2010 Second International Conference on Communication Software and Networks

Hybrid Framework for Behavioral Prediction of Network Attack Using Honeypot and Dynamic Rule Creation with Different Context for Dynamic Blacklisting

Renuka Prasad B,
Research Scholar,Dr. M.G.R Deemed University, Lecturer, RV College of Engineering, Bangalore, Karnataka
E-mail: renukaprasadb@gmail.com

Annamma Abraham,
Professor,Department of Mathematics,
RV College of Engineering
Bangalore, Karnataka
E-mail:annamma65@gmail.com

Abstract--*Honeypots are decoys designed to trap, delay, and gather information about attackers. All the previous work in the field was related to majorly intrusion detection system, but in this research work, the highlight is more focused on the novel approach of creation of a Honeypot schema which is powered by intelligence along with the design of classifier. The output generated by the classifier generates a dynamic list of attacks, which are then queued in the proposed Honeypot architecture built with neural network to understand various approach of behavior and patterns of the attacker. The network administrator collects all such relevant information over the network itself allowing the inbound network connection from the attacker to do so and the system creates a hybrid framework to prevent the probability of vulnerable and hostile situation over the network even before the attack event is performed by the attacker.*

Keywords: Honeypot, Honeynet, Intrusion Detection System, IP Blacklisting.

I. INTRODUCTION

Network administrators usually use a firewall and an intrusion detection system (IDS) to protect their network. The firewall can control the inbound and outbound traffic according to the type of service requested, the user name, and the IP address of packets. The IDS can be deployed between the local area network and the Internet or any other important gateway for detecting suspicious packets. Moreover, the IDS system that uses anomaly detection has a high false-positive ratio. The use of a Honeypot can overcome the inherent deficiencies of the IDS and firewall. More importantly, one can treat it as a platform for security education in a university. If a honeypot is deployed in front of a firewall, it can be treated as an early-warning system. If we deploy it behind the firewall, it can serve as part of a defense-in-depth system and can be used to detect attackers who bypass the firewall and IDS or threats from insiders

II. DEFINITION

The Honeypot expert Lance Spitzner[1] gives an authoritative definition to honeypot: "Honeypot is a security resource whose value lies in being probed, attacked or compromised". In contrast to firewall, IDS and other security technologies, honeypot has many advantages, such as, easily deployed, high facticity of the capture data, low miss alarm and false alarm, lowly price of the deployment and it can detect unknown attack and new worms. Honeypot is classified as low–interaction honeypot, mid-interaction honeypot and high-interaction honeypot based on the level of its interaction to attackers. The level of interaction reflects the freedom of hackers to engage attack activities in the honeypot, the more lifelike of the operating system and network services, the higher of the interaction, the more information we can get, the stronger attraction to hackers, but its deployment and maintenance is more complex and the risk is higher. Now, the development of honeypot stays at the phase of honeynet. The research focus is to construct honeypot with real host, operating system and application program, using the tools of data capture, data analysis and data control to improve its effectiveness. Together with firewall, IDS and anti-worm software, honeypots form into a honeynet defense system (See Figure 1) to ensure network security.

[1] Lance Spitzner is a senior security architect for SUN Microsystem, Inc, and an acknowledged authority in security and Honeypot research. He is a founder of Honeypot Project, a nonprofit group of 30 security professionals. Lance has presented data on Honeypot technologies to Pentagon, an FBI Academy.

