International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 6 | ISSN : 2456-3307

Analyse Honey Pot Traffics to Detect DoS Attacks Using Support Vector Machine

C. Naveena[1], R. Sasikala[2]

[1]M.Phil Research Scholar, Department of Computer Science, Sankara College of Science and Commerce, Coimbatore, Tamil Nadu, India

[2]Assistant Professor, Department of Computer Science, Sankara College of Science and Commerce, Coimbatore, Tamil Nadu, India

ABSTRACT

Honeypots are physical or virtual machines successfully used as intrusion detection tools to detect worm-infected hosts. Denial of service (DoS) attack consumes the resources of a remote client or network itself, there by denying or degrading the service to the legitimate users. In this paper, we present a system that helps in the detection DoS attacks using the data-mining framework. We have used support vector machine classifier to identify the honey pot traffic into normal and DoS attack.

Keywords: Honey pot, Dos, Data mining, SVM.

I. INTRODUCTION

Network Denial-of-Service (DoS) attacks pretence an increasing threat to both public services, such as Google, and private services, such as subscription-based business services, deployed over the Internet. Distinctive private services consist of front-ends (e.g., web servers) and back-ends (e.g., database servers) [1]. Whereas service front-ends can be protected from DoS attacks by massive replication, as in Content Distribution Networks (CDNs) such as Akamai, service back-ends cannot tolerate the same level of replication, because of higher costs and tighter consistency constraints. DoS attacks are difficult to prevent because of inevitable software vulnerabilities, which get exploited by attackers either to directly crash a victim or to compromise *zombie* machines, which are unwittingly used to launch the attack.

Network-level DoS attacks aim at congesting network resources, such as link capacity and router buffers, by flooding them with bogus packets sometimes with spoofed (forged) source addresses. However, wide deployment of source-end DoS defense systems, such as *D-WARD* [2], which autonomously detects and stops abnormal one-way flows, and *Ingress Filtering*, which stops most spoofed attacks, would limit the pervasiveness and effectiveness of network-level and spoofed attacks, leaving floor to *service level* DoS attacks. In service-level DoS attacks, a large number of attack machines manage to acquire service from a victim server, consuming both service-level resources, such as server memory and processing time, as well as network level resources along the path outward from the server.

Honeypots [3], a proactive detection mechanism, are machines that are not supposed to receive any legitimate traffic and, thus, any traffic destined to a honeypot is most probably an ongoing attack and can be analyzed to reveal vulnerabilities targeted by attackers. Coupled with an Intrusion Detection System (IDS) (e.g., [4]), honeypots are effective in detecting hosts exploited by Internet worms that perform random scanning. However, since honeypots are deployed at fixed, detectable locations and on machines different than the ones they are supposed to protect, sophisticated attacks can avoid the honeypots. In this research paper we are focusing honeypot traffics to detect dos attack at network level.

II. RELATED WORKS

In [5] the authors have built a web application honey pot that emulates XSS and SQL injection vulnerabilities found in web applications. In addition, it will dig up cracker's information using JavaScript code. If the request to a honeypot is a normal HTTP request, the honey pot will give a normal response anyway. However, if there is an indication of a threat, honey pot will then simulate these attacks and sends the response as if the attacks succeeded. For every request sent by attacker's browser, our proposed honeypot system will insert JavaScript codes into the response. These codes will be executed by the cracker's browser and collect certain information to be sent back to honey pot.

The system introduces in the [6] implements high-interaction honeypot with Secure Shell installed to study common SSH attacks in Linux environment. This system records usernames and passwords that are attempted by an intruder from the Internet. It also captures detail activities of the attackers while they are interacting inside the target honeypot. Intruders attack SSH servers through dictionary and brute-force mechanism followed by the intrusion. This paper covers both dictionary attack and intrusion. Secure Shell (SSH), is an encrypted channel to communicate remotely which is used mainly in Linux and Unix-based operating systems. SSH uses port 22 to login into a remote machine using usernames and passwords. Even if, the username/password combination mechanism can be replaced with public key authentication, brute-force attacks against the SSH protocol become quite common. Attackers can create automated tools to attack SSH servers using either brute-force or dictionary-based attack methods.

III. PROPOSED WORK

An overview of the complete process of DoS attack detection system using honey pot traffics and data mining technique is shown in figure 1.

