Enhancing Network Intrusion Detection System With Honeypot

Sujata Yeldi, Sweta Gupta, Tanmay Ganacharya, Shirish Doshi, Dhanashree Bahirat
Prof. Rajesh Ingle, Anandamoy Roychowdhary
Pune Institute of Computer Technology (PICT)
Department of Computer Engineering
Pune - 411043, India
y_sujata@yahoo.co.in, sweta_pict@yahoo.com, tanuganacharya@hotmail.com, shirishdoshi@hotmail.com, d25382@yahoo.com, ingle@ieee.org, ary@acm.org

Abstract— Traditionally, the strategy to defend one's organization as best as possible is to detect any failures in the defense, and then react to those failures. The problem with this approach is that it is purely defensive; the enemy is on the attack. Honeypots attempt to change this; they give the organizations the ability to take the initiative. Honeypots help to explore new vulnerabilities in an organization and defining the security policy of an organization. A honeypot is used in the area of computer and Internet security. It is a resource, which is intended to be attacked or compromised to gain more information about the attacker and his attack techniques. It can also be used to attract and divert an attacker from the real targets. This paper focuses on the possibilities of honeypots and their use in an educational as well as productive environment. Also 'Mirage' – honeypot developed by us has been discussed. This system was not only implemented but also tested for several days and the collected data of the attacks was analyzed. After all, a conclusion about the new technology of honeypots and a look into the future of honeypots will be dared.

1. INTRODUCTION

La[1]nce Spitzner, key member of a research group in the United States called Project Honeynet, defines the term honeypot as follows:

"A honeypot is a resource whose value is in being attacked or compromised. This means, that a honeypot is expected to get probed, attacked and potentially exploited. Honeypots do not fix anything. They provide us with additional, valuable information."

A honeypot is a resource, which pretends to be a real target. The main goals are the distraction of an attacker and the gain of information about an attack and the attacker. Honeypots do not help directly in increasing a computer network's security. On the contrary, they do attract intruders and can therefore attract some interest from the blackhat community on the network, where the honeypot is located. An Intrusion Detection System (IDS) plays an important part in nearly every honeypot, and especially in honeynets, as it is an essential component in gathering information.

There are two categories of honeypots - *production honeypots* and *research honeypots*. A production honeypot is used to help mitigate risk in an organization while the second category, research, is meant to gather as much information as possible. These honeypots do not add any security value to an organization, but they can help to understand the blackhat community and their attacks as well as to build some better defenses against security threats.

How can a honeypot be used to add security to a network? A honeypot is a resource, which is intended to get compromised. Every traffic from and to a honeypot is suspicious because no productive systems are located on this resource. In general, every traffic from and to a honeypot is unauthorized activity. All data collected by a honeypot is therefore interesting data. A honeypot will in general not produce an awful lot of logs because no productive systems are running on that machine. Analyzing this data should get much easier by these simple facts. Data collected by a honeypot is of high value and can lead to a better understanding and knowledge, which in turn can help to increase overall network security.

2. LEVEL OF INVOLVEMENT

Honeypots can be classified by level of involvement also. The level of involvement does measure the degree an attacker can interact with the operating system. Two groups of involvement are built:

Low-involvement: A low-involvement honeypot typically only provides certain fake services. On a low-involvement honeypot there is no real operating system that an attacker can operate on. This will minimize the risk significantly because the complexity of an operating system is eliminated. On the other hand, this is also a disadvantage. It isn't possible to watch an attacker interacting with the operating system, which could be really interesting.

0-7803-7651-X/03/$17.00 © 2003 IEEE

A low-involvement honeypot is like a one-way connection. We only listen, but we do not ask questions ourselves. The role of this approach is very passive.

High-involvement: A high-involvement honeypot has a real underlying operating system. This leads to a much higher risk as the complexity increases rapidly. At the same time, the possibilities to gather information, the possible attacks as well as the attractiveness increase a lot. One goal of a hacker is to gain root and to have access to a shell, which is connected to the Internet. A high-involvement honeypot does offer such an environment. As soon as a hacker has gained access, his real work and therefore the interesting part begins.

Unfortunately the hacker has to compromise the system to get this level of freedom. He will then have root rights on the system and can do anything at any moment on the compromised system. This system is no longer secure. Even the whole machine can't be considered as secure anymore. This doesn't matter if he is in a sandbox or a VMWare box because there could be ways to get out of these software boundaries. By providing a full operating system to the hacker, he has the possibilities to upload and install new files. This is where a high-involvement honeypot can show its strength, as all actions can be recorded and analyzed.

