http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (7): 1080-1084, 2007
ISSN 1812-5638

Defeating Hackers Through a JAVA Based Honeypot Deployment

[1]V. Maheswari and [2]P.E. Sankaranarayanan
[1]Department of Master of Computer Applications,
[1-2]Sathyabama University, Chennai, India-600 119

Abstract: A Honeypot is a program, machine or system, which is used for network security. The basic idea is to deceive the attackers by making the honeypot seem like a legitimate system. It traps attacks, records intrusion information about tools and activities of a hacking process, and prevents attacks outbound the compromised system. We propose a system, which is used for identifying the blackhats' in a network and recording their activities like when they break in and what are their motives. This is done by Data control, Data capture and Data Duplication mechanisms using a honeypot developed in JAVA. We implement these mechanisms to control the intruders and capture their activities without them knowing that they are being under control. This helps in collecting the information about the attacks and the areas of attacks and to make the network security stronger.

Key words: Honeypot, intruders, data control, security

INTRODUCTION

The need for honeypot arises due to today's security threats that keep changing day to day. More traditional services are extended to Internet. At the same time, attacks and intrusions in the web application system become more popular. For example, if an attacker breaks into a company's mail-server, the damage that is done depends on how soon the attack is detected. If the attack was immediately detected, the attacker could be removed from the system, and the mail-server rebuilt in a more secure manner. Early and successful detection can prevent or mitigate the compromise of data and resources. The most common method has been Network Intrusion Detection Systems, otherwise known as NIDS. This technology works by monitoring network traffic. When it identifies anything it considers an attack, it generates an alert, notifying the administration. The trick is defining and identifying what an attack is. Different NIDS use different technologies, such as signatures, rules based, or anomaly detection (Zhang *et al*., 2003). Hackers exploit trickier and obscure methods (Doring, 2005). Automated attacking tools like worm, attack scripts and Ddos attacks are truly powerful and distinct (Corey, 2005).

Honeypots are a relatively new security technology and are unique for two reasons. First, they work by having the bad guy actually interact with them. Second, honeypots are not a solution; they do not fix a specific problem. Instead, they are a highly flexible tool with multiple applications for security, from preventing attacks, to detecting unauthorized activity, to gathering intelligence hackers (Doring and Erbs, 2005). One of the best applications of honeypots is detection because they address many of the problems associated with traditional detection. Industry and academia show growing interests in honeypot and related technologies (McGrew *et al*., 2006). Different honeypots with varied level of interaction are there which includes Honeynet, BOF, Specterand number of projects such as Honeypot Project and Honeynet Project (Pouget and Holz, 2005).

Though there are different views and definitions given to honeypots, the general definition covering all the different manifestations of honeypots is A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource (Spitzner, 2004). Conceptually almost all honeypots work the same. They are a resource that has no authorized activity; they do not have any production value. A honeypot should see no traffic because it has no legitimate activity. Unlike other security tools like firewalls or intrusion detection system, honeypot do not solve a specific problem. Honeypots are easily flexible to any kind of environment and any type of attacks. This nature makes honeypot outstanding to aid other security tools. Honeypots collect small amount of data, which makes the analysis part easier. Honeypots can collect in-depth information that few, if any other technologies can match. Honeypots

Corresponding Author: Maheswari, Department of Computer Applications, Sathyabama University, Chennai, India-600 119

come in different shapes and sizes, making them difficult to get a grasp of. The general categorization of honeypotsis based on their level of interaction. The simplest honeypot is simply a socket-based program that opens up a listening port. It ranges from simple port monitor to full product systems, which can be emulated under various OS. A honeypot is a closely monitored capturing resource that we intend to be probed, attacked or compromised. The value of honeypot is determined by the information that we can obtain from it. Monitoring the data that enters and leaves a honeypot let us gather information that is not available to NIDS (Provos, 2003).

Data control, capture and duplication are basic requirements for any kind of honeypots. Data control is concerned with protecting a system that an attacker might target from a compromised honeypot. Data capture relates to the actual goal of deploying a honeypot. The goal is to gather information on attackers and their tactics. This data can help to analyze the steps an attacker took to compromise a honeypot and how the honeypot was used after it was compromised. One common way to achieve data control is to limit the number of outgoing connections or the bandwidth available to honeypot (Krasser *et al*., 2005).

