Research Journal of Applied Sciences 9 (5): 257-261, 2014
ISSN: 1815-932X

A Software Based Approach for Detecting Intrusion in Wireless Ad Hoc Networks

[1]M. Ravichandran and [2]C.S. Ravichandran
[1]Department of Electronics and Communication Engineering,
Saveetha School of Engineering, Saveetha University, Sriperumbudur, Tamil Nadu, India
[2]Department of EEE, Sri Ramakrishna Engineering College, Coimbatore, India

Abstract: Intrusion Detection Systems (IDS) are progressively becoming a key part of network security by supervising the network traffic and attempting to identify and alert all malicious behavior to the user. In this study, Snort, a Signature-Based IDS that identifies attacks in a network traffic based on the rules written and SPADE, a Statistical-Based IDS that flags for anomalous behavior in the network traffic by learning the normal behavior for the packets in that particular network are both used. The integration of Snort and SPADE improves the detection accuracy and a vast majority of anomalous traffic on the network can be identified. However, the system is still prone to false positive errors that occur when a normal activity is misclassified as an attack. This leads to the anomaly based IDS to produce undesirable results as normal packets are being classified as malicious packets and are dropped or rejected by the system and also raising frequent false alerts. The main objective of this research is to find a scenario in which false positives are generated by SPADE and proceed to modify SPADE in such a way that it can be deployed effectively in a Wireless Ad Hoc Network.

Key words: Snort, SPADE, anomaly, wireless ad hoc, false alarms

INTRODUCTION

An Intrusion Detection System (IDS) uses a defined set of rules that have been designed to detect a malicious event. This is often referred to as "misuse" detection. An Anomaly based Intrusion Detection System, on the other hand, operates only from a baseline of normal behavior. While, an IDS looks for a misuse signature, the Anomaly based IDS looks for a strange activity. Anomaly detection can be described as an alarm for strange system behavior. An 'activity profile' of normal usage over a specific user defined interval of time is built. Once it is built, the profile is compared against the real time events and any action that deviates from the baseline is labeled as anomalous.

Statistics based anomaly detection works best for Wireless Ad Hoc Networks. It employs statistics to construct a point of reference for system behavior. The process begins with the training of an anomaly detection sensor. This is achieved by observing certain specific events in the monitoring environment such as network traffic, over a designated time period. When the interval expires, one of several mathematical methods is used to generate a quantitative measure for the observed data. The result is a baseline for some variable of a system's behavior.

With a point of reference in place, the monitoring process is repeated in a live environment. Recorded data is transformed into the same quantitative metric and compared to the baseline. If the deviation exceeds a specified threshold, the event is labeled as anomalous.

SPADE stands for Statistical Packet Anomaly Detection Engine. It is a plug in for the signature based IDS, Snort. SPADE analyses the packets received by Snort and report those packets that it believes are anomalous along with an anomaly score. The anomaly score is determined by evaluating the Source IP, Source Port, Destination IP, Destination Port and Time Period among others. Based on the user specified threshold level, SPADE will either flag the packet as anomalous or allow it to pass through the network without notification.

SPADE can also generate other reports of importance such as a survey about the distribution of anomaly scores and various reports about the feature statistics such as conditional probabilities.

SPADE has a lot of functionality and because it is built on Snort, they can be utilized in conjunction with each other as a hybrid IDS solution. Snort benefits the network by alerting on packets with known signatures where SPADE will learn what normal traffic for the network is and raise alert to any discrepancies.

Corresponding Author: M. Ravichandran, Department of Electronics and Communication Engineering,
Saveetha School of Engineering, Saveetha University, Sriperumbudur, Tamil Nadu, India

LITERATURE REVIEW

Types of IDS: Signature based systems are reactive in that they combat against known attacks that have already affected and damaged a number of systems before being identified. Anomaly based systems are proactive and autonomous and can ensure security without any manual interference. In anomaly-based techniques, the network traffic activity is captured and a profile representing its stochastic behavior is created (Kumar, 2007).

Classification of attacks: An IDS can classify the packets into one among the following categories:

- True Positives (TP): a malicious packet is classified as malicious
- False Positives (FP): a non-malicious packet is classified as malicious packet
- True Negatives (TN): a non-malicious packet is classified as non-malicious
- False Negatives (FN): a malicious packet is classified as non-malicious packet

False Positives (FP) and False Negatives (FN) are known as False alarms.

Advantages of Anomaly based IDS: Anomaly based Intrusion Detection System approaches has a number of merits. Firstly, they do not require prior knowledge about the normal activity of the target system, instead they have the ability to learn the expected behavior of the system from observations. Secondly, if statistical methods are used in anomaly based IDS they can provide accurate notification of malicious activities occurring over long periods of time. Typically, the distribution of portscans is highly anomalous in comparison to the usual traffic distribution. This is particularly true when a packet has unusual features. Hence, even portscans that are distributed over a lengthy time frame will be recorded because they will be inherently anomalous.

Haystack: Haystack is one of the earliest examples of a statistical Anomaly based Intrusion Detection System. A range vector is used to report the normal behavior in each session. If during a session, a feature falls outside the normal range, the score for that feature will be raised. Assuming the features were independent, the probability distribution of the scores is calculated and an alarm will be raised if the score is too large. A database of user groups and individual profiles are also maintained. It is designed to detect six types of intrusions: attempted breakins by unauthorized users, masquerade attacks, penetration of the security control system, leakage, DoS attacks and malicious use (Patcha and Park, 2007; Zhang and Lee, 2000; Gaikwad and Kulkarni, 2012).

One of the drawbacks of Haystack is that it is designed to work offline and so was unsuitable for real time IDS which require high performance systems. Another drawback is the determination of the attributes that would be helpful to indicate the anomalous activities more accurately.

ARCHITECTURE

SPADE: Statistical Packet Anomaly Detection Engine (SPADE) is a plug in for the open source IDS Snort (Garcia-Teodoro *et al*., 2009). SPADE utilizes the rule set of Snort. It analyses recorded data for anomalous behavior based on a computed score called anomaly score.

The anomaly score of a packet is a number which measures its degree of strangeness based on the recent history of the network. The scheme is conceptually simple and consists in a frequency-based mechanism: the fewer times a particular packet has been observed, the higher its anomaly score will be (Estevez-Tapiador *et al*., 2004).

SPADE uses joint probability measurements to decide which packets are anomalous. SPADE utilizes Snort's I/O facilities to grab a copy of the packets, analyze them and tabulate the features into tables which are then used to determine the anomaly score. The anomaly score is assigned by evaluating the source IP, source port, destination IP, destination port and time period among others. Based on the user specified threshold level, SPADE will either label the packet as malicious or allow it to pass through the network without any notification. Here, the threshold fixing is crucial in SPADE because if it is set too high, the analysis engine will miss many malicious packets, i.e., it will trigger false negatives. If it is set too low, the analyst will see many false-positives. SPADE also has an option that will perform automatic threshold adjustment to let SPADE decide what the critical threshold number should be.

Snort: Snort is a open source IDS which consists of user defined rule set for the identification of specific attacks. It combines the benefits of signature, protocol, and Anomaly-based Inspection Methods (Anonymous, 2009). It monitors the network traffic and identifies the malicious packets when the specific rule is satisfied.

OSSEC: Open Source Security (OSSEC) is an open source host based IDS that performs functions like log analysis, real time alerting and active response.

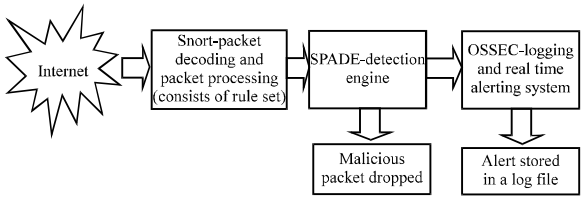

Fig. 1: Architecture of Anomaly based IDS

The proposed architecture utilizes the combination of Snort, SPADE and OSSEC to form an unified efficient IDS (Fig. 1).

ANALZYING SPADE

Attack detected by SPADE: In this research, the capability of SPADE in identifying Port scan attack, a type of intrusion is analyzed. A port scan is a type of intrusion that sends client requests to a range of server port addresses on a host with the objective of finding an active port and exploiting a known vulnerability of that service.

This technique consists of sending a message to a port and listening for a response. The received response indicates port status and can be helpful in determining a host operating system and other information relevant to launching a future attack. When running SPADE in the initial stage with the DARPA dataset, it was found to alert for port scan attacks. Alerts were generated by SPADE for various kinds of port scan namely vertical scan and horizontal scan. A vertical scan consists of a port scan of some or all ports on a single computer. A horizontal scan is a scan of a single port across multiple IP addresses.

False positives in SPADE: Generally, when a port is scanned, a TCP packet with SYN flag set is sent to the particular port. The attacker or the network administrator can deduce if the port is open or closed by the response from the port. If the port is open, it sends a TCP packet back to the target with both SYN and ACK flags set. If the port is closed, it sends a TCP packet with RST and ACK flags set. SPADE generates alerts when an attempt to scan an open port is made. For every scan to an open port, the alerts are generated for the reply TCP packets sent from those ports with both the SYN and ACK flags set. Ports like 80 and 22 are known open ports for http and ssh, respectively. Hence, the SPADE alerts for the responses from those open ports are to be considered as false positives.

Finding false negatives in SPADE: Now that, researchers know SPADE identifies port scan in a network, the task is to find out if it finds all the possible attempts of port scan. A port scan trace is simulated and is pumped to SPADE. If, at a point, an alert is not generated for something that is actually a port scan attack then it is proved that SPADE fails at some point and necessary modifications are made to make it identify port scan attacks given in any order or frequency.

For the identification of a port scanning trace that SPADE will fail to generate an alert for, researchers use Nmap, a port scanning tool. Generally, port scanning tools are used by hackers to see which of the ports in a network or a host is open to exploit the vulnerabilities.

It can also be used, from the network administrator's perspective to check the ports, whose services are not currently under use but are open so that those ports can be closed to avoid vulnerabilities. Nmap can be used to scan particular ports in a host, a range of ports or all of the ports.

Experimental setup: Three machines M1, M2 and M3 are connected via an ad hoc network. Nmap is used to generate port scan traces. Nmap is run on M1. Tcpdump is run on M2 listening to packets at its wlan0 interface and collects the packets in a tcpdump file (Fig. 2). The following port scan traces are pumped recursively from M2 to M3:

- To scan a single port on a particular host from a single IP
- To scan all or a range of the ports on a particular host from a single IP
- To scan a single port on a particular host from multiple IP addresses. Spoof addresses feature of Nmap is used for this purpose
- To scan a certain set of ports on a particular host from multiple addresses

Observations: The observed behavior of SPADE is as follows:

- Using port scan trace (1) SPADE generates alerts if the port is open. In M2, the ports open are 80 and 22. The alerts for those ports are considered as false positives
- When trace (2) is pumped recursively, SPADE alerts for a scan of certain set of ports in the target host from the same source port
- For trace (3) if the port to be scanned is an open port, reply packets are sent from the target back to the spoofed addresses and not to the original source. This is done to imitate that a port is being scanned from various clients. The spoof IPs used does not really exist and hence SPADE alerts are of the form "non-live destination used"

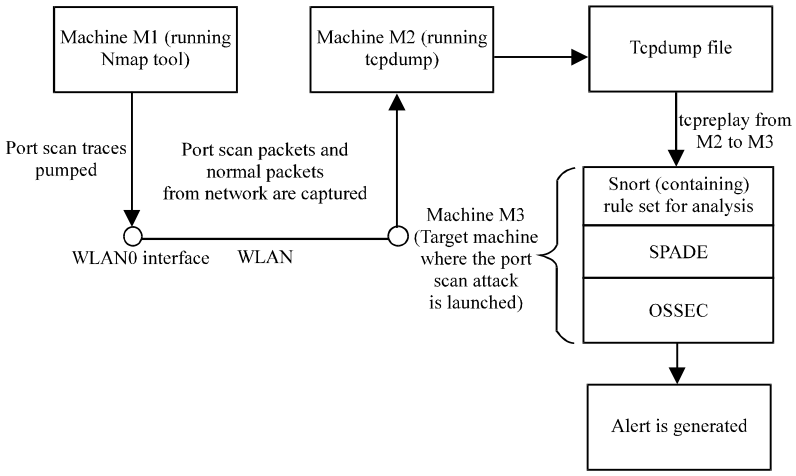

Fig. 2: Experimental setup

When SPADE is pumped separately with the three different traces, a change in the behavior of SPADE is observed with trace (2). A trace similar to trace (2) is generated that scans for 1000 ports on the target host. When an attempt to scan certain ports from the same IP address and the same port is made, SPADE alerts for the scan. Initially, the alerts are generated for the particular set of closed ports. The same type of port scan is performed various times with varying delays.

Every such trial are said to be an epoch. In every epoch, the speed with which the packets are pumped is varied. Initially, when packets are pumped with normal speed (baseline) using tcpreplay, true positive alerts for a specific set of ports on the target host are generated. In subsequent epochs, the speed with which the packets are pumped is increased. In the first epoch, the port scan trace is pumped to M3 using tcpreplay with normal speed. In the second epoch, the same port scan trace is pumped with 150% of the normal speed relative to tcpreplay. SPADE was observed to generate the same number of alerts and there were no false alarms (i.e., no false positives and no false negatives). In the next run, the same trace is replayed at 200% of the original speed relative to tcpreplay. The number of false positive alerts increased. Further, decreasing the speed of the packets in the next epoch, there were no alerts generated.

The observed inference from the above case is that when an attempt to scan a set of ports for open condition is made continuously from the same IP and Port, SPADE is able to detect it under normal conditions. But when the relative speed with which the port scan traces are sent is increased, SPADE produces false positives, i.e., it labels a non-malicious activity as malicious. Now the next step is to improvise SPADE in such a way that it will be able to identify all port scan traces pumped in any speed or frequency.

IMPROVING THE PERFORMANCE OF SPADE

The Threshold Adapting Method used in SPADE averages the anomaly score of all the packets over a certain period of observation. When the port scan traces are pumped fast, they are narrowly distributed across the time scale. The observation period is the number of minutes in a period of observation that is converted to the count of packets received. By default, its value is the average number of hours in a week. There is also another parameter in SPADE called the target rate that sets the number of alerts in a given observation period. If it is assigned a value >1 it corresponds to hourly alert rate. If it is assigned a value <1 then it corresponds to the fraction of considered packets to report based on the best estimate of the packet rate (Biles, 2003).

In a scenario where the port scan traces are pumped at a faster rate the distribution of anomalous packets is in a narrow scale across N observation periods. Hence, the average reporting threshold will be smaller compared to the case where the packets are pumped at a slower rate in which the anomalous packets are spaced in many of the N observation periods resulting in a larger average threshold value. If the observation period is too big for the packet rate, it leads SPADE to adjust its threshold to a smaller value because larger the period, smaller the average value.

On the other hand, when the observation period is small, it results in a larger average threshold leading to

Table 1: Variation of TCP replay speed and No. of alerts generated before modification of SPADE (target rate = 1.5 and observation period-default value (N))

Epoch	Packet relay speed (tcpreplay)	Packet rate	No. of alerts
1	Normal speed-baseline (speed at which no false alarm occurred)	24.36	16 (TP)
2	150% of baseline	36.54	16 (TP)
3	200% of baseline (2x times the normal speed)	48.72	21 (16 TP+5 FP)
4	Normal speed	24.36	No alerts

Table 2: Variation of TCP replay speed and No. of alerts generated after modification of SPADE (target rate and observation period adjusted)

Epoch	Packet relay speed (tcpreplay)	Packet rate	Target rate	Observation period	No. of alerts
1	Normal speed-baseline (speed at which no false alarm occurred)	24.36	1.5	Default (N)	16 (TP)
2	150% of baseline	36.54	1.5	Increase by 1.5N times	16 (TP)
3	200% of baseline (2x times the normal speed)	48.72	0.5	Increase by 2N times	16 (TP)
4	Normal speed	24.36	1.5	Default (N)	16 (TP)

many false alarms. The target rate is adjusted every hour based on the average of measurements over the observation period. Hence, a change in the value of observation period should ensure that the desired target rate is met. If the adjusted average threshold value is smaller than the target rate would not be met resulting in no alerts.

The target rate and observation period parameters are modified such that it balances the threshold on either extremes of the frequency. Hence, both the parameters are adjusted for various values to find out the optimum response of SPADE for a particular network.

RESULTS

Table 1 indicates the speed at which packets are pumped in each epoch and the response of SPADE by the number of alerts generated. This data corresponds to the rate before modification was done to SPADE. Packet rate is the No. of packets per sec pumped by tcpreplay to SPADE for analysis. Table 2 corresponds to the rate after modification was done to SPADE.

CONCLUSION

The major task in this research was finding a scenario in which false positives were generated by SPADE. It was found that SPADE failed in this regard when portscanning attack was attempted at a fast rate. This failure was overcome by modifying the parameters in SPADE according to the network environment. SPADE after modification is able to identify and generate alerts for portscan attacks in all scenarios. Thus, Snort and SPADE can be effectively deployed in Wireless Ad Hoc Networks.

REFERENCES

Anonymous, 2009. Intrusion detection systems. Information Assurance Tools Report, 6th Edn., September 25, 2009. http://iac.dtic.mil/iatac/download/intrusion_detection.pdf.

Biles, S., 2003. Detecting the unknown with Snort and Statistical Packet Anomaly Detection Engine (SPADE). Computer Security Online Ltd., Technical Report. http://webpages.cs.luc.edu/~pld/courses/447/sum08/class6/biles.spade.pdf.

Estevez-Tapiador, J.M., P. Garcia-Teodoro and J.E. Diaz-Verdejo, 2004. Anomaly detection methods in wired networks: A survey and taxonomy. Comput. Commun., 27: 1569-1584.

Gaikwad, V.S. and P.J. Kulkarni, 2012. One versus all classification in network intrusion detection using decision tree. Int. J. Sci. Res. Publ., 2: 1-5.

Garcia-Teodoro, P., J. Diaz-Verdejo, G. Macia-Fernandez and E. Vazquez, 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. Comput. Secur., 28: 18-28.

Kumar, S., 2007. Survey of current network intrusion detection techniques. http://www.cs.wustl.edu/~jain/cse571-07/ftp/ids.pdf.

Patcha, A. and J.M. Park, 2007. An overview of anomaly detection techniques: Existing solutions and latest technological trends. Comput. Networks, 51: 3448-3470.

Zhang, Y. and W. Lee, 2000. Intrusion detection in wireless ad-hoc networks. Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, August 6-11, 2000, Boston, MA., USA., pp: 275-283.