3. HONEYNET

The honeypots run on a single machine. To make honeypots look more like productive systems, honeynets are setup. The common elements of a honeynet are:

- A firewall computer, which logs all incoming/outgoing connections and provides Network Address Translation (NAT) service and some Denial of Service (DoS) protection.
- An intrusion detection computer. The IDS box is sometimes on the same box as the firewall, but it should be on an entirely separate computer that can see all of the network traffic. It also logs all the network traffic and looks for known exploits and attacks.
- A remote syslog computer. The honeypot is slightly modified so that all commands an intruder would use are sent to syslog. Syslog is then set to remote log to a remote syslog host.
- The honeypot itself. When setting up the honeypot as little changes to it are made. Any changes made could tip off an intruder that this is a honeynet. Placement of honeypots in a network is very crucial. Placing a honeypot on the intranet can be useful if the detection of some bad guys inside a private network is wished. If the main concern is the Internet, a honeypot can be placed at two locations: In front of the firewall, Behind the firewall (intranet), Demilitarized Zone (DMZ).

Honeynets allow the simulation of realistic productive environments at the cost of a more or less immense administrative and technical expenditure. The information collected from a honeynet can also be used to educate people. For instance, log entries that are a result of a break-in can be shown to other system administrators so they will know what to look for. It will also provide users with a picture of what kind of attacks are going across Their network

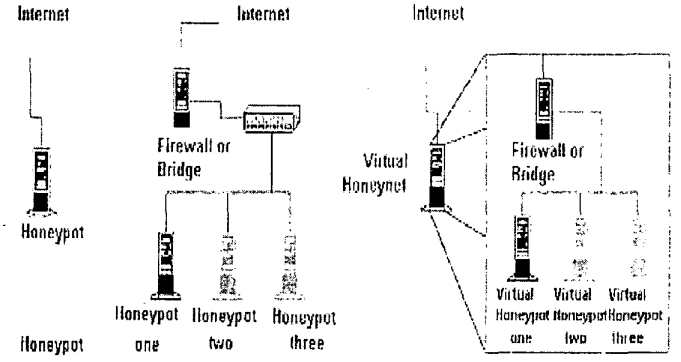

Figure 1 Honeynet Architecture

4. AVAILABLE HONEYPOTS

There are few free and commercial honeypots available in the market. Their functionality differs greatly, as well as their complexity and ease of use. The products range from simple low-involvement to risky high-involvement honeypots.

Some of the available honeypots are mentioned below.

- **ManTrap** by Recourse Technologies: A *mid-involvement* honeypot running on Solaris operating systems
- **Deception Toolkit** by Fred Cohen: A set of perl scripts emulating some services and acting as a low-involvement honeypot
- **Specter** by Neo Worx: A *low-involvement* honeypot running on different operating systems
- **BackOfficer Friendly** by M. Ranum and A. Lambeth: A small little toy which can be used as a honeypot
- **Home grown honeypots**: Honeypots developed by individuals as experimental solutions

5. INTRUSION DETECTION SYSYTEM

IDS plays a very important role in the deployment of honeypots. As the name already says, an intrusion

detection system is used to detect intrusions or possible intrusions into an observed environment. Different types of IDSes exist, which use different methods to detect intrusions in various environments. Two possible places to implement an intrusion detection mechanism:

Network based intrusion detection: Network intrusion detection systems listen to network communications. They recognize intrusions, which come through the networking environment. Basically a network intrusion detection system (**NIDS**) is a service, which listens on a network interface looking for suspicious traffic. They are mostly signature based.

Host based intrusion detection: Host intrusion detection systems (**HIDS**) reside on a resource, which they supervise. This resource is mostly a computer server or workstation. HIDS look at generated log files, changes in the file system or check for changes in the process table.

On each place, different mechanisms for detecting intrusions are applicable:

Signature based intrusion detection: Signature based intrusion is based on signatures of known attacks. These signatures are stored and compared against events or incoming traffic. If a pattern matches, an alert is generated.

Anomalies based intrusion detection: Anomaly based intrusion detection systems base their decisions on anomalies, things that do not normally occur. If a user suddenly starts a new program he never used or logs in to a machine at 4 o'clock in the morning (what he never did before), the system generates an alert announcing that something isn't running as usual.

Snort: Snort is a freely available intrusion detection system, which can be distributed and modified under the GNU General Public License. Snort can be run in one of three different modes:

- **Sniffer Mode:** In this mode, Snort is used as a packet sniffer and can be configured to show only IP headers or the IP payload as well.
- **Logger Mode:** All packets can be logged to a file and inspected at a later time.
- **Intrusion Detection Mode:** The main mode of Snort. All packets are compared to a database of signatures. If one matches, the packet gets logged and an alert can be sent.