978-0-7695-3961-4/10 $26.00 © 2010 IEEE
DOI 10.1109/ICCSN.2010.82

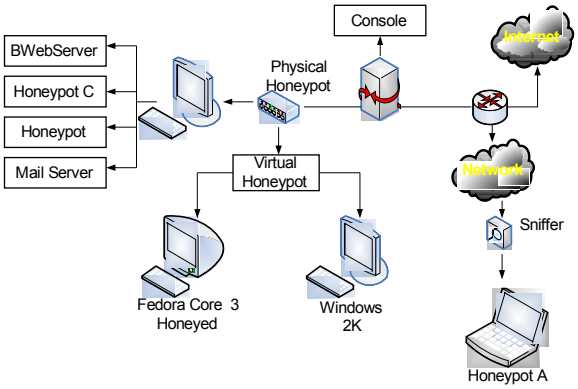

Figure 1. IDS Architecture

III. PREVIOUS WORK

Previous research has focused on the economics of vulnerabilities. Eric Rescorla developed a model showing that most exploits are due to disclosed vulnerabilities[1]. Therefore, he recommended against advertising existing vulnerabilities. Ashish Arora and his colleagues introduced a model that found that neither immediately advertising a vulnerability nor suppressing it were optimal. [2] Steve Beattie and his colleagues focused on the delay before applying a security patch, recommending a delay between 10 and 30 days. [3] Huseyin Cavusoglu and his colleagues introduced an economic model for improved security-patch management using liability and cost sharing as incentives for vendors to more efficiently provide software patches.[4] A few articles have included empirical studies of the vulnerability remediation process. In another work,[5] Arora and his colleagues used empirical evidence to support their previous model.[2] They used two types of data: malicious activity collected from 14 honeypots with various configurations and vulnerability data from the Common Vulnerabilities and Exposures (CVE) ICAT database. Arora and his colleagues classified the vulnerabilities as secret, published, or patched. The authors collected data for nine weeks between November 2002 and December 2003 and then analyzed them to discover the impact of the vulnerability class on the attack frequency. William Arbaugh and his colleagues also described a study that used data collected by the Computer Emergency Response Team (CERT, www.cert.org) and proposed a vulnerability discovery life cycle.[6] According to their proposed life-cycle model, attackers are more interested in exploiting the latest vulnerabilities.

IV. OBJECTIVE OF PROJECT

The major objective of this research work is to propose a hybrid framework to mitigate the attacking strategy of the intruders by creating a classifier based on artificial neural network deploying Honeypot and to create a dynamic security policies to validate and prevent the inbound attacks with different context for dynamic blacklisting of the IP address of the attacker hence after.

V. PROPOSED SYSTEM

In the proposed system, the organizational network is assumed to be vulnerable, where the main intention behind the research work is to understand the methodology of the attacking strategy and then build a system which could formulate the dynamic listings of the attackers and induce blacklisting of the intruder's resources forever. The Proposed system's main purpose is to identify parameters which include general security policies e.g. firewall rules, signatures, filters and vulnerabilities of most popular applications at client side and some of the relations between hosts / IP and DNS which finally is fed to a classifier to classify the abnormal behavior or unusualness events or most common harmful events with an intelligence, instead of algorithm, the classified can be made more dynamic for listing the intrusions. The attacking strategies are enumerated and dynamically listed in order to analyze the profile of attacker behavior. With the increasing popularity of web services, attackers are getting attracted to hack the services and the servers on which they run. One of the major threats is that of intruders which may maliciously try to access the data or services. Thus there is a need to protect the servers on which web services are running from intruders. Therefore, Intrusion Detection systems need to be employed. Honeypot collects very little data and what it collects is normally of high value. This information can be used in extraction of intrusion detection signature.

The main goals of the research work are:

- To create a module through which the various pattern of the intrusion and rogue application can be trapped.
- To create a neural classifier, this will initiate a continuous learning process for trapping the network behavior under vulnerable situation.
- To create a module, where the network administrator can have the complete analysis of behavior of the probable attacker.

- To divert the attention of the attacker from the real network, in a way that the main information resources are not compromised
- To build attacker profiles in order to identify their preferred attack methods, similar to criminal profiles used by law enforcement agencies in order to identify a criminal's *modus operandi*
- To provide a better performance than the traditional approach characterized by more scalable and efficient reducing the overheads.
- To identify new vulnerabilities and risks of various operating systems, environments and programs which are not thoroughly identified at the moment

VI. SYSTEM ANALYSIS

One major challenge in current intrusion detection mechanism is that how to identify the camouflaged intrusion more accurately from a huge amount of alerts. Some research work has shown that most of the damages result from vulnerabilities existing on the network and hosts. It is therefore demanding to apply vulnerability analysis in order to improve IDS for better network security. It is considered as a surveillance and early-warning tool. A honeypot can take on other forms, such as files or data records, or even unused IP address space. A honeypot that masquerade as an open proxy in order to monitor and record the activities of those using the system is called a sugarcane. Honeypot should have no production value and hence should not see any legitimate traffic or activity. Whatever they capture can then be surmised as malicious or unauthorized. Honeypot can carry risks to a network, and must be handled with care. If they are not properly walled off, an attacker can use them to break into a system.

The consecutive approach is to initiate the classifier output which are probable attacks / suspicious events / behavior and feed those classified list of attacks to the Honeypot that is based on neural network which has a learning module through which some intelligence is obtained about the attacks and finally some test data is fed to the system which will just say whether the given test data is an attack or not and attack based on the intelligence obtained while learning. During this process a dynamic rule creation mechanism will be built instead of creating alarms and having one more step of inspection by the administrator where they have to decide on that whether it was an attack or not . In the dynamic rule creation mechanism very easily a suspicious intruder can be detected and as well as an intrusions could be detected and based on the behavior and context blacklisting of the resource /host/IP /network can be done without much overheads.

