INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

MITIGATION FRAMEWORK FOR ATTACKS

Devyani Jadhav[1], Shital Moture[2], Sunetra Moholkar[3], Vinay Mutyal[4]

[1-4]Computer Engineering, MIT College of Engineering, Pune, Maharashtra, India. Email: devyanitajadhav@gmail.com[1], smoture@gmail.com[2], sunetranm@gmail.com[3], vinay9133@gmail.com[4]

Abstract: - Today, increase in the use of network resources is followed by a rising volume of security problems. New threats and vulnerabilities are discovered every day and affect users and companies at critical levels, from privacy issues to financial losses. Monitoring network activity is a mandatory step for researchers and security analysts to understand these threats and to build better protections. Honeypots are served as early warning, it is an information system resource used to divert attackers and hackers away from critical resources as well as a tool to study an attacker's methods.

This study addresses the challenge of improving the scalability and flexibility of honeypots. For a better integration into the organization network, this architecture was combined with Behavior analysis engine. Since this system is written in Java, it can potentially run on various platforms, windows or UNIX, workstations or handheld devices. It has a rule-based decision engine, whose design is guided by the analysis of real world attack data. The attack data was collected by opening up our honey pot to solicit possible attacks.

This study marks a major step toward leveraging honeypots into a powerful security solution. The contributions of this study will enable security analysts and network operators to make a precise assessment of the malicious activity targeting their network

Index Terms: Privacy, honeypot technology, data security, client side attack

1. INTRODUCTION

Business of the most of organizations is greatly affected by the internet today as business of most part of the world is shifted on internet. Global communication is getting more important every day. At the same time, computer crimes are increasing. To address this concern, network operators and security researchers have developed and deployed a variety of solutions. The goal of these solutions is two-fold: first to monitor, and second to protect network assets. Monitoring allows researchers to understand the different threats. Data are being collected to better characterize and quantify malicious activity. While addressing this solution it is observed that most of these measures are based on known facts, known attack patterns. As in the military, it is important to know, who your enemy is, what kind of strategy he uses, what tools he utilizes and what he is aiming for. Gathering this kind of information is not easy but important. By knowing attack strategies, countermeasures can be improved and vulnerabilities can be fixed. Generally, such information gathering should be done silently, without alarming an attacker. All the gathered information leads to an advantage on the defending side and can therefore be used on productive systems to prevent attacks.

Even if , in the organization network, the firewall, IDS/IPS are being placed for the security of the organizational network, but the protection given by these security products are limited as all these devices are relying on the signature based detection techniques, therefore no network can be hundred percent secure by putting these security devices. The firewall provides security by allowing only specific services through it. The firewall implements a policy for allowing or disallowing connections based on organizational security policy and business needs. The firewall also protects the organization from malicious attack from the Internet by dropping connections from unknown sources.

Honeypots plays the significant roles in terms of detection of known and unknown attacks spreading in the network. In network deception, attackers are usually intentionally presented with host(s) on the network that has one or more vulnerabilities. Honeypots are network deception tools that are capable of presenting such illusion. A popular low- interaction honeypot is the Honeyd program by Niels Provos, which was originally written in C and built for UNIX platforms. It does not have monitoring and intrusion detection capabilities. In our research, we developed a Java base Honeypot to address some of the above deficiencies. After an initial prototype of our Honeypot became available, we ran the program on a Linux machine in a remote DSL location. We configured the program to open many vulnerable ports so as to solicit attacks. Incoming packets were captured and saved into a tcpdump log file. The tcp dump data containing real attack attempts provide a valuable resource for us in the design of an decision engine for the system. We observed that the tcpdump data collected through Honeypot program only contain limited information about network connections, since possible network intruders are turned away when they are presented with false system information. In general, the tcpdump data we collected only have connection setup and/or termination information for TCP traffic and port information for UDP traffic. The challenge, therefore, is to design a simple and efficient decision engine that can catch potential attacks with only limited amount of information available.

