REDUCING THE FALSE ALARM RATE OF NETWORK ATTACKS WITH THE USE OF HONEY POTS TOGETHER WITH AGENT-BASED INTRUSION DETECTION SYSTEM

BABAK KHOSRAVIFAR AND AMIRHASAN AMINTABAR
Eastern Mediterranean University (T.R.N.C)
babak@ciu.edu.tr, amirhasan.amintabar@emu.edu.tr

Abstract: *When traditional firewall and intrusion detection systems (IDS) are used to detect possible attacks from the network, they often make wrong decisions and abort the safe connections. In this paper a novel system is presented which is based on distributed agents and a pseudo-network called honey pot. Utilizing the honey pot scheme, this system is capable to avoid many wrong decisions made by IDS. In this system alarming adversaries, initially detected by the IDS, will be forwarded to a honey pot network for a more close investigation. If, as a result of this investigation, it is found that the alarm decision made by the IDS is wrong, the connection will be guided to the original destination. This action is hidden to the user. The policy of attack detection via honey pot or IDS will be dynamically updated and adapted based on the previous records of adversaries. Such a scheme significantly decreases the alarm rate and provides a higher performance of IDS. In this paper the architecture of the proposed system is described, a theoretical analysis of its behaviour is given and its possible extension and implementation are explained.*

Keywords: Agent-Based IDS, Honeyd.

1 Introduction

Computers today are no longer used as standalone units. More often, they are networked into large distributed systems where each individual computer can use applications which are distributed throughout the system and shares resources with all the other subsystems [3]. By increasing the usage of the Internet and implementing commonly used tasks through it, the concept of distributed applications has been drastically grown. There are numerous reasons and requirements for using distributed systems. One class of such systems is Geographically Distributed Artificial Intelligent (GDAI) systems consist of some dynamic agents which are geographically distributed through the Internet. Another important class is grid systems. Grid applications enable clients to solve computationally complex problems in a coordinate way [7].

A known negative feature of distributed systems is the threat of intrusions. Since they are distributed and there is a communication between them, they are more vulnerable to attacks and denial of service. Port-blocking is a traditional solution for the vulnerability of the Internet applications. Currently firewall method and Intrusion Detection Systems (IDS) have been practically developed to block variety of threats through incoming port connections. The duty of firewall which cooperates with the IDS is to block some connections between local servers and remote clients. However there is always the possibility of malfunctioning of the firewall which blocks the immune connection on a mistake. These results are false alarms [1] [2].

The paper's first contribution proposes an infrastructure which deals with attack recognition. This will help to prevention of future similar threats. For this purpose a honey pot which is real-time adaptable to the main net has been used as a virtual unreal network. Second contribution deals with correctness of the detection. If the IDS makes a mistake in detecting an attack, honey pot can be a good solution to retaliate the mistake. Honey pot in relation to the end-user finds out whether it is a real threat or just a normal connection. If not, it will shift the connection with all the saved information to the main net while the end-user does not feel any changes.

The paper is organized as follows; Section 2 briefly surveys the existing systems to reduce the false alarm rate of network attacks. Section 3 presents a proposal model of the proposed system. Section 4 describes the structure of the Manager as the main component of the proposed scheme. In Section 5, the work of the Manager is outlined. Section 6 explains how a network attack is handled by the proposed system. Finally, Section 7 concludes the paper.

2 Existing System Architecture to Reduce False Alarm Rate

2.1 Intrusion Detection System

Intrusion detection technology in general helps find out the illegal intrusions from inside and outside by tracking the intruders' trail, such as the records of failure access trails. It acts as an active defense against illegal intrusions. Thus it plays an important role in network security. As far as the computer and network system are concerned, IDS is a software system which detects the evil attacks from outside illegal intruders and the illegal behavior of inner users when they exceed their authority [8].

A typical IDS consists of the following parts: event generator, event analyzer, response units and event databases. The data are exchanged by Gidos_generalized intrusion detection object [9] between the parts. Once an adversary is detected event generators obtain the required information and transform it into standard format. Respectively event analyzers analyze the data and generate Gidos. Gidos are processed by response units. Event databases store the events and Gidos.

Current network intrusion detection systems often work as misuse detectors, where the packets in the monitored network are compared against a repository of signatures that define characteristics of an intrusion. Successful matching fires an alert. Generally, a good signature must be narrow enough to capture precisely the characteristic aspects of adversaries. At the same time, it should be flexible enough to capture variations of attacks. Failure in any way may lead to either large amounts of false positives or false negatives [4]. Figure 1 clarifies the system in more detail.

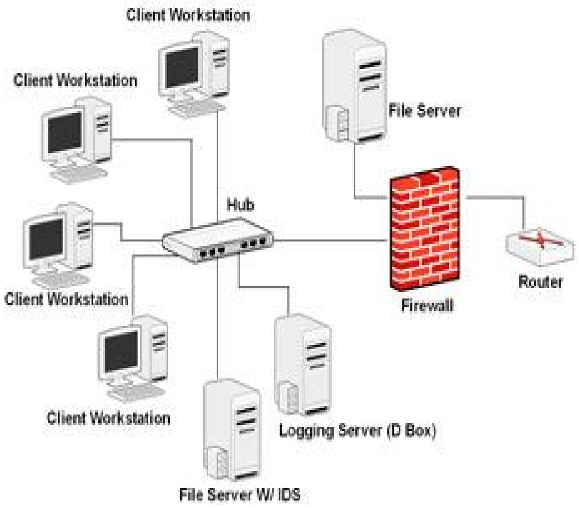

Figure 1: Basic Intrusion Detection System

2.2 honey Pot Systems

Honey pot is an unreal network system designed to trap crackers and intruders. This system was introduced initially by Clifford Stoll in 1990 [14]. The author used unimportant computer information to trap a computer attacker at Lawrence Berkeley Labs. The honey pot is used as bait in the form of a vulnerable system to trap hackers and keep them away from accessing the critical information in the main system. Using a honey pot system instead of the main network makes it possible to observe the adversary activities, collect information and detect new information coming from the intruders. It is important that honey pots do not weaken traditional Internet security systems' performance. Some examples of potential malicious traffic can be listed as spoofed IPs, ICMP packet, UDP packets depending on their source and destination and TCP traffic directed to well-known vulnerabilities with unusual characteristics [15].

Since 1990 there have been many research activities on implementation of honey pot systems. In particular, “Cybercop Sting” [15] as an unreal server simulates a network consisting of several types of devices including Windows NT servers, Unix servers and routers. Each of them is capable of interacting, recording and routing adversary attacks to the network. “Netfacade” is a commercial honey pot which simulates IP addresses. It runs vulnerable services like “Cybercop” but on a larger scale. It is capable of simulating class C network up to 254 systems. In addition, it can simulate 7 different systems with various services. One more system, “Back Officer Friendly” was released in 1998. Although it did not deliver much functionality, it is very useful as it demonstrates the basics of honey pot concept to people. As a matter of fact, 1998 was an indication of raising the interest in honey pot systems. “Honey Pot Research Alliance” was a project involving the entire security systems. As a consequence many honey pot tools were introduced via the research organizations like Snort, Sebek and Advanced Virtual Honey Pots [15]. Snort was capable of blocking and disabling attacks instead of just detecting them. Sebek was able to capture hacker activities in honey pot by logging their keystrokes and finally virtual honey pots deployed multiple honey pots in one computer [16].

2.3 Agent-Based IDS

Agent-Based IDS is a distributed system used to improve the detection speed. It uses some external agents in the network to detect the attack and warns the generation process and the core which manages the system and generates reports for the administrators [7, 10].

3 Design and Implementation of a New Scheme

This section contains the explanation of our approach which is considered as a solution to impersonation and eavesdropping factors of security problem. This topology consists of distributed agents which are all under the surveillance of a manager. The manager, as a main part of the topology, cooperates with honey pot in order to lure the attacker and provide an attractive but diversionary playground.

