International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.4, No.4, April 2015 DOI:10.15693/ijaist/2015.v4i2.476-480

Malicious Node Detection by Using Frequency Based Approach

Ms. P. Gowrilakshmi
P.G Student of ECE Department
P.S.R.Rengasamy college of engineering for women

Mr. N.R. Sathis kumar
Assistant Professor of ECE Department
P.S.R.Rengasamy college of engineering for women

Dr. K. Ramasamy
Principal & Associate Professor of P.S.R.Rengasamy college of engineering for women

***Abstract*—The adhoc networks are very much vulnerable to Dos attacks on the network layer. Black hole attack, gray hole attack are the widespread attack on adhoc networks. Here the malicious nodes interrupt data transmission in the network by transmitting false routing information. The purpose of this paper is to present an efficient Adhoc On-demand Distance Vector (AODV) protocol that removes the malicious node by isolating it, thereby ensuring secure communication. In order to achieve this goal, the intermediate node receiving false routing information from its neighbor node is programmed to consider that neighbor node as malicious. In adhoc network, nodes can join or leave at any time. So, an efficient security mechanism is needed. A homomorphic algorithm is used among authenticated neighbors' in the adhoc network to provide more security. To reduce the energy consumption of the nodes, data compression techniques are used.**

Index terms—MANETs AODV, LZW data compression techniques.

I. INTRODUCTION

In current network infrastructure, network intrusions have already become a critical issue for the whole network communications according to the Microsoft Security Intelligence Report 2011 [1]. To mitigate this issue, network intrusion detection systems (NIDSs) [2, 3] have been widely implemented in current network environment, aiming to enhance network security by defending against different kinds of network attacks (e.g., virus, Trojan, worm). These intrusion detection systems are also prevalent to be deployed in a distributed environment such as agent-based network and mobile ad hoc network (MANET).

Generally, network intrusion detection systems can be categorized into two folders: signature-based NIDS and anomaly-based NIDS. The signature-based NIDS [4, 5] (also called rule-based NIDS or misuse-based NIDS) detects an attack by comparing its signatures1 (also called rules) with incoming packet payloads. An alarm will be triggered, alerting relevant attack situation, if and only if an accurate match is identified. On the other hand, the anomaly based NIDS [7, 9] identifies an attack by discovering significant deviations between its established normal profile2 and current observed network events. Based on its detection method, the anomaly based NIDS has the capability of discovering novel network attacks. However, it is very hard for an anomaly-based NIDS to model and establish an accurate normal profile in most cases since the network traffic is flexible and irregular [6]. Therefore, in real deployments, the signature-based NIDS is more popular and widely used than the anomaly-based NIDS on that false alarm rate (FAR), which indicates the possibility that an intrusion detected when there is no intrusion, of the signature-based approach is far lower than that of the anomaly-based approach [8]. Over the past ten years, NIDSs have already been applied in a distributed network environment that carries out local detection and perform global analysis and detection. Spafford and Zamboni [10] defined such distributed NIDSs according to their locations and the number of data analysis components.

In general, a distributed NIDS consists of multiple NIDS agents, which are deployed in different network locations, and a central data analysis component. All these distributed NIDSs (e.g., IDS [11], GrIDS [12], AAFID [10]) are organized in a hierarchical structure. The local detection components (e.g., a NIDS agent) can detect local intrusions and pass their analysis results and identified patterns in higher levels (e.g., a central analysis server) of the hierarchical structure. On the whole, a distributed NIDS can detect and respond to network attacks by collecting network data and analyzing security events from its NIDS agents. By deploying multiple agents, the distributed system can overall improve the capability of a detection system to detect some particular attacks (e.g., DoS [17]) that are very hard to be identified by a single NIDS Problem.

Although the signature-based NIDSs have been widely implemented in various organizations (e.g., insurance companies, banks), their poor performance in a high volume traffic environment is a big problem to impede their development. The time consumed for a signature-based NIDS is mainly spent in comparing their signatures with network packet payloads in which the processing burden is at least

linear to the size of an input payload string [16]. For instance, Snort [4, 14] which is an open-source, lightweight signature-based network intrusion detection system, can quickly exhaust computer resources in a high volume traffic environment. It generally spends nearly 30 percent of its total processing time in comparing its signatures with incoming payloads, while its consuming time can exceed 80 percent when deployed in a heavy traffic environment [13, 15]. In this case, Snort has to drop a massive of network packets which can severely decrease the whole network security and cause a lot of security risks (i.e., missing a lot of malicious network packets).

In this paper, we adapt the frequency analysis techniques such as the Discrete Fourier Transform (DFT) used in signal processing for the design of intrusion detection algorithms. We demonstrate the effectiveness of the frequency based detection strategy by running synthetic network intrusion data in simulated networks using the OPNET software. The remainder of the paper is organized as follows. Section 2 describes our frequency-based approach to intrusion detection. Section 3 reports experimental results evaluating the effectiveness of the detection strategy using simulated network traffic data that contain DOS attack packets. Finally, Section 4 concludes the paper and points out directions for further research.

II. Frequency Based Intrusion Detection

Our work includes network intrusion detection and simulation. The contributions that we have made for network intrusion detection are:

First, we proposed a frequency-based anomaly detection method. It searches for periodical patterns in the various time-sequence created by the attack traffic. The method aims at denial of service attack, Probes and certain other types of attacks. We achieved better results with the experiments on DARPA 1999 synthetic data set.

Second, we proposed another novel distributed network anomaly detection system, which is composed of end user agents and a centralized graphical analyzer. A user agent observes the behavior change at the connection level of each computer, in particular, the appearance of new connections and the sudden changes of the once trusted connections. A novel weight assignment schema is developed within each local agent to quantitatively carry out the local abnormalities in the format of node weight and link weight. The weight information is, then, further sent to the central analyzer to build the weighted link graph. The graph generated by the analyzer integrates those otherwise isolated connection abnormalities into a global map to facilitate the understanding of the intrusion incidents within a LAN.

Therefore, the contributions made by our system are:

- First, our system distributes the intrusion detection process among different computers, which effectively divide and digest each part of the network activities in parallel.
- Second, our system provides two layers of protection to a LAN, at the first layer, each agent provides the local intrusion detection as independent personal IDS, at the second layer, the central analyzer automatically identifies the attacks that would create the characteristic graph shapes, and further locates the source of these attacks.
- Finally, a novel weight assignment based on the connection behavior analysis is used by each agent to report the abnormalities to the central analyzer. These abnormalities are presented in the weighted link graph by the central analyzer using a spring-based graph drawing tool, which make the intrusion visualization legible and meaningful.

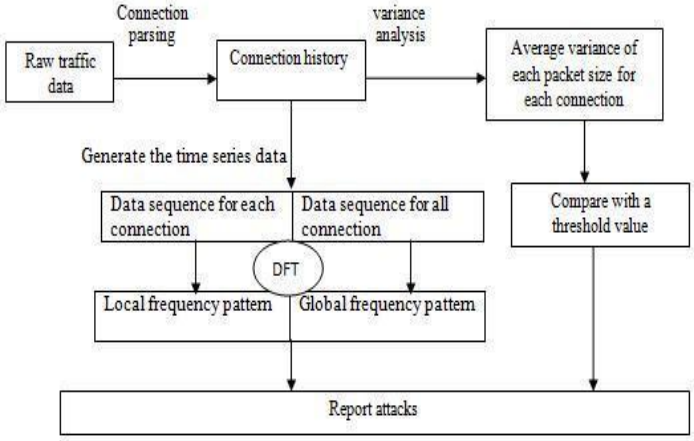


Figure 1: Strategy of frequency-based intrusion detection

For the network intrusion simulation, we built a simulation model to simulate the DOS attack traffic using the explicit traffic generation. The explicit traffic simulation for intrusion study has rarely been done in other works. Most approaches for intrusion simulation use analytical traffic simulation, which use global parameters to control the simulation process and does not provide the functionality to simulate every network packet and their logically movement. Our work simulates the intrusions using explicit traffic, which makes packet level analysis possible.