The objective of our research is not to develop a full-fledged Intrusion Detection System (IDS), but rather to identify and enhance an intrusion detection scheme that works effectively with limited amount of connection information. First, we manually developed new rules fBusiness of the most of organizations is greatly affected by the internet today as business of most part of the world is shifted on internet. Global communication is getting more important every day. At the same time, computer crimes are increasing. To address this concern, network operators and security researchers have developed and deployed a variety of solutions. The goal of these solutions is two-fold: first to monitor, and second to protect network assets. Monitoring allows researchers to understand the different threats. Data are being collected to better characterize and quantify malicious activity. While addressing this solution it is observed that most of these measures are based on known facts, known attack patterns. As in the military, it is important to know, who your enemy is, what kind of strategy he uses, what tools he utilizes and what he is aiming for. Gathering this kind of information is not easy but important. By knowing attack strategies, countermeasures can be improved and vulnerabilities can be fixed. Generally, such information gathering should be done silently, without alarming an attacker. All the gathered information leads to an advantage on the defending side and can therefore be used on productive systems to prevent attacks.

Even if , in the organization network, the firewall, IDS/IPS are being placed for the security of the organizational network, but the protection given by these security products are limited as all these devices are relying on the signature based detection techniques, therefore no network can be hundred percent secure by putting these security devices. The firewall provides security by allowing only specific services through it. The firewall implements a policy for allowing or disallowing connections
based on organizational security policy and business needs. The firewall also protects the organization from malicious attack from the Internet by dropping connections from unknown sources.

Honeypots plays the significant roles in terms of detection of known and unknown attacks spreading in the network. In network deception, attackers are usually intentionally presented with host(s) on the network that has one or more vulnerabilities. Honeypots are network deception tools that are capable of presenting such illusion. A popular low- interaction honeypot is the Honeyd program by Niels Provos, which was originally written in C and built for UNIX platforms. It does not have monitoring and intrusion detection capabilities. In our research, we developed a Java base Honeypot to address some of the above deficiencies. After an initial prototype of our Honeypot became available, we ran the program on a Linux machine in a remote DSL location. We configured the program to open many vulnerable ports so as to solicit attacks. Incoming packets were captured and saved into a tcpdump log file. The tcpdump data containing real attack attempts provide a valuable resource for us in the design of an decision engine for the system. We observed that the tcpdump data collected through Honeypot program only contain limited information about network connections, since possible network intruders are turned away when they are presented with false system information. In general, the tcpdump data we collected only have connection setup and/or termination information for TCP traffic and port information for UDP traffic. The challenge, therefore, is to design a simple and efficient decision engine that can catch potential attacks with only limited amount of information available.

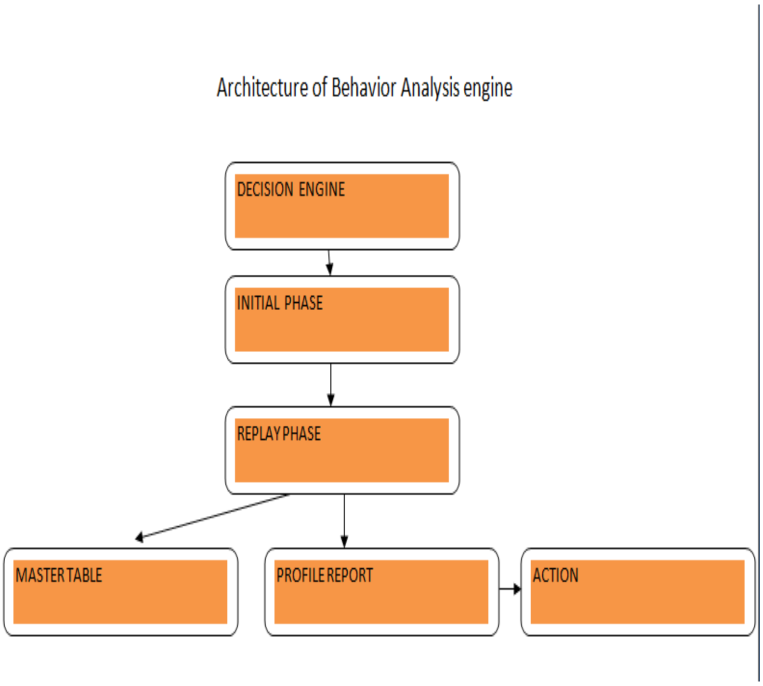
The objective of our research is not to develop a full-fledged Intrusion Detection System (IDS), but rather to identify and enhance an intrusion detection scheme that works effectively with limited amount of connection information. First, we manually developed new rules fBusiness of the most of organizations is greatly affected by the internet today as business of most part of the world is shifted on internet. Global communication is getting more important every day. At the same time, computer crimes are or the alerts found in our analysis .or the alerts found in our analysis. The new rules, along with the existing rule database, can handle most of the cases for intrusion detection. Second, a analyzer was written to classify the generated alerts for a better user interface, which contains the percentage of various types of attacks and also presents frequency of ports being scanned

