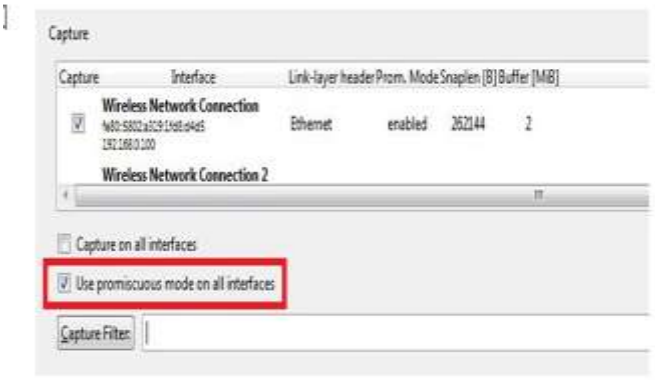

Fig 4.2: Promiscuous Mode in Wireshark

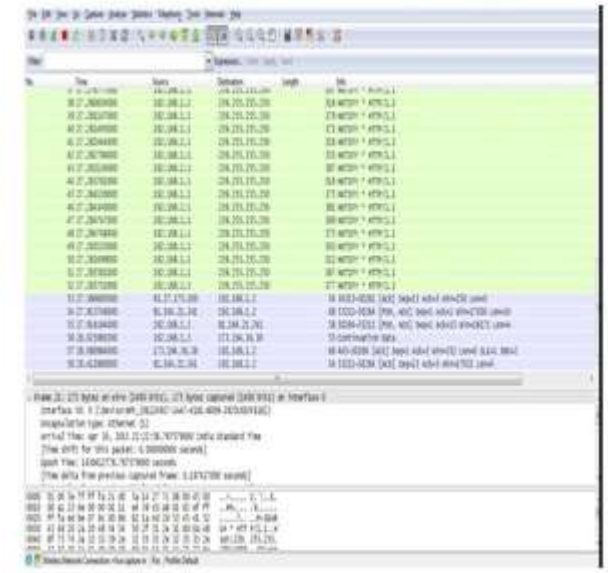

Fig 4.3: Wireshark For Capturing Network

The packets that can be seen painted in the green, blue, and black can be seen. Wireshark uses different colors to recognize the types of traffic at just one glimpse. By default, TCP is green traffic, DNS traffic is dark blue, UDP traffic is light blue, and the TCP packets with some problems are

 ISSN: 2321-8169
147 - 155

idnetified by black, for eg., they can be delivered out-of-order

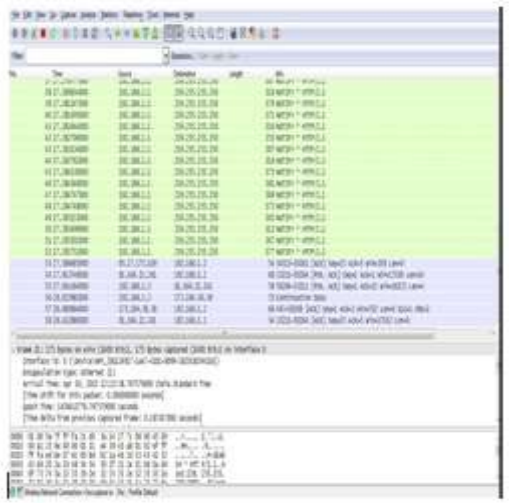

Fig 4.4: Traffic Patterns Detected Via Wireshark

Preprocessing

After capturing packets, the packets are exported to comma separated values(CSV). This CSV file is imported by MATLAB for cluster analysis. The import is done by csvread function of MATLAB with the help of regular expressions. csvread fills unfilled delimited fields with the zero. At the point when the csvread function reads record files by lines that end with a nonspace delimiter, for eg., a semicolon, it gives back a matrix, M, that has an extra very last column of the zeros. csvread imports any difficult number as a sum total into a compound numeric field, and converts the real and the imaginary parts into the specific numeric type.