Snort already comes with a large repository of signatures (around 800 signatures). The signature language is easy to learn and writing one's own rules isn't much of a problem.

Snort has the ability to load additional plugins. In the tested version there were already a few plugins supplied, which enabled brute force detection of Back Orifice, Portscans, IP defragmentation, TCP stream reassembly and more.

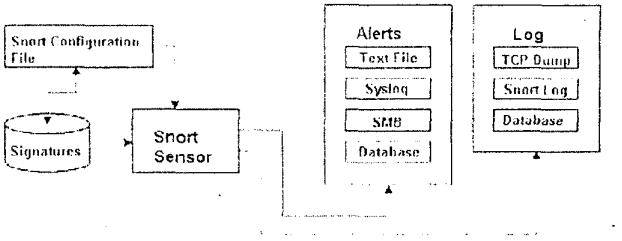

Fig. 2 Snort overview

Snort is only a core IDS engine. There are no supplied analysis toolkits or remote administration Graphical User Interfaces (GUIs). But there are a few available front-ends and analysis toolkits available. One of the best known is ACID (Analysis Console for Intrusion Databases). It is a web-based analysis toolkit that can be used to inspect Snort data (which has to be written into a database).

6. ENHANCING NIDS WITH HONEYPOT

(Developed By Us)

The purpose of the project is to help mitigate risk by adding value to the security measures of an organization. *Mirage* extends Snort, a Network Intrusion Detection System (***NIDS***). Snort detects the intruder and *Mirage* redirects him to the honeypot. At the same time, it also adds him to the hostile IP address list and next time even if the same intruder (IP address) goes undetected by Snort, *Mirage* redirects him to the honeypot. The honeypot attracts and diverts the attacker from their real targets by emulating the real services. The honeypot has been made intelligent enough to emulate not only real hosts but also unused IP addresses in the LAN and provide services on them.

Mirage includes a good Low Interaction Honeypot (i.e. presenting the Bad Guy with emulators of vulnerable programs like ftp, Summarize or capture limited interactions, Simpler to deploy (no system administration), Less likely to be penetrated, more likely to be detected by the Bad Guy). It protects the network assets while still gathering attack data for further analysis. There is also a facility for multiple layers of logging and its analysis is done using ACID (Analysis Console for Incident Databases). ACID is a PHP-based analysis engine to search and process a database of security incidents generated by Snort. This logging to a remote log server makes *Mirage* secure. This is helpful in cases where the blackhats detect the honeypot and try to clear their traces in the logs.

The fig. 3 shows the network configuration of the honeypot and the production hosts.

6.1 Network Design

To ensure isolation, creation of a sub-network within a larger network environment is done. One of the four

computers is configured as a gateway to the network. It serves as a firewall, intrusion detection system. One of them is a honeypot. Another computer is used to collect all logs and store them and the fourth one is the production host that we want to secure. The gateway has three network cards one connecting to the Internet (eth0), (eth1) providing connection between gateway and Production and (eth2) providing connection between gateway and Honeypot. Having a separate gateway for the honeypot helps a great deal in the sense that it helps filter out traffic and makes it easy to monitor/manage any network activity associated with the honeypot. It also provides a secure logging system and gives you better options for securing the production host.

IP range used: 172.16.0.1 – 172.16.0.4
Gateway:

Eth0	: 10.11.1.1
Eth1	: 172.16.0.1
(To Production Hosts)	
Eth2	: 172.16.0.2
(To Honeypot)	
Honeypot	: 172.16.0.25
Production	: 172.16.0.25
Log Server	: 172.16.0.4