To detect and correspondingly to develop the defense mechanism, there is a need to study the behavior of the malicious servers in terms of how the attacks are spreading and how they are exploiting the client side applications to target the innocent users.

2. THE PROPOSED ARCHITECTURE

In this proposed architecture system captures all packets coming to its server it also hosts information of each packet captured and store it in database of referencing it in future. This stored data is used by behavior analyzer for packet against rules. The following figure shows overall architecture of the proposed system.

Fig 1: Proposed Architecture



There are three major parts of this system –

i) Data Capture – contains the traffic logging components such as Honeypot and Tcpdump for collecting data.

ii) Data Analysis – is used for analysis and extraction which contains data analysis part of pattern extraction mechanism for extracting precise attack pattern.

iii) Pattern Extraction – is the one used for extracting good quality attack signatures.

2.1 Data Capture

Data Capture makes log of all the activities made by attacker. Honeypot also works this way. Honeypot log and network traffic log from Tcpdump are the two sources for log capture. The framework supports several ways of logging network activity. It also creates connection logs that can report connections attempted and created for all protocols. For analyzing the complete attack scenario, the system requires full payload of the packets coming and leaving honeypot. This is task to be performed by the second element i.e. Tcpdump which will capture every packet's full payload. Network monitoring tool Tcpdump is one of the most well-known sniffers for Linux. Build with the libpcap (packet capture library) interface, which is collecting information from packets on the network, also take those which are intended for other hosts machines. This is done through the network interface card with its ability of entering into promiscuous mode. Then packets header information is dumped into log file.

2.2 Data Analysis

For extraction of the precise attack signature made, a data analyzer is developed which is shown in architecture. This interface shows a graphical output which is used by security administrator for finding out mostly attacked port and mostly attacked IP address. The proposed methodology for extracting more precise attack pattern is described below:

i) Configure honeypot to simulate network.

ii) Run Tcpdump for analysis of traffic.

iii) Invoke the auto-run shell script which will be runing in a particular time interval and executing parsers utility for parsing of data from the log file and insert it into the database. The realization of parser utility can be done in any language, which has strong string tokenization capability like Java.

iv) Execute the auto-run shell-script to push the logs data into the database.

v) Login to the interface for viewing the attack patterns and analyzing the data for extraction of good quality signature.

To enable the Security Administrator to select the suspicious data, the GUI has the following features:

i) Ability to display packet information from the database.

ii) Ability to display real time network traffic from data stored in database, as well as historical traffic statistics.

iii) Display the ports, which were attacked within a certain time range.

iv) A timeline based hit statistic showing how many hits per second Honeypot got in a certain time range.

vi) A textual hit statistic over a certain time range. By specifying an IP or a port number it is possible to focus on specific events.

2.3 Database Module

The database records all the packets that are received by the Honeypot program. The database can be composed of raw data which can get disseminated and analyzed later with the help of various methods, for example, using intrusion detection engine. In the proposed method, there are two reasons for using database module. First one is it makes easier searching for a particular packet or range of packets using database, and all required is to construct the correct query syntax. Second, the database facilitates different representations of generated data. The database will be recording all the packets (IP, TCP, and UDP) that are being received by the Honeypot and Tcpdump. There is no need of configuration of graphical interface that is running independent of the Honeypot. This independence is the result of database module. Since past events are all recorded in a database, the web GUI can analyze events without having to interfere with normal operations of the Honeypot. Working this way proposed system will allows for a good selection of data for extraction of attack patterns opposite the existing methods, which are blindly applying content-based pattern extraction algorithm on whole data captured by the honeypot.