Parsing Packets

Textscan is also used for importing packets. textscan attempts to match the data in the file to formatSpec, which is conversion specifiers of string. formatSpec is reapplied by textscan throughout the entire file and stops when it cannot match formatSpec to the data. The wireshark gives the packet in following columns, so format spec is chosen accordingly**.**

ID	Packet Identity
TIME	Time of Arrival
Source	Source IP Address of Packet IPv4 and IPv6
Destination	Destination IP Address of Packet IPv4 and IPv6
Length	Length of Packet
Info	Packet Information

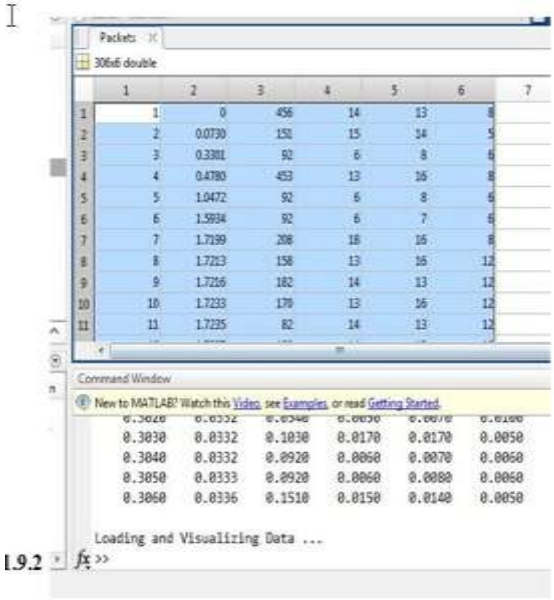

Fig 4.5: Captured and Preprocessed Packets in MATLAB

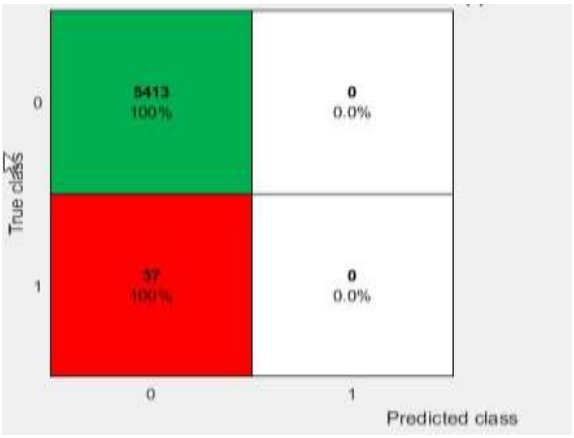

Fig 4.7: Best Data Subset performance Shown using Confusion Matrix

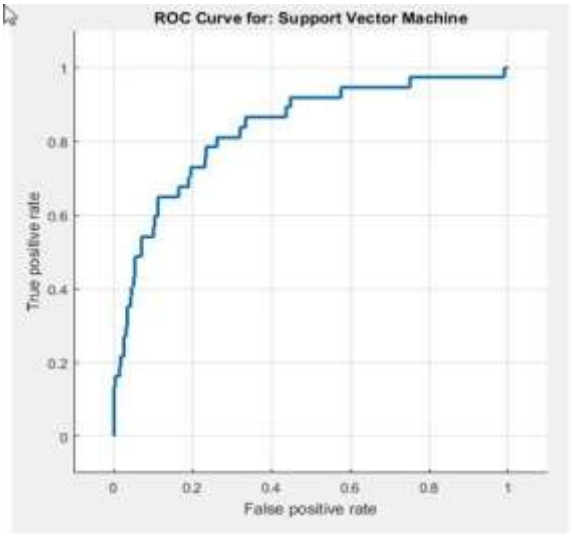

Fig 4.6: ROC Curve of the SVM Classifier after training, Showing more than 90% of area under the curve (AUC)

 ISSN: 2321-8169
147 - 155

Overall Accuracy of Decision Tree

Each of the decision trees that are generated for using a k fold cross validation method for all of the eight data subsets was evaluated, by k = 10. Percentage accuracy was calculated for each subset, together with the (FPR) and the (FNR). The values in Table below are the average of k-fold for the test set.