The algorithms can be used to evolve simple rules for network traffic. These rules are used to differentiate normal network connections from anomalous connections. These anomalous connections refer to events with probability of intrusions. The rules stored in the rule base are usually in the following form:

if { condition }
then { act }

For the problems presented above, the *condition* usually refers to a match between current network connection and the rules in IDS, such as source and destination IP addresses and port numbers (used in TCP/IP network protocols), duration of the connection, protocol used, etc., indicating the probability of an intrusion. The *act* field usually refers to an action defined by the security policies within an organization, such as reporting an alert to the system administrator, stopping the connection, logging a message into system audit files, or all of the above. For example, a rule can be defined as:

if {
the connection has following information:
source IP address 124.12.5.18;
destination IP address:130.18.206.55;
destination port number: 21;
connection time: 10.1 seconds }
then
{stop the connection}

This rule can be explained as follows: if there exists a network connection request with the source IP address 124.12.5.18, destination IP address 130.18.206.55, destination port number 21, and connection time 10.1 seconds, then stop this connection establishment. This is because the IP address 124.12.5.18 is recognized by the IDS as one of the blacklisted IP addresses; therefore, any service request initiated from it is rejected. In this implementation, the network traffic used for this algorithm is a pre-classified data set that differentiates normal network connections from anomalous ones. This data set is gathered using classifier. The data set is manually classified based on experts' knowledge. It is used for the fitness evaluation during the execution of the prototype. By starting the process with only a small set of randomly

generated rules, we can generate a larger data set that contains rules for IDS. These rules are "good enough" solutions for IDS and can be used for filtering new network traffic.
In order to fully exploit the suspicious level, there is a need to examine all fields related with a specific network connection. For simplicity, it is considered some obvious attributes for each connection. The definition of rules (for TCP/IP protocols) is shown in Table 1.

The corresponding rule for the "Example Value" attribute in Table 1 could be translated as:

if {
the connection has following information:
source IP address 209.11.??.??;
destination IP address:130.18.176+?.??;
source port number: 42335;
destination port number: 80;
connection time: 482 seconds;
the connection is stopped by the originator;
the protocol used is TCP;
the originator sent 7320 bytes of data; and
the responder sent 38891 bytes of data
}
then
{stop the connection}

VII. NEURON STRUCTURE

Altogether there are fifty-seven genes in each chromosome. For simplicity, we use hexadecimal representations for the IP addresses. The rule can be explained as follows: if a network connection with source IP address 209.11.??.?? (209.11.0.0 ~ 209.11.255.255), destination IP address 130.18.176.?? (130.18.176.0 ~ 130.18.255.255), source port number 42335, destination port number 80, duration time 482 seconds, ends with state 11 (the connection terminated by the originator), uses protocol type 2 (TCP), and the originator sends 7320 bytes of data, the responders sends 38891 bytes of data, then this is a suspicious behavior and can be identified as a potential intrusion. The actual validity of this rule will be examined by matching the historical data set comprised of connections marked as either anomalous or normal. If the rule is able to find an anomalous behavior, a bonus will be given to the current chromosome. If the rule matches a normal connection, a penalty will be applied to the chromosome. Clearly no single rule can be used to separate all anomalous connections from normal connections. The population needs evolving to find the optimal rule set.

It is useful when representing a network block (a range of IP addresses or port numbers) in a rule. Once the spatial information is included in the rules, the capability of the IDS can be greatly improved as an intrusion may initiate from many different locations. The inclusion of the duration time of a network connection in the chromosome ensures incorporation of temporal information for network connections. The maximum value of duration time is 99999999 seconds, which is more than a year. This is helpful for identifying intrusions because complex intrusions may span hours, days, or even months.

VIII. OTHER PARAMETERS

There are also other parameters that need to be considered, such as mutation rate, crossover rate, number of populations, and number of generations. These parameters should be adjusted according to the application environment of the system and the organization's security policy.