2.4 Pattern Extraction

We are having a fully automated method for pattern extraction. Following are the steps required for finding the good quality attack pattern -

i) Identify data of interest (i.e. of significance) from the database by looking at the GUI.

ii) Analyze combined data from different data sources i.e. honeypot and Tcpdump. For each received packet initiate the following sequence of activities:

a) If there is any existing connection state for the new packet, that profile is updated otherwise new profile is created.

b) If the packet is outbound, don’t process the packet.

c) Perform protocol analysis at the network and transport layer.

d) For each stored connection, perform header comparison in order to detect matching IP networks, initial TCP sequence numbers, etc.

iii) Apply content-based string matching on the payload of interest by applying following sequence of activities:

a) If the connections have the same destination port, perform pattern detection on the exchanged messages.

b) If a new signature is created in the process use the signature to augment the signature pool otherwise stop the process.

Along with the existing database of rules, new rules are added for catching peculiar situations.

The web GUI has the following features:

1. Ability of displaying packet information from the database.
2. Ability to display real time network traffic from database data, as well as historical traffic statistics.
3. Real-time configuration interface for J- Honeypot.
4. Configuration of alert notification.

The GUI was built for entry-level security personnel to use and understand quickly, although professional and management folks may find the tool useful as well .User is provided with a quick way for analysis of past events so as to identify attacks. Creating alerts and configuration of honeypots is also allowed by this architecture. Alerts are generated using rules and are classified into serious and miscellaneous alerts.

3. SCOPE OF ARCHITECTURE

There can be any number of attacks that are introduced in the system on a network, here in our project we are focusing on only client side attacks. At the initial phase we are focusing on ip spoofing and DOS attacks.

In the IP spoofing attack if it is not a legitimate user then also we will give access to him for accessing the proxy server and all the activities being done by him are analyzed and tracked for making a useful pattern.

Whereas in DOS attacker will list that user as a blacklist and notify time when he is making an attack. That blacklisted user if tries to access the system then he will be treated as attacker and forwarded to dummy system because he is there in blacklist.

4. CONCLUSION

Honeypots are use to detect and trap attacks being held on network, but they do not prevent attacks. For that we have to do some actions. By using this mitigation framework attack will be indentified and and it will be transmitted to unused ip and security will be provided to data.

By behaviour analysis we will have different patterns used for attacking system so we can use this information for making different security tools. Proposed system is effectively enhanced by cooperating features like making the platform independent, filtering the packets, filtering the suspect content from the network traffic and gather and report network statistics. This framework is not just only for an administrator's tool. It can be used for network traffic monitoring, traffic analysis, troubleshooting and other useful purposes.

The framework presented in this paper shall be useful in extracting good quality signatures from the data obtained by the logs of honeypot and traffic analyzer. Thus the signatures obtained through system are of high quality and result in more precise intrusion detection, not giving too many false positives or negatives. It can also act as an intrusion indicator i.e. how, when and from where different intrusion attempts are taking place. This can be shown through the graphical interface. Honeypots are increasingly deployed in networks; however, they are mostly used passively and administrators watch it just for what happens. The proposed system gives better control to the security administrator on intrusion detection process for extracting good quality attack signature. In our research we developed a Java based network deception tool with monitoring and rule-based intrusion detection capability. We have interfaced it with database, developed a rich set of logging functionalities, and provided a convenient GUI for users to visualize the results.

5. FUTURE SCOPE

As the propsosed architecture is capturing logs and maintaining recored of all the activities being made in the system by the attacker this logs may get usefull for building up security tools. This build security tools can get used by various system to protect their data against many types of attack. In future with the help of tools generated we can also protect system by the insiders attacks and ofcource the one which is made by outsiders.