Figure 2 explains our main system structure in the left hand site and indicates the main Network accompanied by firewall and manager in the right hand side. In this topology main Network is a “TCP/IP based Network” which contains clients and servers. *Clients* are windows 2000 or XP (or Redhat Linux or Fedora core 9.0) based computers. Snort 2.0 is used as an IDS and is configured to work as an agent. Honeyd is a honey pot system that is configured to work in this system. We considered a “switched network” in our approach. It is a configurable switch which is supposed to mirror all the traffic to the specific port. Its duty is to send a copy of all packets to the port where the manager is plugged in. Numerous servers and services can be supported by this system such as application, authentication & authorization, file, database servers, firewalls, gateways, workstations and etc. [11]. There are some systems listed in table 1.

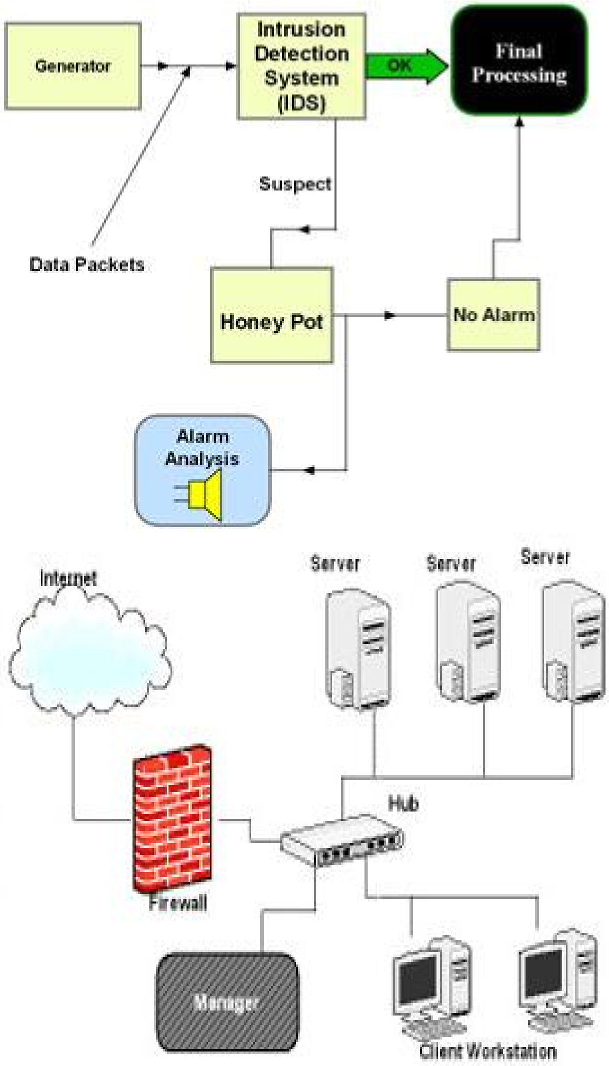

Figure 2: The main structure

Table 1: Some supported systems

Windows 2000 & Windows 2003	Web Server	IIS 6.0 & older
	Domain Controller (Active Directory)	
	DNS	
	Mail Server	Microsoft Exchange 2000 & 2003
Linux	Web Server	Apache HTTP Server 2.0.38 & older
	Mail Server	Qmail 1.01 & Sendmail 8.12.10

4 Manager

Manager is actually the boss in the proposed architecture. The platform used is Linux. Since the platform is more complicated than other parts, we describe its partitioned infrastructure in Figure 3. This structure consists of four divisions; Director to make decisions and rules, Records as a Database which used to save information relevant to detection, Honey Pot as a pseudo net used to prevent false blocking and signature creation [4], Executive to transform the director's information into executable orders.

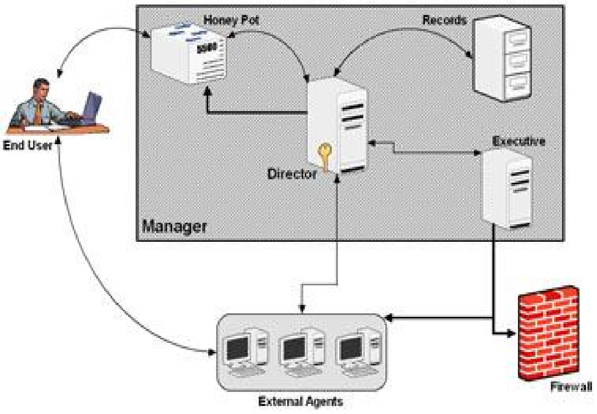

Figure 3: The manager's structure

4.1 Director

Director acts as a commander in this system. Once the external agents report an attack to the manager, the director compares the situation with the "Agent Risk Threshold". Risk threshold is the boundary which would warn an attack if interaction exceeds the limit of the specific agent. It is accessible in records. If the risk of the attack exceeds the threshold, it would engage the manager parts to deal with the connection.

The honey pot interacts under the control of the director and sends reports to it. The director has a report analysis section which closely investigates all the reports. To do so it refers to records and makes decisions. This is the main duty of director. It also determines whether the interaction has exceeded the "Honey pot Risk Threshold" or not. After detecting an attack via agents, if honey pot realizes that the connection is exceeding of its risk threshold, the attack will be ascertained and consequently the corresponding report, concerning the information about the attack, will be ensured to the data base.

The director manages all tasks and makes decisions by employing the parts dynamically. It also sends all the relevant information and directs to precision of decision making. For this aim it uses the signature creation algorithm shown in figure 4.

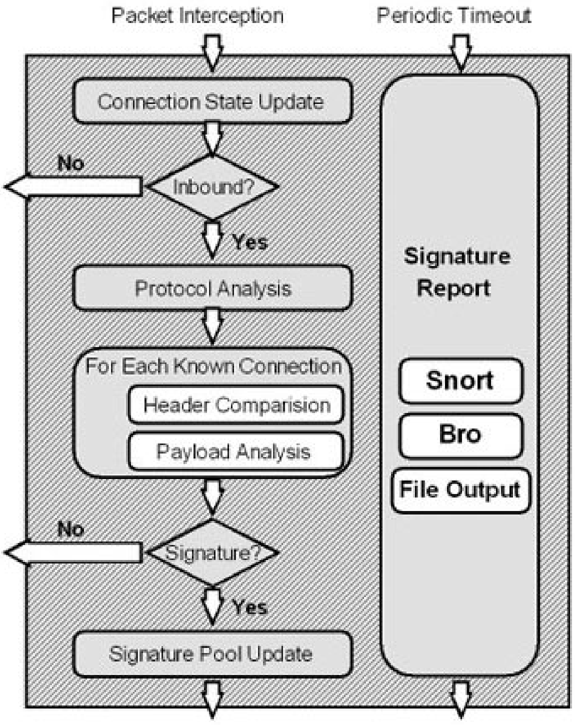

Figure 4: Signature Creation Algorithm [4]

Director's responsibilities are as follows: Confirming the attack detection, informing executive to make rules derived from the threat and also add them to the Database when the adversary has been approved(see figure 4); Blocking the attack by providing a situation where the connection would not be broken up; Establishing the connection between honey pot and the main network; Matching the honey pot with the main net when the system is initialized and Handing over related information to executive to generate orders. Director carefully uses records in order to make decisions. It uses honey pot for identifying the attacker and looking into connection as scrutiny.

4.2 Records

The records provide a Database including a crime table, policy table, reports and event logs. Director for making decisions refers to this Database and also enters new rules and signatures derived from new attack. It constantly modifies the rule and signature list that are in the external agent.

4.3 Honey Pot

Honey pot is an unreal Network used to identify the adversaries. It is responsible for holding up the connection as long as the director makes a decision about the situation. For this purpose Honeyd [4] has been used in this topology. Honeyd is a framework used as virtual honey pot for simulating different computer systems at the network level. IP protocol suites [17] are supported by Honeyd. It is capable to respond to network packets whose destination IP address is one of the simulated virtual honey pots. Honeyd is dynamically adapted by the network behavior of each configured operating system.

Honeyd is installed on a Unix system. It listens on the network interface card (NIC) for incoming ARP requests. If an ARP request recognized Honeyd initiates an ARP request itself. If no response to the own ARP request is given and a rule for the requested IP exists in the configuration file, Honeyd overtakes the IP and starts pre-configured services on the specified ports. Honeyd is able to emulate the behaviour of most common IP-stack implementations (Windows and Linux) that can be detected by the tools nmap[1] and x-scan[12].

There are several ways to do this. For example, creating special route for the virtual honey pots that point to the Honeyd host, or using proxy ARP [17] or network tunnels worth to be considered. Here one of the main roles of the honey pot is the signature creation algorithm. Signature creation algorithm is the process by which the system detects new and unknown threats. There is an example of configuration in figure 5.

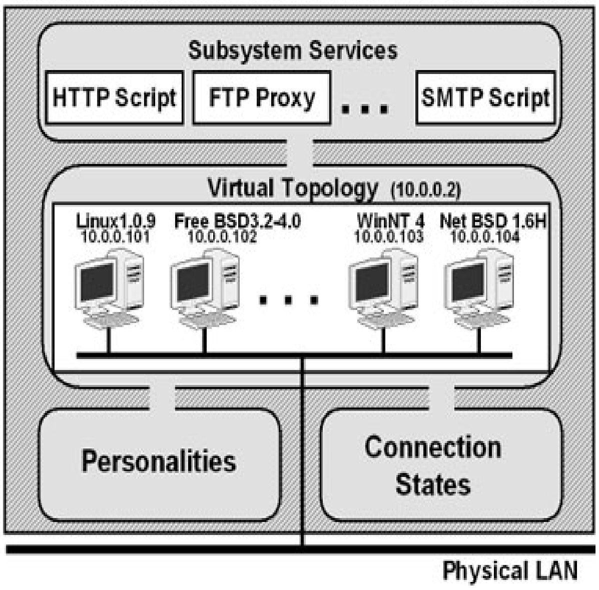

Figure 5. Sample honeyd structure

4.4 Executive

Executive acts as director's subordinate. Its duty is to generate orders for various purposes. For example, when honey pot notified the attack, director would inform the executive. Accordingly the executive will generate orders as a precaution of further threats, and also it will order the firewall to block such packages. It establishes the real-time connection with external agents.

4.5 External Agents

The first step of the detection process is done via agents, the real intrusion detection systems used for the primary detection. Intrusion detection system is discussed in section 2. Considerable point in this system is that the agent handles all of the network connections of the computer which has been installed on in addition to act as an IDS.

5 The Detection Process

Our proposal is a combination of the honey pot technique with Intrusion Detection System (IDS) in a new way that as soon as an intrusion is detected, honeyd takes steps to identify the suspected attacker.

In fact, by initializing the system, the director matches honey pot with the main Net, and then honeyd is ready to interact with the suspected adversaries. In our method, director as a commander supervises data transmission reports. When the external agents detect an attack such as, they report the event to the director which will decide how to manage it. Such attacks can be listed as follows; Nimda, CodeRed (3 versions), MyDoom, W32/Welchia.D, Attempts to access the IIS-samples, Attempts to get '/etc/psswd, Attempts to execute cmd.exe, Attempts to open a new network socket in order to download further hacking utilities, A warm that starts on each infected system an email relying server which it uses for its further spreading. Director orders switch to send a copy of any packages which their destination is the said agent, and also commands the agent not to answer, thus director will receive the trail of suspended threat packages. All of the process must be hidden to the end-user.

Indeed, the connection will be shifted to the honey pot. Honeyd interacts like a normal network with end-user, as director monitors all the event reports which have been sent by honeyd. It refers to the Database saved in records in order to make decisions. If honeyd by any hooks requires any information to send to the end-user, director via executive will ask external agents to respond and send back. This information will be dispatched to the end-user under the tightened security of director, so director completely keeps the connection under the control.

When director completely verified the attacker, it would command the executive to break up the connection and also stored information of the attacker will be added to the Database. Executive with a SNMP will order firewall to block such packages henceforth. Thus the attack was detected, besides the attacker identified which is substantial information for administrator. If the director ascertained that the detection was wrong, without breaking up the connection via executive, it will ask external agents to carry on connection. It will also hand over all saved information relevant to the connection. Therefore the number of false positives will be decreased. It's a crucial factor to show the precision and accuracy of the detection.

6 An Attack Scenario

Due to variety of the threats and also their numbers, it is almost impossible to distinguish the recognizable threats precisely, however majority of known attacks which can be detected by famous IDS especially the proposed one in this paper are detectable. Typical examples can be listed as follows: DOS Port Scan, Dictionary attacks. The real example will explain the detection process more precisely; We have a web server in a network. A part of our website is hosted on the server which waits for the user to login. MYSQL database has been employed for this method.

Assume a user who tries to login enters ((ali')) as a username in the user view part. This expression can be a test for an attack which called SQL injection, but there is a probability that he has unintentionally pushed the wrong button. At this step the based agent on the server has detected the suspected threat; it would report it to the manager. The manager after checking its importance level, if it exceeds the risk threshold level of the agent, it will command the agent not to respond the packages with the suspected destination address anymore. Hence, honey pot will answer to the user instead.

Honeyd interacts directly. It sends interaction reports to the director to compare with the threshold. End-user will be recognized as an attacker as soon as its behaviour exceeds honey pot risk threshold. Assume that the threat was important. After commanding to the agent, honey pot via its virtual server, will interact with the end-user. For instance it will send the user a message with this warn that your username is wrong. Then his subsequent acts will be completely checked. If he followed other ways of attack, he/she will be recognized as an attacker, but if he/she tried to re-enter his username and password correctly, the system will find out that he's a safe user. By sending the information to the agent of the main server, the manager will order it to re-establish the former connection with the end-user henceforth.

7 Conclusion

The paper presents a novel system on a distributed agent based network. The system, in case of suspended adversaries, changes the path of connection to the honey pot system for a more close investigation. Decreasing false alarms' rate is the system's objective. The system also tries to store any necessary information about the identified attacker. We should pay more attention to the following problems:
How to change the connection when the director engages honey pot? We do research for shifting the connection with the all related data in more details. We have to expose director to more aggressive traffic patterns to get a better understanding of its

performance. Director has to intelligently deal with the data communication between honey pot and the end user. The coordinating model of honey pot in relation with agents still needs further discussion.

References

[1] Hannon C. and Richard J.R., "Addressing Security Issues in Geographically Distributed System". *Proceeding of the Fourth Mexican International Conference on Computer Science, IEEE* 2003.

[2] Grosspietch K.E and Silayeva T.A., "A Combined Safety/Security Approach for Cooperative Distributed systems". *Proceeding of the 18'th International Parallel and Distributed Processing Symposium, IEEE* 2004.

[3] Dobry R. and Schanken M.D., "Security Concerns for distributed Systems". *National Security Agency, Fort Meade, MD 20755-6000, IEEE* 1994.

[4] Kreibich C. and Crowcroft J., "Honeycomb: Creating Intrusion Detection Signatures Using Honey pots". *ACM SIGCOMM Computer Communications Review, Volume 34, Number1:* January 2004.

[5] Vigna G, Valeur F. and Kemmerer R.A., "Designing and Implementing a Family of Intrusion Detection Systems". *Reliable Software Group, Department of Computer Science, University of California Santa Barbara.*

[6] Snort 2.0 Documentation , 2003 , *"http://www.snort.org"*

[7] Guangchun L., Xianliang L. and Jiong Zhang L., "MADIDS: A Novel Distributed IDS Based on Mobile Agent". *Information Center of UEST of China, ChengDu.*

[8] Zhengjun T., "The design and implement of Network industry". *Published by. 4.2002.*

[9] Donghai H., Chao W. and Li Q., "Example Anatomy of IDS". *Published by Tsinghua University 5.2002.*

[10] Ehsani M. , Houshyarifar H., "An Infrastructure to Prevent, Detect and Investigate Computer Crimes in Cyber Cities".

[11] Even L.R., "Honey pot systems explained". *SANS, July, 2000.*

[12] An Introduction to Cryptography, Network Associates, Inc. 1999.

[13] Gartner F, "Byzantine Failures and Security: Arbitrary is not (always) Random". *Proc. GI-Jahrestagung* 2003, Frankfurt 2003.

[14] Teiltagung Sicherheit, The Cuckoo's Egg, Tracking a Spy Through the Maze of Computer Espiounage, 1990.

[15] Hoepers C. and Steding J.K., "Honeynets Applied to the CSIRT Scenario".

[16] Talabis R., Honeypots 101: "A Brief History of Honeypots". *The Philippine hpneynet project* 2002.

[17] Stivens W.R., TCP/IP Illustrated. *Volume 1,* Addison-Wesley, 1994.