There are multiple threats to data capturing. First, encryption can be used to hide the content of traffic to the capturing facilities on the honeypot. Moreover, an attacker can disable capturing tools running on honeypots, and known limitations of these tools can be exploited. An attacker may be able to detect the presence of data control facilities on a honeynet (Joo *et al*., 2006). For example, the attacker can try to initiate a number of connections beyond a connection limit by the honeynet. If this fails, this is an indication that something is odd about the machine the attacker compromised. Moreover, if the honeywall alters certain malicious data packets to prevent other machines on the Internet from being compromised, the attacker can deduce from failed attacks that a honeywall is present in the network. If the attacker sends those malicious packets back to a machine under the attacker's control, he or she can actually see whether the packet has been altered. Data capture facilities may also be detectable by an attacker. For example, the Sebek software described above, which is used to capture data from honeypots, can be detected, or certain indications can suggest that the software runs on a honeypot.

There are many free and commercial honeypots each built of their own kind and type. These honeypots range from simple low-involvement to risky high-involvement. Honeyd is a low-to-medium-interaction honeypot system. Installed on a Unix system it listens on the Network Interface Card (NIC) for incoming ARP requests. Bait'n'Switch is a honeypot response mechanism that distracts the attacker from the valuable targets. Bait'n'Switch is realized as a snort in-line extension. Honeynet is a high interaction honeypot. Instead of simulating a single vulnerable host, a honeypot simulates a complete network and amplifies the detection and analyzation possibilities. Now methods of data capture and data control proposed by Honeynet project shows greater flexibility and higher access control ability, which can be applied on both production honeypot and research honeypot The Deception Toolkit (DTK) is a set of free tools (mostly written in perl) designed by Fred Cohen. DTK uses deception to counter attacks. J-Honeypot is a honeypot developed in JAVA has a rule-based intrusion detection engine, whose design is guided by the analysis of real world attack data. was collected by (Mai *et al*., 2004).

This study has been undertaken to implement a Java Based Honeypot for efficient Data Control and Data Capture. The existing honeypots are either can be used in an Unix/Linux Environment or in a windows platform. In this paper we have given the implementation of Java based honeypot and the intruder's activity is recorded through the multiple layer of Data control and Data capture. The system provides for the intruders an access to a duplicated data, which helps in recording their activities. The Firewall is placed next to the honeypot, which restrict the outbound connections from the honeypot. A similar work on platform independent honeypot has been done which was used as a rule based analysis tool (Mai *et al*., 2004). This honeypot has been designed in Java to make it a Platform independent product.

IMPLEMENTATION

Architectural model for JAVA based honeypot: Figure 1 gives the overall architecture of our implementation. It includes a Graphical User Interface, which makes easier to view the events that occurred to the honeypot.

Data control module: Data Control controls the activities of attackers by limiting options. This area is designed to control the activities of the intruder. If the person is an authorized user and access is from unrestricted IP. It provides access to all files. If the person is an authorized

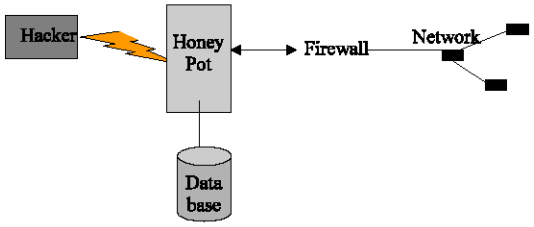

Fig. 1: JAVA honeypot architeture

user and the access is from restricted IP. It provides limited options. Numbers of Outbound Connections are limited using a Firewall.

Data capture module: Data capture deals with collecting and recording the activities of Honeypot. It Deceives Intruder by capturing all activity within the honeypot and the information that enters and leaves the honeypot, without attackers knowing they are being watched. Attacker does not know that he is working in a fake system. Information collected is stored in database. Captures specific amount of data by reducing traffic. Captures the attackers activity on honeypot itself. Captures the attackers activity even if it is in encrypted form.

Data duplication module: This module does the Data Duplication by producing the fake files for the attackers. This also deceives the intruders that a real data is being accessed by them and continue with their activities.

Database module: MS-ACCESS was used as a database to log and store all the activities of the intruders. The data, which is received by the honeypot, can be stored, disseminated and analyzed later using various methods for example Intrusion detection engine.

RESULTS AND DISCUSSION

Simulation environment: The JAVA Based Honeypot was setup and run in an Intranet environment. The setup was verified with the valid users and Invalid users and also Autorized and Unauthorized user access. The data control activity was done based on the user's identity and the options were limited to them. The data capture was done as soon as the hackers are diverted to the honeypot and it was recorded in the database.

Placement of honeypot: Placement of honeypot is critical in protecting the internal network (Fig. 2). Basic rule is to locate honeypot where it is relatively isolated from your production system. Router and Firewall work together to monitor and control traffic. Rules for firewall and router will depend on whether you are monitoring internal or external intrusions.

Simulation results: The users were checked for authentication and if authorized user is from an unrestricted IP address the user was permitted to access all the files. If the person is an authorised user and the access is from restricted IP. It provides limited options. The Fig. 3 shows the Data Control implementation.

If the user is unauthorized then it is an illegal access into the system. Instead of denying the access to the hacker, the honeypot deceives the hackers by allowing them into the network but providing only with the fake files and capturing and storing their activities (Fig. 4 and Fig. 5).

The above simulation results shows that this JAVA based honeypot can be used to control and capture the activities of the hackers. If the honeypot is configured correctly, the intruder won't have any idea that he is in a

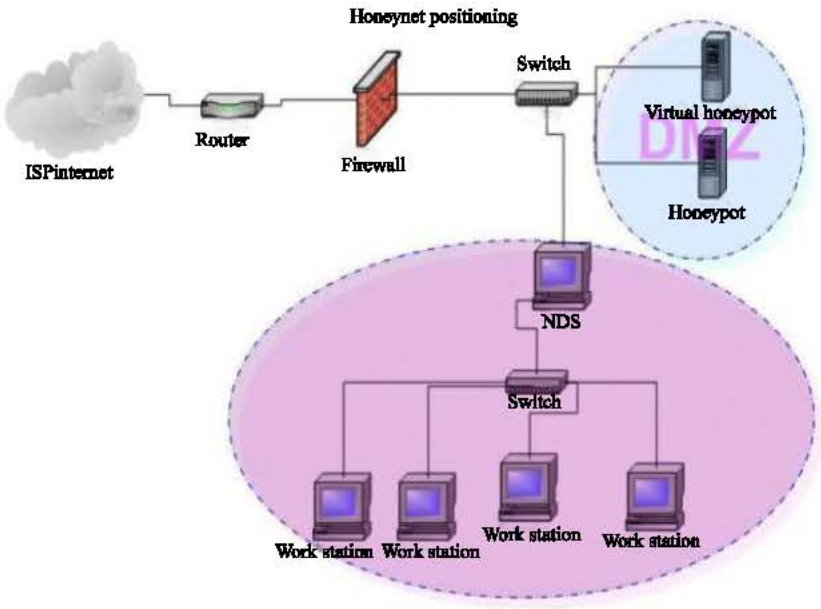

Fig. 2: Placement of honeypot

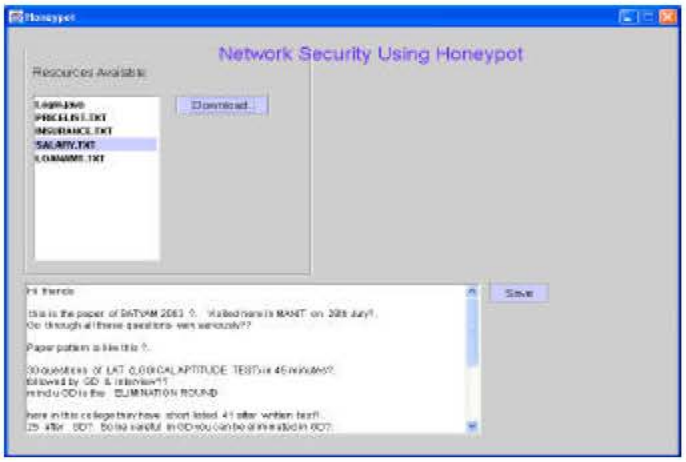

Fig. 3: Resources available

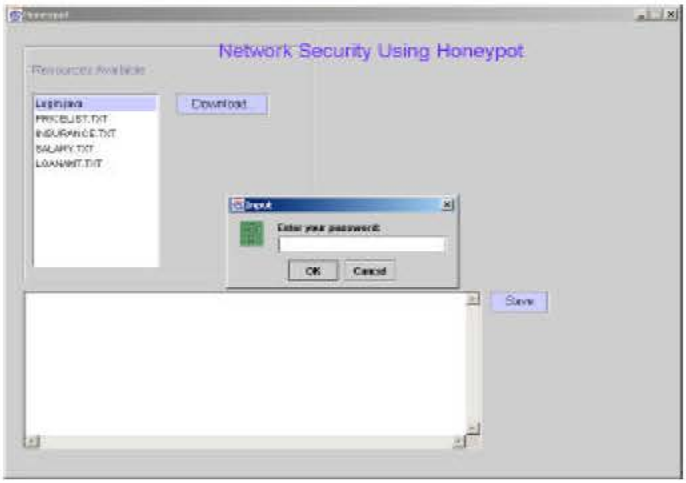

Fig. 4: Activity capture screen

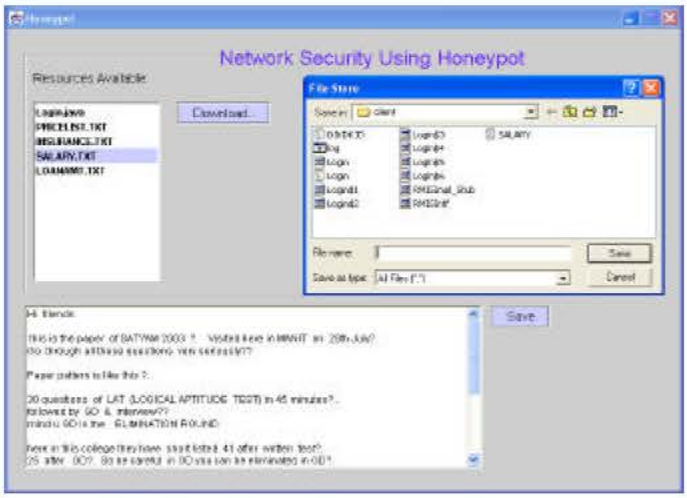

Fig. 5: Activity capture screen

fake system and that his actions are being monitored. Deceives the Intruder by providing Dummy files instead of the original files.

Even though the person is an authorized user, it gathers information if the request is from an restricted IP address. Easy to install, configure, deploy and maintain and also reliable.

Future enhancements: Our Implementation of honeypot is a low-interaction form of honeypot, which has been simulated under internal network. The Intrusion Detection System can be combined with this to generate an alert system whenever hackers are trying to intrude into the network if this is an Internet based application. The IDS can analyze the incoming packets and can be used to generate new rules and signatures based the attack pattern. The Emulation of the various Protocols and Operating system can be done using this Java based Honeypot.

CONCLUSION

In present study we have implemented a Java based honeypot and have shown that it can be used to control the activities of the intruders. This work supplements the J-honeypot which has been developed with a rule-based detection capability. The existing honeypots are either windows based or Linux based honeypots. We have developed a Java based honeypot which is used for implementation of Data capture and Data control. Also it captures their actions and stores it in a database, which can be used to analyze the tactics and tools used by the hackers. It also deceives the intruders that they are working with real data, by providing duplicate data. The honeypot has been designed with GUI, which helps the users to visualize their results.

REFERENCES

Corey, J., 2005. Advanced Honeypot Identification and Exploitation. http://www.phrack.org/fakes/p63/p63-0x09.txt (current 9 Jun. 2005).

Doring, C., 2005. Improving Network Security with Honeypots. Thesis.

Doring, C. and H. Erbs, 2005. Improving network security with honeypots. Conceptual framework for a honeypot solution, ulyanovsk state technical University German Honeynet Project.

Joo, O.C., R. Budiarto and G.C. Sodhy, 2006. Honeynet in network security using multi-layer Data control and data capture mechanisms. Proceedings of the 3rd IASTED International Conference on Communications and Computer Networks, pp: 114-119.

Krasser, S., J.B. Grizzard and H.L. Owen, 2005. The use of honeynets to increase computer network security and user awareness. J. Security Edu., 1: 23-37.

Mai, Y., R. Upadrashta and X. Su, 2004. J-honeypot: A java-based network deception tool with monitoring and intrusion detection, ITcc, international conference on information technology: Coding and computing (ITCC'04) 1: 804.

McGrew, R., B. Rayford and J.R. Vaughn, 2006. Experiences with honeypot systems: Development, deployment and analysis. Proceedings of the 39th Hawaii International Conference on System Sciences, 2006

Pouget, F. and T. Holz, 2005. A pointillist approach for comparing honeypots. Proc. of the conference on detection of intrusions and malware and vulnerability assessment. (DIMVA, 2005), Vienna, Austria, July 2005. French and German Honeynet Project.

Provos, N., 2003. A Virtual Honeypot Framework , CITI Technical Report 03-1.

Spitzner, L., 2004 . The honeynet project: Trapping the hackers, IEEE security and privacy, March/April 2004, pp: 15-23.

Zhang, F., S. Zhou, Z. Qin and J. Liu, 2003. Parallel and Distributed Computing, Applications and Technologies, PDCAT., pp: 231-235.