A. Frequency extraction

For a given data sequence s(n) where $n \geq 0$ is a discrete value representing the time, we apply the Discrete Fourier Transform (DFT) to compute the frequency information. Fourier Transform is a well-known tool used in signal processing. The DFT takes the original time series in the time domain, and transforms them into the associated frequency data in the frequency domain. The DFT coefficients are defined as follows:

$$F(K) = \sum_{n=0}^{N-1} s(n) e^{-i2\pi Kn/N}$$

Expanding the right-hand side yields the following:

$$F(K)=\sum_{n=0}^{N-1} s(n)\cos(2\pi Kn/N)-i\sum_{n=0}^{N-1} s(n)\sin(2\pi Kn/N)$$

Where N is the length of s (n). Using the Fast Fourier Transformation (FFT) procedure, the frequency data $F(k)$ can be computed in O (N log N) time.

B. The average variation of packet sizes

The average variance ζ of the packet size $p(i)$, $1 \leq i \leq n$, for a connection is defined as follows:

$$\zeta = \frac{1}{n}\sum_{i=1}^{n} \sqrt{(p(i)-\mu)^2}$$

Where n is the number of packets within a connection, and μ is the mean of packet sizes defined as follows:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} p(i)$$

When the goal of an attack is to consume the resources of the victim machine or to probe the information, the payload of each packet normally would be fabricated and is of a similar size. As a result, the average packet size variance would be relatively small of an attack traffic compared to that of the normal traffic. This factor is used in detecting intrusions.

III. TESTING THE DETECTION STRATEGY

To implement our intrusion detection strategy, a firewall is placed on each of the target machines to capture the network traffic data. The firewall maintains a table of the recent connections within a given time window. Each table entry contains an IP address that has been connected to the protected computer at least once within the time window. The traffic data of each connection are converted to a time series; the average variance of the packet sizes is computed and recorded by the firewall. In addition, information about port connections are recorded which will be useful in detecting probe attacks. Finally, the firewall will report the suspicious IPs based on the observed frequency patterns and the variance of packet sizes. In the next section, we will describe our intrusion simulation framework using NS2, and report the simulation results evaluating the frequency based intrusion detection strategies.

IV.SIMULATION STUDY OF NETWORK INTRUSION DETECTION

Intrusion detection algorithms are typically effective in detecting intrusions of known signatures, but poor at detecting new attacks. Studying and testing a new intrusion detection algorithm against a variety of intrusive activities under realistic background traffic is an interesting and difficult problem. Such studies can be performed either in a real environment or in a simulated environment. We now briefly summarize these two approaches of simulating intrusion and their relative merits, followed by a description of our approach.

A. Simulation in Real Environments

In this approach many real users produce significant background traffic by using a variety of network services, e.g., mail, Telnet, etc. This background traffic is collected and recorded, and intrusive activities can be emulated by running exploit scripts. The advantage of this approach is that the background traffic is sufficiently realistic, which eliminates the need of analyzing and simulating the patterns of normal user activities. However, the following drawbacks have been reported. The testing environment may be exposed to the attacks from the Internet, in addition to the simulated attacks. The inaccuracy of the results may increase if the background traffic contains unexpected intrusive data originated from outside sources. ” Normal system operations could be interrupted by simulated attacks.

B. Simulation in Experimental Environments

Most researchers perform testing and studies in experimental or simulated environments, due to the high risk of performing tests in a real environment. In this approach realistic background traffic can be generated in the following three ways. (1) Employing humans to manually perform whatever occurs in a real environment. (2) Using simulation scripts to generate data conforming to the statistical distributions of background traffic in a real environment. (3) Collecting from a real environment and replicating in the simulated environment.

The disadvantages of studying intrusions in simulated environments include:

– The overload of manually producing background traffic involving many users can be very high.

– The behaviors of real users are difficult to model, and there are no standardized statistical distributions of the background traffic.