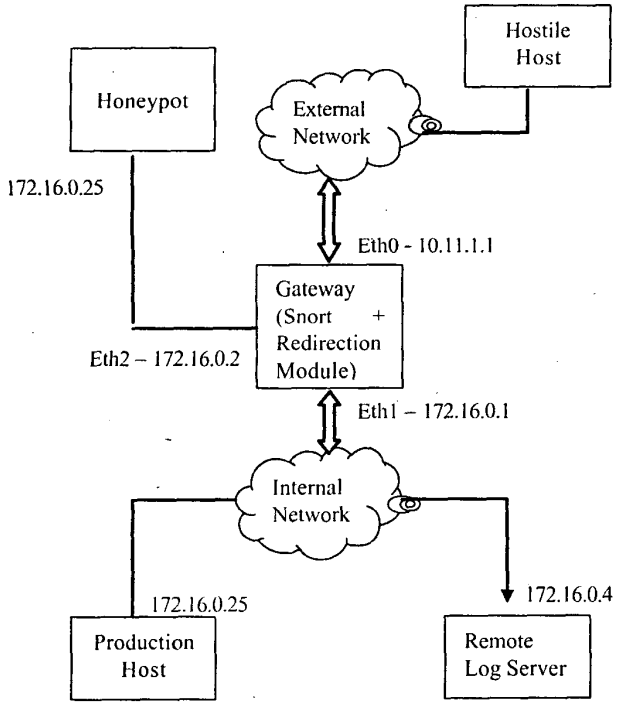

Fig. 3 Network Design

Gateway and Firewall

Redhat Linux 7.2 is installed on the computer that served as the gateway with the basic options to start up the OS and provide network connectivity. Then it is configured with the services and tools required for the setup.

Iptables

Iptables are used to setup the firewall and configure the gateway. While configuring the firewall care should be taken to write rules that make sure of the following

- Setup IP forwarding between the three network interfaces of the gateway
- Avoid spoofing from the internal network. The packets which have source IP as one of that of the internal network should be allowed to go outside.
- Restrict any traffic coming from the honeypot to the gateway.
- Allow minimal but necessary traffic from the Internet to reach the gateway.
- Restrict the possibility of DoS attacks from the honeypots.

Alerting Tools

The gateway and honeypot runs cron jobs (scheduled jobs) that email all the logs from the honeypot on an hourly basis.

7. WORKING OF PROJECT

Mirage constitutes the following modules:

- **Redirection Module:**
  This module redirects an established connection from a real server (e.g. FTP, Telnet, and HTTP) to a honeypot (which has fake services like http, telnet, and ftp) when Snort alerts that the connected user is an attacker. Normal TCP/IP routing between the production host and the external host is being purposely broken. Source routing and packet marking is used to achieve the desired effect of redirecting the broken connection to the honeypot. Packet marking is implemented for specific time period only.

- **Honeypot Module:**
  The Honeypot does the following functions:
  1. Emulates services – File Transfer Protocol (FTP), Hyper Text Transfer Protocol (HTTP) and Telnet. The known vulnerabilities in existing standard services have been avoided in the fake services written by us.
  2. Emulates used as well as unused IP addresses and provide services on these addresses i.e. emulates entire production LAN on a single host.
  3. Logs the user activities.
  4. Generates alerts and mails for administrator
  5. Checks whether the system is compromised, rootkits are installed and if backdoors are open, then it recovers the system back to the safe state.

Logging Module:

Honeypot can be used to capture as much information as possible about the intruder since that explains us how the system was compromised. Hence *Logging module* plays a very important role. The different logging layers that we have implemented are illustrated below:

- Ethereal
- Syslog
- Snort
- Tripwire
- Bash keystroke logging

The First Layer – Ethereal

The first layer of logging uses a packet sniffer on the gateway: the program of choice was Ethereal. This software captures every part of a packet as it passes through the gateway. Ethereal helps us see all the commands issued from the hacker to the honeypot once it was hacked, the only potential problem is if the hacker uses secure shell (ssh) to connect, then the packets are encrypted. Ethereal will record all packets going through the gateway in both directions and also all packets sent between the honeypot and the syslog server.

The Second Layer – Syslog

The second layer of logging is syslog on the honeypot itself. If the honeypot is compromised then the intruder might erase the logs or modify them to remove his activities. So the syslog on the honeypot is made to send the logs to a remote server. There was a problem though, if the hacker got in and looked at the syslog.conf file carefully and saw it was logging remotely to another machine they would very quickly realize something was up. The syslog package from the honeypot was removed; we then downloaded the source and re-installed it and this time changing syslog so it read a different config file before re-compiling it. The original configuration file was left in the same place /etc/syslog.conf, but the actual file syslog was made to read from a different file. Syslog monitors each and every activity the intruder does on the honeypot.

The Third Layer – Snort

The third layer is Snort, very easy to install from rpm and configured in a matter of minutes. Snort acts as a packet sniffer and IDS system depending on how you run it. Snort is run so that it would record every packet entering the network and log them to /var/log/snort on the snort server. Like Ethereal, Snort would give the IP address of the attacker, port numbers time etc. Snort also acts as an IDS system so it compares the packets to well known attacks and generates an alert file if any attacks are recognized as malicious. Like Ethereal snort would capture all traffic entering and leaving the network and all traffic between the honeypot and the syslog server.

The Fourth Layer – