Gajendra Singh Chandel, Priyanka Murotia / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue4, July-August 2012, pp.1722-1726

Manet Threat Alarming Based On System Statistics & Support Vector Machine

Gajendra Singh Chandel [1] , Priyanka Murotia [2]
Department Of Information Technology , Sri Satya Sai Institute Of Science & Technology Sehore, Bhopal (M.P) , India [1,2]

Abstract

This paper presents a technique for network threat detection and alarming. The network security is one of the most important aspects while designing a Computer Network; the robust Security mechanism is required especially for the networks involving ecommerce & confidential data. In this paper we presents a statistical & classification approach to identify the attack. The proposed system takes system statistics like types of packets, delay, drop rate, buffer overflow etc. as input & returns the threat type (Blackhole, Wormhole, and Flooding etc.).

Keywords- Machine Learning, Network Security, Security Threats, SVM (Support Vector Machine).

1. INTRODUCTION

Network security starts with authenticating the user, commonly with a username and a password. Since this requires just one detail authenticating the user name i.e. the password, which is something the user 'knows' this is sometimes termed one-factor authentication. With two-factor authentication, something the user 'has' is also used (e.g. a security token or 'dongle', an ATM card, or a mobile phone); and with three-factor authentication, something the user 'is' is also used (e.g. a fingerprint or retinal scan). Once authenticated, a firewall enforces access policies such as what services are allowed to be accessed by the network users [2]. Though effective to prevent unauthorized access, this component may fail to check potentially harmful content such as computer worms or Trojans being transmitted over the network. Anti-virus software or intrusion prevention system (IPS)[3] help detect and inhibit the action of such malware. An anomaly-based intrusion detection system may also monitor the network and traffic for unexpected (i.e. suspicious) content or behavior and other anomalies to protect resources, e.g. from denial of service attacks or an employee accessing files at strange times. Individual events occurring on the network may be logged for audit purposes and for later high-level analysis. Communication between two hosts using a network may be encrypted to maintainprivacy. Honeypots (a honeypot is a trap set to detect, deflect, or in some manner counteract attempts at unauthorized use of information systems), essentially decoy network-accessible resources, may be deployed in a network As surveillance and early-warning tools, as the honeypots are not normally accessed for legitimate purposes. Techniques used by the attackers that attempt to compromise these decoy resources are studied during and after an attack to keep an eye on new exploitation techniques. Such analysis may be used to further tighten security of the actual network being protected by the honeypot [4].

2. BASICS OF THREAT DETECTION SYSTEM

A Threat detection system is a device or software application that monitors network or system activities for malicious activities or policy violations and produces reports to a Management Station [1]. Some systems may attempt to stop a Threat attempt but this is neither required nor expected of a monitoring system. Threat detection and prevention systems are primarily focused on identifying possible incidents, logging information about them, and reporting attempts. In addition, organizations use threat detection systems for other purposes, such as identifying problems with security policies, documenting existing threats, and deterring individuals from violating security policies. Threat detection systems have become a necessary addition to the security infrastructure of nearly every organization.

Threat detection systems typically record information related to observed events, notify security administrators of important observed events, and produce reports. Many Threat detection systems can also respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which involve the Threat detection systems stopping the attack itself, changing the security environment (e.g., reconfiguring a firewall), or changing the attack's content.

*2.*1. Threat Detection Techniques

2.1.1) Expert Systems*:* these work on a previously defined set of rules describing an attack. All security related events incorporated in an audit trail are translated in terms of if-then-else rules. Examples are Wisdom & Sense and Computer Watch (developed at AT&T).

2.1.2) Signature Analysis*:* Similarly to expert System approach, this method is based on the attack knowledge. They transform the semantic description

of an attack into the appropriate audit trail format. Thus, attack signatures can be found in logs or input data streams in a straightforward way. An attack scenario can be described, for example, as a sequence of audit events that a given attack generates or patterns of searchable data that are captured in the audit trail. This method uses abstract equivalents of audit trail data. Detection is accomplished by using common text string matching mechanisms. Typically, it is a very powerful technique and as such very often employed in commercial systems (for example Stalker, Real Secure, NetRanger, Emerald eXpert-BSM).

2.1.3) State-Transition Analysis*:* here, an attack is described with a set of goals and transitions that must be achieved by an intruder to compromise a system. Transitions are represented on state-transition diagrams.

2.1.4) Statistical Analysis Approach*:* this is a frequently used method (for example SECURENET). The user or system behavior (set of attributes) is measured by a number of variables over time. Examples of such variables are: user login, logout, number of files accessed in a period of time, usage of disk space, memory, CPU etc. The frequency of updating can vary from a few minutes to, for example, one month. The system stores mean values for each variable used for detecting exceeds that of a predefined threshold. Yet, this simple approach was unable to match a typical user behavior model. Approaches that relied on matching individual user profiles with aggregated group variables also failed to be efficient. Therefore, a more sophisticated model of user behavior has been developed using short- and long-term user profiles. These profiles are regularly updated to keep up with the changes in user behaviors. Statistical methods are often used in implementations of normal user behavior profile-based Intrusion Detection Systems.

2.1.5) Neural Networks*:* Neural networks use their learning algorithms to learn about the relationship between input and output vectors and to generalize them to extract new input/output relationships. With the neural network approach to intrusion detection, the main purpose is to learn the behavior of actors in the system (e.g., users, daemons). It is known that statistical methods partially equate neural networks. The advantage of using neural networks over statistics resides in having a simple way to express nonlinear relationships between variables, and in learning about relationships automatically. Experiments were carried out with neural network prediction of user behaviors. From the results it has been found that the behavior of UNIX super-users (roots) is predictable (because of very regular functioning of automatic system processes). With few exceptions, behavior of most other users is also predictable. Neural networks are still a computationally intensive technique, and are not widely used in the intrusion detection community.

2.1.6) User Intention Identification: This technique (that to our knowledge has only been used in the SECURENET project) models normal behavior of users by the set of high-level tasks they have to perform on the system (in relation to the users' functions). These tasks are taken as series of actions, which in turn are matched to the appropriate audit data. The analyzer keeps a set of tasks that are acceptable for each user. Whenever a mismatch is encountered, an alarm is produced.

2.1.7) Machine learning*:* This is an artificial intelligence technique that stores the user-input stream of commands in a vectorial form and is used as a reference of normal user behavior profile. Profiles are then grouped in a library of user commands having certain common characteristics [3].

2.1.8) Data Mining*:* generally refers to a set of techniques that use the process of extracting previously unknown but potentially useful data from large stores of data. Data mining method excels at processing large system logs (audit data). However they are less useful for stream analysis of network traffic. One of the fundamental data mining techniques used in intrusion detection is associated with decision trees [3]. Decision tree models allow one to detect anomalies in large databases. Another technique refers to segmentation, allowing extraction of patterns of unknown attacks [3]. This is done by matching patterns extracted from a simple audit set with those referred to warehoused unknown attacks [3]. A typical data mining technique is associated with finding association rules. It allows one to extract previously unknown knowledge on new attacks [3] or built on normal behavior patterns. Anomaly detection often generates false alarms. With data mining it is easy to correlate data related to alarms with mined audit data, thereby considerably reducing the rate of false alarms [3].

3. SVM (SUPPORT VECTOR MACHINE)

Support Vector Machines (SVMs) have developed from Statistical Learning Theory [9]. They have been widely applied to fields such as character, handwriting digit and text recognition, and more recently to satellite image classification. SVMs, like ANN and other nonparametric classifiers have a reputation for being robust. SVMs function by nonlinearly projecting the training data in the input space to a feature space of higher dimension by use of a kernel function. This results in a linearly separable dataset that can be separated by a linear classifier. This process enables the classification of datasets which are usually nonlinearly separable in the input space. The functions used to project the data from input space to feature space are called kernels (or kernel machines), examples of which include polynomial, Gaussian (more commonly referred to as radial basis functions) and quadratic functions. Each function has unique parameters which have to be

determined prior to classification and they are also usually determined through a cross validation process. A deeper mathematical treatise of SVMs can be found in [10].

By their nature SVMs are intrinsically binary classifiers however there exists two strategies one against one & one against all the first one gives much better performance hence it is selected in our proposal.

4. PROPOSED ALGORITHM

The proposed algorithm can be divided into two parts in first part the modeling & simulation of the different attacks are performed this part provides the training data set for the second part where this data is used to train the SVM which is later used for threat detection & classification.

4.1 Modeling, Simulation & Analysis of Attacks

After gathering the required details of all types (1.blackhole, 2.wormhole, 3.selfish, 4.sleep, 5.flooding, 6.replay, & 7.normal condition) of attack each type of attack is simulated in OPNET network simulator with following simulation parameters.

TABLE 1
SIMULATION PARAMETERS

Parameter Name	Value
Number of Nodes	20
Simulation Time	60 minutes
Area	1x1 Km.
Node Speed	10 Km/h
Packet Size	1024 bits
Routing	AODV
Transmitter Power	5 mW
Antenna Type	Omni-directional

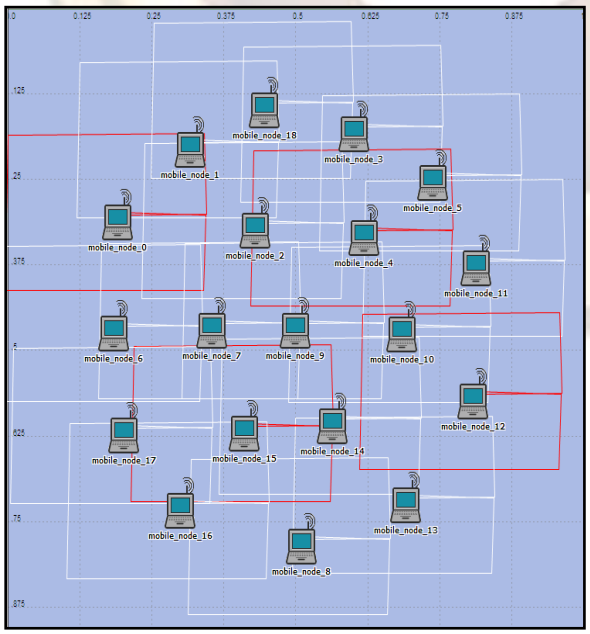

Figure 1: Snap shot of the simulated network.

4.2 **Parameters Description Chosen for Statistical Analysis.**

1) Number of Hops per Route*:* This statistic represents the number of hops in each route to every destination in the route table of all nodes in the network.

2) Route Discovery Time: The time to discover a route to a specific destination is the time when a route request was sent out to discover a route to that destination until the time a route reply is received with a route to that destination. This statistic represents the time to discover a route to a specific destination by all nodes in the network.

3) Routing Traffic Received*:* Amount of routing traffic received in packets/sec in the entire network.

4) Routing Traffic Sent: Amount of routing traffic sent in packets/sec in the entire network.

5) Total Cached Reply Sent: When a node receives a route request and is not the target of the route request, it looks up its route table to determine if it has any route to the target of the route request. If so; the node sends back a "Cached Route Reply" and does not re-broadcast the request packet. This statistic represents the total number of cached route replies sent by all nodes in the network.

6) Total Packet Dropped: When no route is found to the destination, the node drops the packets queued to the destination.

7) Total Replies Sent From Destination: Once the destination node receives a route request, it sends a route reply to the source of the request. This statistic represents the total number of route reply packets sent from all nodes in the network if they are destinations of route requests. This statistic represents the total number of application packets discarded by all nodes in the network.

8) Total Route Error Sent: A node may send Hello messages to its neighbor to confirm next hop reachability. If next hop reachability cannot be confirmed, the node sends back a route error message to all nodes that use that next hop to reach various destinations. This statistic represents the total number of route error packets sent by all nodes in the network.

9) Total Route Replies Sent: A node would send back a route reply to the source of the request; if a) It was the destination of the request b) It had a route to the destination in its route table.

This statistic represents the total number of route reply packets sent by all nodes in the network (both cached route replies and route replies if it's a destination).

10) Total Route Requests Sent: This statistic represents the total number of route request packets sent by all nodes in the network during route discovery.

4.3 Simulation Results from This Part of Algorithm

TABLE 2:
COLLECTED STATISTICS OF DIFFERENT PARAMETERS FOR BLACK-HOLE ATTACK

	1	2	3	4	5	6	7	8	9	10
5	1.01	0.083	182	10	127	1	17	6	141	20
10	1.01	0.083	230	13	127	1	17	6	141	20
15	1.01	0.048	242	14	68	1	10.8	5	78	12
20	1	0.033	253	14.2	47	1.23	8.9	5.3	44	9.8
25	1.02	0.076	260	14.4	36	2.2	7.8	4.5	39	9.9
30	1.01	0.066	262	14.8	30	2.41	6.4	3.8	25	7.1
35	1.01	0.066	263	14.9	25	2.41	5.9	3.4	21	6.3
40	1	0.049	264	15	20	2.6	5.4	3.2	19	5.9
45	1	0.042	265	15	21	3.02	5.7	3.3	20.1	6
50	1	0.038	266	15.1	19	2.7	5	3.1	18	5.2
55	1	0.037	268	15.2	20	2.72	4.8	3.6	18.1	5.3
60	1	0.033	270	15.2	18	2.81	5.1	3.3	18	5.8

The table 2 shows the network statistics for the Black-hole attack. In the table each row shows the collected parameters after certain time of simulation (like 5, 10 and 15 minutes etc.) like above table we simulated the network for Black-hole, wormhole selfish, sleep, flooding, replay attacks & for normal condition.

4.4 Training & Validation

In this stage the collected data from previous stage is classified into different group based on the attack type, then each data set is normalized by detecting its maximum and minimum values by the following formula

$$V_{norm} = \frac{V - V_{min}}{V_{max} - V_{min}}$$

The normalized values set are arranged in an array to represent system condition by a vector this vector can be represented by

$Trn_{vect} = [V_{norm\,1}, V_{norm\,2}, V_{norm\,3}, \ldots . V_{normn}]$

Hence the system states can be treated as n dimension vector.

Now these vectors with their classification group are used to train the SVM (Support Vector Machine) in one against one method.

After that for detecting the system status at any time the system data is collected and converted in to the vector as stated above and then it is applied to the trained SVM which classifies the network condition to one of the most matched trained conditions.

4.5 Simulation Results for This Part of Algorithm

The simulation of the proposed algorithm is performed using MATLAB 7.5 on IBM P4 PC with windows XP operating system.

TABLE 3
SHOWING THE PREDICTED RESULTS BY THE PROPOSED ALGORITHM.

Training Data	Attack Type	TPR	TNR	FPR	FNR	Accu--racy
10	1	1	1	0	0	1
10	2	1	0.9	0.1	0	0.95
10	3	0.93	0.93	0.06	0.06	0.93
10	4	0.91	0.91	0.08	0.08	0.91
10	5	1	0.90	0.09	0	0.95
10	6	1	0.88	0.11	0	0.94
10	7	0.91	0.91	0.08	0.08	0.91

Attack types for table 3: 1.Blackhole, 2.Wormhole, 3.Selfish, 4.Sleep, 5.Flooding, 6.Replay, 7.Normal Condition.

5. CONCLUSION

The Network threat detector presented in this paper is capable of generating alert as well as classifying the individual attacks. The Detection accuracy of the system is up to 95% which is excellent also the algorithm have very low FPR (max 11%) hence reduces the chances of false alarming, further it could achieve much better performance for only binary detections like attack and non attack conditions in future it can also be modified with different classification techniques to get much better results.

REFERENCES

[1] Simmonds, A; Sandilands, P; van Ekert, L (2004). "An Ontology for Network Security Attacks". Lecture Notes in Computer Science. Lecture Notes in Computer Science 3285: 317–323.

[2] P. Garcı´a-Teodoroa, J. Dı´az-Verdejo, G. Macia´-Ferna´ndeza, E. Va´zquezb "Anomaly-based network intrusion detection Techniques, systems and challenges " published on computers & security 28 (2009) 18 – 28".

[3] http://www.windowsecurity.com/articles/ids-part2-classification-methods- chniques.html

[4] Axelsson S.: Intrusion Detection Systems: A Taxomomy and Survey. Technical Report No 99-15, Dept. of Computer Engineering, Chalmers UniversityofTechnology,Sweden,March2000,http://www.ce.chalmers.se/staff/sax/taxonomy.ps

[5] Bass T.: Intrusion Detection Systems Multisensor Data Fusion: Creating

Cyberspace Situational Awareness. Communication of the ACM, Vol.43,Number1,January2000,pp.99105,http ://www.silkroad.com/papers/acm.fusion.ids. ps.

[6] Debar H., Dacier M., Wespi A.: Towards a taxonomy of intrusion-detection systems. Computer Networks, 31, 1999, pp. 805-822.

[7] Dorosz P., Kazienko P.: Omijanie intrusion detection systems. Software 2.0 no 9 (93), September 2002, pages 48-54. (In Polish only)

[8] Dorosz P., Kazienko P. Systems wykrywania intruzów. VI Krajowa Konferencja Zastosowan Kryptografii ENIGMA 2002, Warsaw 14-17 May2002,p.TIV4778,(InPolisonly)http://ww w.enigma.com.pl/konferencje/vi_kkzk/index .htm

[9] V. N. Vapnik, The Nature of Statistical Learning Theory. New York: Springer-Verlag, 1995.

[10] V. Kecman, Learning and Soft Computing: Support Vector Machines, Neutral Networks and Fuzzy Logic Models. Cambridge, MA: MIT Press, 2001.