C. Network Intrusion Simulation Using NS2

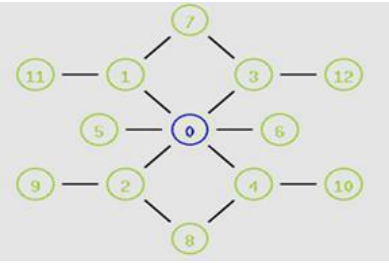


Figure 2: Network intrusion simulation

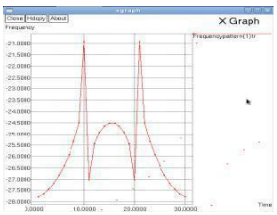
NS2 and OPNET are two popular tools used for network simulation. Our studies use NS2 as a simulation platform for testing the intrusion detection strategies. Figure2 shows the NS2 model for simulating the DoS attack. The number of virtual PC nodes arranged like star topology in the figure 2. The top node in the center column is the "generator", which prepares the packets extracted from the traffic source. Once a packet is ready, it is given to its source PC node, and from there it will be sent to the destination PC node through the hub (located at the bottom of the center column). There is no delay between the generator and the end PC nodes, so the traffic flow is consistent with the captured traffic source. The number of the virtual PCs is the outcome of preprocessing the source traffic file. Since there are 12 distinct IP addresses in the source, the model uses 12 PC nodes connected to each other through a hub. Node 0 (the top node in the right side column) is the "hacker", and node 1 (below node 0 in the figure) is the "victim" of the DoS attack. There is a "firewall" node between the victim and the hub, which we use to capture suspicious data packets to or from the victim using the DoS attack's signature.

We have to create 13 nodes and create a link between each node. Here star topology is used to construct a network. The link definition is used to give a node position (for NAM) and generate a connection history, according to the link created in the network. The agent is used to set traffic for both global and local networks that have packet size and interval. They can also record traffic for each node to find the attacker node. In DFT they can assign both real and imaginary value. The missing roots are computed by a complex conjugating the given root. These processes are followed by naïve formula

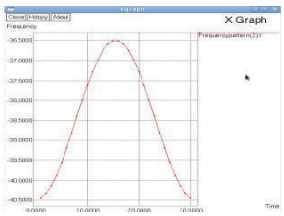
The steps that can be carried out in DFT are first converted to internal format and compute a list of n^{th} roots of unity and check if the input length is about the power of two because of construction p is the power of two.

Finally, compute the transform using the fast DFT or slow DFT and again, they convert back into the input format. After applying the DFT we can compute the average variance of the packet size for each connection for finding the mean value of the packet size. Finally, we can correlate the average variance of packet size with the frequency patterns.

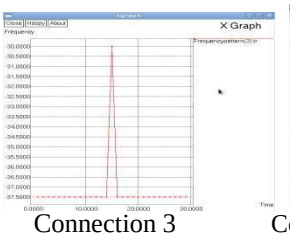
If the average variance will be below two means the node is not affected by the attacks. If above two the node can be affected by the attacks.



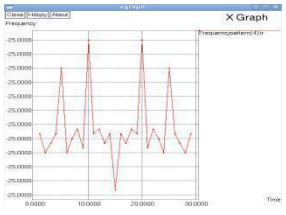
Connection 1



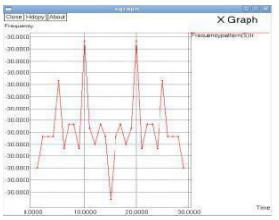
Connection 2



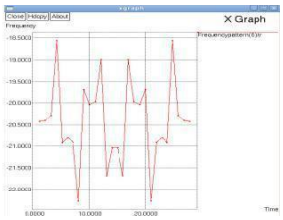
Connection 3



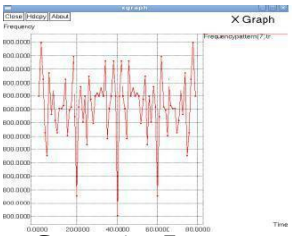
Connection 4



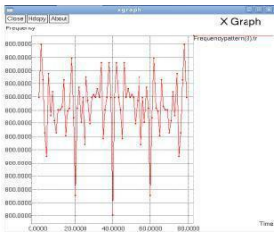
Connection 5



Connection 6



Connection 7



Connection 8

Figure 3: Frequency patterns

The frequency pattern is plotted on the above figure. The distinctive frequency patterns are generated in it. From the connection1 to connection6 we have a frequency pattern value as -40.5000 to -18.5000 & for connection7 & connection8 we have a frequency value as 800. Because the connection7 & connection8 are affected by the DoS attack in it