Victim hosts are an active network counter-intrusion tool. These computers run special software, designed to appear to an intruder as being important and worth looking into. In reality, these programs are dummies, and their patterns are constructed specifically to foster interest in attackers. The software installed on, and run by, victim hosts is dual purpose. First, these dummy programs keep a network intruder occupied looking for valuable information where none exists, effectively convincing him or her to isolate themselves in what is truly an unimportant part of the network. This decoy strategy is designed to keep an intruder from getting bored and heading into truly security-critical systems. The second part of the victim host strategy is intelligence gathering. Once an intruder has broken into the victim host, the machine or a network administrator can examine the intrusion methods used by the intruder. This intelligence can be used to build specific countermeasures to intrusion techniques, making truly important systems on the network less vulnerable to intrusion. The overview of the architecture of the project is as given below in figure 2:

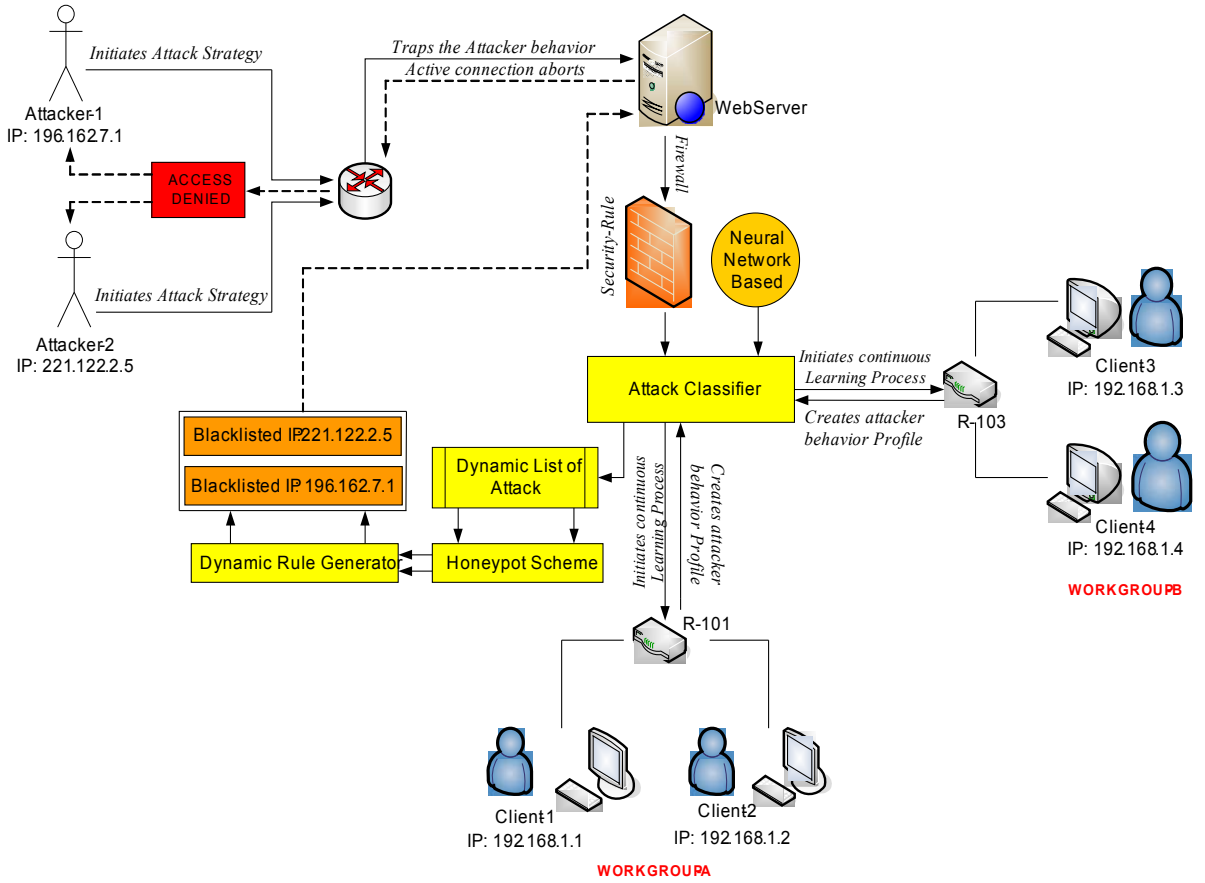

Figure 2. Overview of the proposed System

A. TECHNICAL REQUIREMENT SPECIFICATION

The basic hardware Requirement specification of the proposed system is minimum Intel Pentium IV Processor, 2 GB Hardisk, 40 GB HDD,CD-ROM, LAN/ Internet Connection to Server Machine, TCP/IP network for communication between clients and server.

The software requirement specification includes the operating System with Windows XP with SP2, Programming Tool to be used is J2EE, and IDE deployable is either NetBeans or MyEclipse

B. EXPERIMENTAL SETUP

The experimental test bed consist of conducting an empirical analysis to determine which service to determine which service vulnerabilities attackers tend to target to prioritize the vulnerability analysis process. More precisely, we will attempt to quantify the link between vulnerabilities and malicious connections by empirically studying the malicious traffic received over a six-month duration on ten target computers with running important data is connected with server, running Microsoft Windows service packs. Our work here should allow other research teams to replicate the experiment on their organization's network over additional time periods depending upon the results of the experiment.

C. RESULT ANALYSIS:

The client side attack is always the high in number than the server side attack. So in the experimental test bed of 10 systems, we will intentionally create an intrusion in those 10 system, as we will have the complete knowledge about the network assuming it to be intruded, then from a hostile system, we will attempt to run the intrusion program, which will activate the event of the application to trace and analyse the specific pattern of hostile situation generated from that particular system. So, various information like IP address, MAC Address, port number with various hardware information will be trapped by Honeypot mechanism and all the trapped

information from the attacker will be analysed about the degree of threats level. Based on the degree of threat level, our proposed application will not only fail the attack intention but also will blacklist the resource profile of the attacker. We can conduct here empirical analyses to attempt to understand the relationship between the number of malicious connections • and the total number of vulnerabilities,

Malicious connections and the number of vulnerabilities for different services, and known successful attacks and the number of vulnerabilities for different services. To quantify the relationship between the amount of malicious traffic and the number of vulnerabilities, we calculated correlation coefficients. We then conducted a more detailed analysis to understand the cause of the highly correlated events. The assumption that a normal distribution can be associated with the amount of malicious traffic and the number of vulnerabilities might be incorrect. More precisely, we don't make any assumption about the distribution of the amount of malicious traffic and the number of vulnerabilities.

IX. SCOPE OF PROJECT

One of the exclusive scope of the proposed system is the ability to track and capture the various patterns of the threatening situation by the system which makes sure that the organizational network is 100% safe from any probability of attacks from the specific attacker, thereby assuring the best of the intrusion prevention techniques.

X. CONCLUSION

In this research work we have proposed a system where the network administrator will observe and analyse various types of attacking tendencies originating from variable source in network. The process basically understand the pattern and behavior of the hostile circumstances over the network and then it creates the profiles of the attackers based on this pattern analysis, which will protect the network system of the organization by blacklisting the origination of the resource profiling over the network itself thereby assuring the organizational network to be the most secure one in any future probability of network threats from those attackers.

Reference

[1] E. Rescorla, "Is Finding Security Holes a Good Idea?" IEEE Security & Privacy, vol. 3, no. 1, 2005, pp. 14–19.

[2] A. Arora, R. Telang, and H. Xu, "Optimal Policy for Software Vulnerability Disclosure," Proc. 3rd Workshop Economics of Information Security (WEIS 04), 2004; www.dtc.umn.edu/weis2004/xu.pdf.

[3] S. Beattie et al., "Timing the Application of Security Patches for Optimal Uptime," Proc. Usenix 16th Systems Administration Conf. (LISA 02), Usenix Assoc., 2002, pp. 233–242.

[4] H. Cavusoglu, H. Cavusoglu, and J. Zhang, "Economics of Security Patch Management," Proc. 5th Workshop Economics of Information Security (WEIS 06), 2006; http://weis2006.econinfosec.org/docs/5.pdf.

[5] A. Arora et al., "Impact of Vulnerability Disclosure and Patch Availability: An Empirical Analysis," Proc. 3rd Workshop Economics of Information Security (WEIS 04), 2004; www.dtc.umn.edu/weis2004/telang.pdf.

[6] W.A. Arbaugh, W.L. Fithen, and J. McHugh, "Windows of Vulnerability: A Case Study Analysis," Computer, vol. 33, no. 12, 2000, pp. 52–59.

[7] Characterization of Attackers' Activities in Honeypot Traffic Using Principal Component Analysis, S. Almotairi, A. Clark, G. Mohay, and J. Zimmermann Information Security Institute, Queensland University of Technology Brisbane, Queensland, Australia, IEEE 2008

[8] Cooperation of Intelligent Honeypots to Detect Unknown Malicious Codes, Jungsuk SONG, Hiroki TAKAKURA, IEEE 2008

[9] False Positives Reduction via Intrusion Alert Quality Framework, Najwa Abu Bakar, Bahari Belaton, Azman Samsudin, School of Computer Science, University Science Malaysia, IEEE 2005