6. REFERENCES

[1] M. Bailey, E. Cooke, D. Watson, F. Jahanian, and N. Provos, “A hybrid honeypot architecture for scalable network monitoring,” TechnicalReportCSE-TR-499-04, University of Michigan

[2] P. Diebold, A. Hess, and G. Schafer, “A honeypot architecture for detecting and analyzing unknown network attacks,” 14th Kommunikation in Verteilten Systemen

[3] C. Gates, M. Collins, M. Duggan, A. Kompanek, and M. Thomas, “More NetFlow tools: For performance and security.”

[4] F. Raynal, Y. Berthier, P. Biondi, and D. Kaminsky, “Honeypot forensics,” Information Assurance Workshop, Proceedings from the Fifth Annual IEEE SMC, pp. 22-29.

[5] Y.K.Jain, S. Singh “Honeypot based Secure Network System” in IJCSE. Vol 3.No.2 Feb 2011.

[6] HONEYPOT SECURITY February 2008

[7] Advanced Honeypot Architecture for Network Threats Quantification, Robin Berthier, Ph.D., 2009

[8] E. Alata, V. Nicomette, M. Kaâniche, M. Dacier, and M. Herrb, “Lessons learned from the deployment of a high-interaction honeypot,” edcc, 2006, pp. 18-20.

[9] P. Baecher, M. Koetter, T. Holz, M. Dornseif, and F. Freiling, “The nepenthes platform: An efficient approach to collect malware,” Lecture Notes in Computer Science, vol. 4219, 2006, p. 165.

[10] Lance Spitzner, “The Value of Honeypots, Part One: Definitions and Values of Honeypots”, http://www.securityfocus.com/infocus/1492.

[11] Niels Provos, “A Virtual Honeypot Framework”, In Proceedings of the 13th Usenix Security Symposium (Security 2004), San Diego, CA, August 2004, Pp. 1–14.

[12] Christian Kreibich, Jon Crowcroft, “Honeycomb-Creating Intrusion Detection Signatures” Using Honeypot, ACM SIGCOMM Computer Communication Review archive Volume 34,Issue1 January 2004, Pp. 51 – 56.

[13] Paul Innella and Oba McMillan, "An Introduction to Intrusion Detection Systems", http://www.securityfocus.com/infocus/1520.M. Bailey, E. Cooke, D. Watson, F. Jahanian, and N. Provos, A hybrid honeypot architecture for scalable network monitoring, TechnicalReportCSETR-499-04, University of Michigan

[14] P. Diebold, A. Hess, and G. Schafer, A honeypot architecture for detecting and analyzing unknown network attacks, 14th KommunikationinVerteilten Systemen

[15] C. Gates, M. Collins, M. Duggan, A. Kompanek, and M. Thomas, More NetFlow tools: For performance and Security

[16] .16. F. Raynal, Y. Berthier, P. Biondi, and D. Kaminsky, Honeypot forensics, Information Assurance Workshop, Proceedings from the Fifth Annual IEEE SMC, pp. 22-29.

[17] Y.K.Jain, S. Singh Honeypot based Secure Network System in IJCSE. Vol 3.No.2 Feb 2011.

AUTHORS

***1. Shiv H. Sutar**, received the Master’s degree from University of Pune city Pune in Information Technology and Bachelor’sdegree in Computer Science from Swami Ramanand Tirth Marathawada University, city Nanded, state Maharashtra, country India. Currently he is working as an Assistant Professor in MIT College of Engineering, Pune. With 10 years of experience in Academics & Research. His area of interest is Systems Programming, Wireless Sensor Network and Security, Compilers.*

***2. Devyani V. Jadhav** pursuing bachelors degree form University Pune in Computer Engineering at MIT College Of Engineering, city* Pune, state Maharashtra, country India.

***3. Sunetra N. Moholkar** pursuing bachelors degree form University Pune in Computer Engineering at MIT College Of Engineering, city* Pune, state Maharashtra, country India.

***4. Shital S. Moture** pursuing bachelors degree form University Pune in Computer Engineering at MIT College Of Engineering ,city* Pune, state Maharashtra, country India.

***5. Vinay Mutyal** pursuing bachelors degree form University Pune in Computer Engineering at MIT College Of Engineering, city Pune, state Maharashtra, country India.*