V. CONCLUSION & FUTURE WORK

In this paper, we proposed a novel frequency-based intrusion detection strategy for detecting the automated, scripted attacks, which typically exhibit frequency patterns over time. We integrated this strategy into a simulation framework using NS2. With the aid of our preprocessing tools, we extracted relevant information from the previously captured data sources, and instantiated the NS2 simulation model. The experimental results demonstrated that the frequency-based intrusion detection approach is effective in, but not limited to, detecting the DOS and probe attacks that typically run from prewritten scripts and have a relatively long duration. But there are still some possible issues that we can improve in future experiments. The future work could include measuring our scheme with more parameters such as bit-string length, packet size and false match rate. To investigate the particular effect of these parameters can lead us to better understand the overall performance our scheme. In addition, our future work could also include using the method of link-analysis to evaluate network traffic and identify network attacks by analyzing signatures in MNS tables from different agents.

REFERENCES

[1] Microsoft Security Intelligence Report (SIR),Volume11,2011.http://www.microsoft.com/security/sir/default.px

[2] V. Paxson, "Bro: A System for Detecting Network Intruders in Real-Time," Computer Networks 31(23-24), pp. 2435–2463, 1999.

[3] K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)," NIST Special Publication 800-94, Feb 2007.

[4] M. Roesch, "Snort: Lightweight intrusion detection for networks," In: Proceedings of Usenix Lisa Conference, pp. 229–238, 1999.

[5] G. Vigna, R.A. Kemmerer, "NetSTAT: A Network-based Intrusion Detection Approach," In: Proceedings of Annual Computer Security Applications Conference (ACSAC), pp. 25–34, 1998.

[6] S. Axelsson, "The base-rate fallacy and the difficulty of intrusion detection," ACM Transactions on Information and System Security, 3(3), pp. 186–205, 2000.

[7] A. Valdes and D. Anderson, "Statistical Methods for Computer Usage Anomaly Detection Using NIDES," Technical Report, SRI International, January 1995.

[8] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," In: Proceedings of IEEE Symposium on Security and Privacy, pp. 305–316, 2010.

[9] A.K. Ghosh, J. Wanken, and F. Charron, "Detecting Anomalous and Unknown Intrusions Against Programs," In: Proceedings of Annual Computer Security Applications Conference (ACSAC), pp. 259–267, 1998.

[10] E.H. Spafford and D. Zamboni, "Intrusion detection using autonomous agents," Computer Networks, 34(4), pp. 547– 570, October 2000.

[11] S.R. Snapp, J. Brentano, G.V. Dias, T.L. Goan, L.T. Heberlein, C.L. Ho, K.N. Levitt, B. Mukherjee, S.E. Smaha, T. Grance, D.M. Teal, and D. Mansur, "DIDS (distributed intrusion detection system) motivation, architecture, and an early prototype," In: Proceedings of National Computer Security Conference, pp. 211–227, ACM Press, 1998.

[12] S. Staniford-chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagl, K. Levitt, C. Wee, R. Yip, and D. Zerkle, "GrIDS - A Graph Based Intrusion Detection System For Large Networks," In: Proceedings of the 19th National Information Systems Security Conference, pp. 361-370, 1996.

[13] R.L. Rivest, "On the worst-case behavior of string-searching algorithms," SIAM Journal on Computing, pp. 669–674, December 1977.

[14] Snort, The Open Source Network Intrusion Detection System. http://www.snort.org/.

[15] M. Fisk and G. Varghese, "An analysis of fast string matching applied to content-based forwarding and intrusion detection," Technical Report CS2001-0670, University of California, San Diego, 2002.

[16] H. Dreger, A. Feldmann, V. Paxson, and R. Sommer, "Operational experiences with high-volume network intrusion detection," In: Proceedings of the 11th ACM conference on Computer and communications security, pp. 2–11, ACM Press, 2004.

[17] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," ACM SIGCOMM Computer Communication Review, 34(2), pp. 39–53, 2004.

[18] Markku Antikainen, Tuomas Aura, and Mikko Särelä, "Denial-of-Service Attacks in Bloom-Filter-Based Forwarding," IEEE/ACM transactions on networking, vol. 22, no. 5, october 2014.