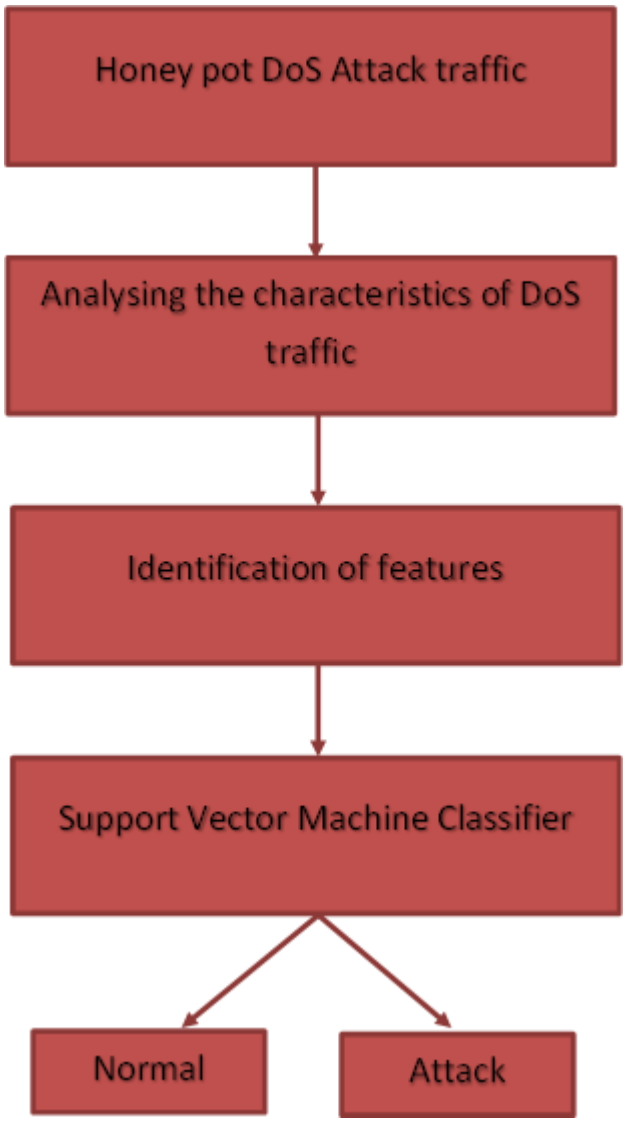

Figure 1. Proposed System

The preliminary step for the detection of DoS attacks in network is data collection and necessary preprocessing to convert it into a form, which can be used by the learning algorithms.

A. Honey pot based DoS attack traffic

In this paper, we have used the dataset obtained from [7] which consists of Honey pot DoS attack traffic. Dataset comprises of 247 MB of data which provides a complete DoS attack scenario in honey pot infrastructure environment.

B. Analyzing the characteristics of DoS traffic

A DoS attack is initiated by sending needless and superfluous messages to the server/network for authentication of requests having invalid return addresses.

The server/network, when unable to locate the return address for sending authentication, waits for a long time and gets stuck before the connection closes. Upon the closure of connection, the attacker once again starts sending more messages with invalid return addresses for authentication to make the server/network undergo the complete process again. The server/network gets stuck and remains busy, causing the service interruption for other users.

Unlike other security attacks, DoS attacks usually do not aim at breach of security. Rather, they are focused on making websites and services unavailable to genuine users resulting in loss of time and money. These attacks can last many days, jeopardizing the image of an organization and causing revenue loss towards compensation to users for unavailability of services at the time of an emergency.

DoS attacks can be of various types depending on the outcomes. Some examples are Smurf attack, Ping flood, Ping of death, Teardrop attack, Email bomb, etc. Also, the motive of these attacks could be many, including extortion, personal rivalry, cyber warfare, business competition, etc.

C. Feature Identification

The basic features are directly extracted or derived from the header information of IP packets and TCP/UDP segments in the tcpdump files of each session [8]. The so-called content-based higher-level features use domain knowledge to look specifically for attacks in the actual data of the segments recorded in the tcpdump files. These address ‘r2l’ and ‘u2r’ attacks, which sometimes require only a single connection or are without any prominent sequential patterns [9]. Typical features include the number of failed login attempts or whether root access was obtained during the session. The selected features are listed in the table 1.

S.No	Features Name
1	Duration
2	Protocol_type
3	Service
4	Src_bytes
5	Dst_bytes
6	Flag
7	Land
8	Wrong_fragment
9	Urgent

Table 1. Selected Features

D. Support Vector Machine classifier

Support Vector Machines (SVMs) are learning machines that plot the training vectors in high dimensional feature space, labeling each vector by its class. SVMs classify data by determining a set of support vectors which are members of the set of training inputs that outline a' hyper plane in the feature space. SVMs provide a generic mechanism to .fit a hyper plane to perform a linear classification of the patterns through the use of a kernel function [10]. The user may provide a function (e.g., linear, polynomial, or sigmoid) to the SVMs during the training process, which selects support vectors. The number of free parameters used in the SVMs depends on the margin that separates the data points but not on the number of input features, thus SVMs do not require a reduction in the number of features in order to avoid over fitting-an apparent advantage in applications such as intrusion detection. Another primary advantage of SVMs is the low expected probability of generalization errors.

IV. EXPERIMENTAL RESULTS

The experiments are conducted using support vector machine classifier to build an efficient honeypot based DoS attack detection system. We have evaluated our classifier with various evaluation measures, such as accuracy, F-measure and false positive rate.

Accuracy is percentage of correctly identified Internet worms.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

True Positive (TP) =Number of samples correctly predicted as DoS attack.
False Positive (FP) = Number of samples incorrectly predicted as DoS attack.
True Negative (TN) = Number of samples correctly predicted as normal.
False Negative (FN) = Number of samples incorrectly predicted as normal.

Precision is a measure of what fraction of test data is detected as worm are actually from the worm classes.

$$\text{Precision (P)} = \frac{TP}{TP+FP}$$

Recall measures the fraction of worm class that was correctly detected.

$$\text{Recall (R)} = \frac{TP}{TP+FN}$$

F-Measure is a measure of test's accuracy, which measures the balance between precision and recall.

$$\text{F-Measure} = \frac{2 * Precision * Recall}{Precision + Recall}$$

False Positive Rate (FPR) is percentage of wrongly identified normal classes.

$$\text{Positive Rate (FPR)} = \frac{FP}{FP+TN}$$

The experimental results are given in table 3.

Measures	**Values**
Precision	0.96
Recall	0.970
F-Measure	0.949
Accuracy	95.90
False Positive Rate	0.57

Table 3. Experimental results of neural network Classifier

From the table, one can observe that the support vector machine classifier achieves highest accuracy of 95.90% and false positive rate of 0.57 when detecting the honeypot based DoS attack traffic.

V. CONCLUSION

This paper has analyzed existing honeypot system to identify significant parameters to identify the DoS attack effectively. Honeypots are a powerful tool for detecting unknown attacks. Because it only has malicious traffic, it is easier to identify an attack. They can be classified as low-level interaction and high-level interaction honeypots. We have used support vector machine classifier to identify the DoS attack traffic. Through this classification we have achieved the accuracy of 95.90%.

VI. REFERENCES

[1]. Weiler, N. (2002). Honeypots for distributed enial-of-service attacks. In Enabling Technologies: Infrastructure for Collaborative Enterprises, 2002. WET ICE 2002. Proceedings. Eleventh IEEE International Workshops on (pp. 109-114). IEEE.

[2]. Bhunia, S., Su, X., Sengupta, S., & Vazquez-Abad, F. (2014, January). Stochastic model for cognitive radio networks under jamming attacks and honeypot-based prevention. In International Conference on Distributed Computing and Networking (pp. 438-452). Springer, Berlin, Heidelberg.

[3]. Deshpande, H. A. (2015). HoneyMesh: Preventing Distributed Denial of Service Attacks using Virtualized Honeypots. arXiv preprint arXiv:1508.05002.

[4]. Mirza, M., Usman, M., Biuk-Aghai, R. P., & Fong, S. (2016). A Modular Approach for Implementation of Honeypots in Cyber Security. International Journal of Applied Engineering Research, 11(8), 5446-5451.

[5]. Djanali, S., Arunanto, F. X., Pratomo, B. A., Baihaqi, A., Studiawan, H., & Shiddiqi, A. M. (2014, November). Aggressive web application honeypot for exposing attacker's identity. In Information Technology, Computer and Electrical Engineering (ICITACEE), 2014 1st International Conference on (pp. 212-216). IEEE.

[6]. Zemene, M. S., & Avadhani, P. S. (2015, August). Implementing high interaction honeypot to study SSH attacks. In Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on (pp. 1898-1903). IEEE.

[7]. MACCDC - Pcaps from National CyberWatch Mid-Atlantic Collegiate Cyber Defense Competition https://www.netresec.com/?page=MACCDC

[8]. Zhan, Z., Xu, M., & Xu, S. (2013). Characterizing honeypot-captured cyber attacks: Statistical framework and case study. IEEE Transactions on Information Forensics and Security, 8(11), 1775-1789.

[9]. Dongxia, L., & Yongbo, Z. (2012, March). An intrusion detection system based on honeypot technology. In Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on (Vol. 1, pp. 451-454). IEEE.

[10]. Yang, Y., & Mi, J. (2010, April). Design and implementation of distributed intrusion detection system based on honeypot. In Computer Engineering and Technology (ICCET), 2010 2nd International Conference on (Vol. 6, pp. V6-260). IEEE.