Table 4.1: Accuracy, FPR, FNR rates for Decision tree Classifier for various sets of data

Subset	1	2	3	4	5	6	7	8
Accuracy (%)	77.35	76.72	45.42	77.73	77.75	77.56	78.15	75.31
False Positive Rate (FPR)	0.0281	0.0385	0.0974	0.0318	0.0317	0.0320	0.0312	0.0472
False Negative Rate (FNR)	0.3750	0.1428	0.7142	0.1428	0.1428	0.1428	0.1428	0.1666

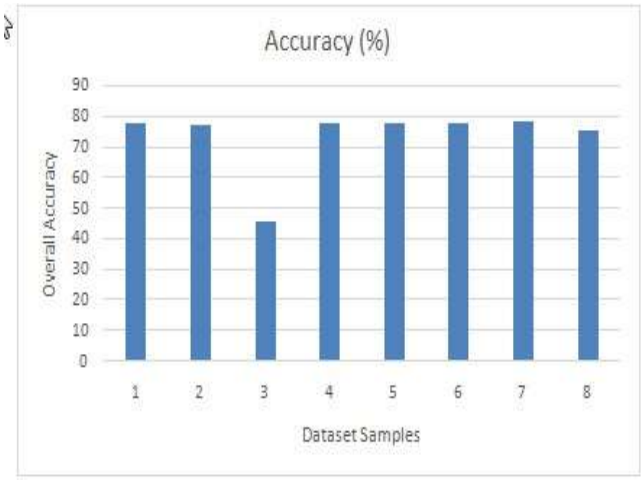

Fig 4.8 : FPR and FNR for Decision tree Classifier for 8 K-Folds

Accuracy of SVM Classifier for Detecting Honeypots

Each of the Support vector generated for using a k fold cross validation method for all of the eight data subsets was evaluated, by k = 10. Percentage accuracy was calculated for each subset, together with the (FPR) and the (FNR). The values in Table below are the average of k-fold for the test set.

Subset	1	2	3	4	5	6	7	8
Accuracy (%)	95.0391	96.3295	91.9542	96.3416	93.0104	91.6528	91.4818	93.3841
FPR	0.0253	0.0347	0.0877	0.0286	0.0285	0.0288	0.0281	0.0425
FNR	0.1375	0.1285	0.1428	0.1285	0.1285	0.1285	0.1285	0.1499

Table 4.2: Accuracy, FPR, FNR rates for SVM Classifier for various sets of data

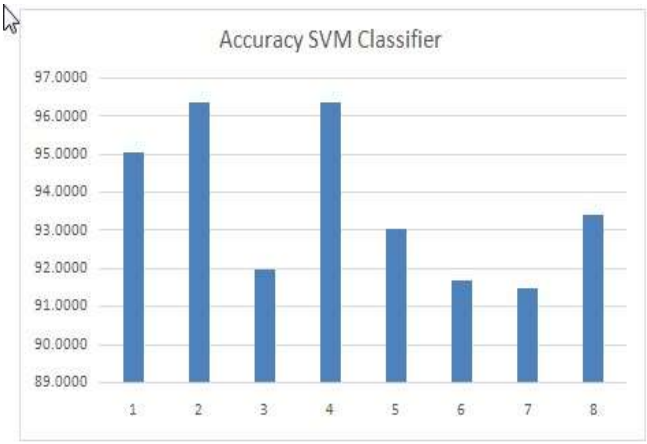

Fig 4.9 : Accuracy rates (%) for SVM Classifier, minimum accuracy of SVM is for the 7th subset and maximum is for 2nd subset of data with average accuracy about 93.5%

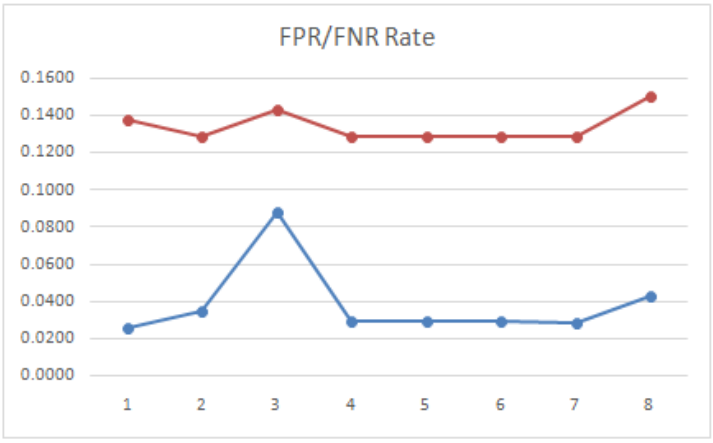

Fig 4.10: With SVM Classifier we See a great reduction in FPR and FNR rates that correspond to the inaccuracy of the model, FPR and FNR have dropped 10% which is quite significant

 ISSN: 2321-8169
147 - 155

Decision Tree	77.53	76.72	45.42	77.73	77.75	77.56	78.15	75.51
SVM	95.0391	96.3295	91.9542	96.3416	93.0104	91.6528	91.4818	93.3841

Table 4.3: Data Sample wise comparison of Decision tree and SVM Classifiers

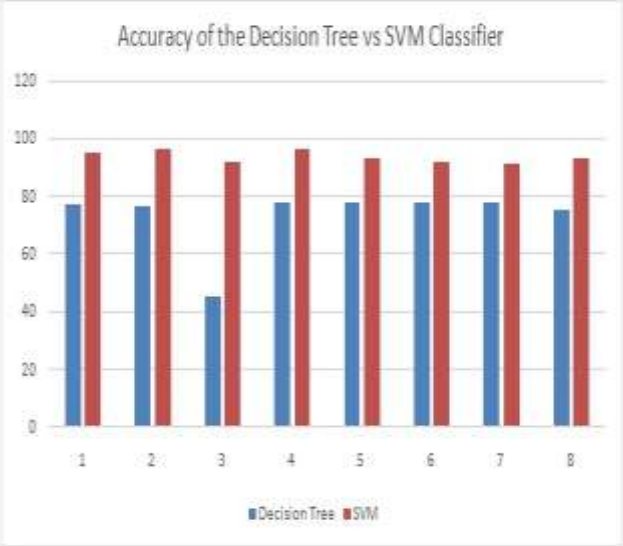

Table: Accuracy of Decision tree and SVM Classifiers for data samples

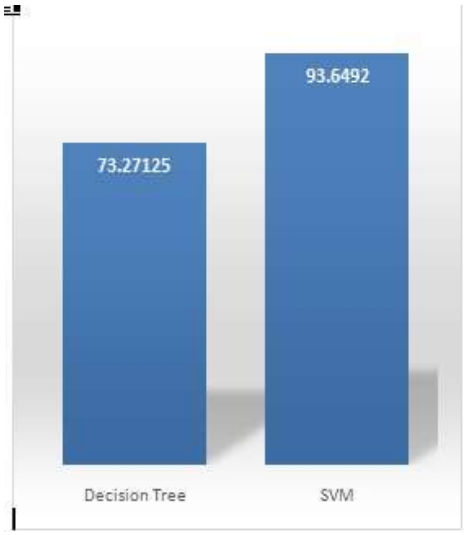

Fig 4.11 : Average Accuracy of Decision tree and SVM Classifier

V. CONCLUSIONS

The application of Honeypot after the main consideration should be given to the security problem of network environment. The goal is to find the research honeypot attack, analysis of unknown species, and not known type of attack. It takes into account the problem of data analysis, and the amount of data generated by a hacker attack is amazing. These problems need to be further explored and studied. The aim is to get the better predictive performance of such algorithms by extenuating three of their primary disadvantages. The consistency of the data in the decision tree relies on upon feeding the exact inner and outer data at the arrival.

Other primary error of the decision tree examination is that the decisions enclosed within the decision tree are dependent on the expectations, and unreasonable expectations can direct to the flaws and errors in the decision tree. SVM is a influential machine erudition tool that is dependent upon firm arithmetical and mathematical fundamentals pertaining to generalization and optimization hypothesis. It offers a robust technique for many aspects of data mining including classification, regression, and outlier detection. SVM is an alluring process because of its high generalizing ability and its capability to hold high-dimensional contribution data. In contrast to neural systems or decision trees, it has some learning parameters to choose, and to produce some stable and reproducible outcomes.

FUTURE SCOPE

Making the new rule generated automatically from server honeypot to server IDS Honeypot was successfully performed using SVM Classifier. Support vectors Machines obtained from the Honeypot IDS can be successfully sent to the server, and then based on the log that is obtained by IDS server created rule. In this paper a successful rule generated is still in the form of alerts if any illegal activity coming into the network, is expected to further the development of systems that can be made a rule to block illegal activity.

New attack pattern will emerge; still, this attack was not handling by the snort rule. This traffic will need to further investigated, so that can result with a new rule. In this research, we used only SVM classifiers statistics to analyze the traffics, further research, new Ensemble classifier based approach which is more effective and efficient should be done for improving accuracy. We can also work on analyzing Snort Data.

REFERENCES

[1]. John P. John, Fang Yu, Yinglian Xie, Arvind Krishnamurthy, and Martín Abadi. "Heat-seeking honeypots: design and experience." In Proceedings of the

20th international conference on World wide web, pp. 207-216. ACM, 2011.

[2]. K. Munivara Prasad, A. Reddy, and M. Ganesh Karthik. "Flooding attacks to internet threat monitors (ITM): modeling and counter measures using botnet and honeypots." arXiv preprint arXiv:1201.2481 (2012).

[3]. Deepa Srinivasan, and Xuxian Jiang. "Time-traveling forensic analysis of vm-based high-interaction honeypots." In Security and Privacy in Communication Networks, pp. 209-226. Springer Berlin Heidelberg, 2012.

[4]. K. Munivara, Prasad A. Reddy, and V. Jyothsna. "IP traceback for flooding attacks on internet threat monitors (ITM) using honeypots." arXiv preprint arXiv:1202.4530 (2012).

[5]. Osama Hayatle, Hadi Otrok, and Amr Youssef. "A game theoretic investigation for high interaction honeypots." In Communications (ICC), 2012 IEEE International Conference on, pp. 6662-6667. IEEE, 2012.

[6]. Yilun Zhao and Jeremy J. Blum. "Agent-based optimization of emulations of network server applications in honeypots." In Consumer Electronics (ISCE), 2012 IEEE 16th International Symposium on, pp. 1-6. IEEE, 2012.

[7]. Hong-Geun Kim, Dong-Jin Kim, Seong-Je Cho, Moonju Park, and Minkyu Park. "Efficient Detection of Malicious Web Pages Using High-Interaction Client Honeypots." Journal of Information Science and Engineering 28, no. 5 (2012): 911-924.

[8]. Tammo Krueger, Hugo Gascon, Nicole Krämer, and Konrad Rieck. "Learning stateful models for network honeypots." In Proceedings of the 5th ACM workshop on Security and artificial intelligence, pp. 37-48. ACM, 2012.

[9]. Bhumika, and V. Sharma. "Use of Honeypots to Increase Awareness regarding Network Security." International Journal of Recent Technology and Engineering 1, no. 2 (2012): 171-175.

[10]. ROMAN JASEK, MARTIN KOLARIK, and TOMAS VYMOLA. "APT Detection System Using Honeypots." In Proceedings of the 13th International Conference on Applied Informatics and Communications (AIC'13), WSEAS Press, pp. 25-29. 2013.

[11]. Sanjeev Kumar, Rakesh Sehgal, Paramdeep Singh, and Ankit Chaudhary. "Nepenthes Honeypots Based Botnet Detection." arXiv preprint arXiv:1303.3071 (2013).

[12]. Paulo Simões, Tiago Cruz, Jorge Gomes, and Edmundo Monteiro. "On the use of Honeypots for Detecting Cyber Attacks on Industrial Control Networks." Inproc of 12th European Conf. on Information Warfare and Security (ECIW 2013). 2013.

[13]. Urban, Bilstrup and Melanie Rosenberg. "A Pilot Study of Using Honeypots as Cyber Intelligence Sources." In Intelligence and Security Informatics Conference (EISIC), 2013 European, pp. 224-224. IEEE, 2013.

[14]. Craig Valli, Priya Rabadia, and Andrew Woodward. "Patterns and Patter-An Investigation into SSH Activity Using Kippo Honeypots." (2013).

[15]. Osama Hayatle, Hadi Otrok, and Amr Youssef. "A Markov Decision Process Model for High Interaction Honeypots." Information Security Journal: A Global Perspective 22, no. 4 (2013): 159-170.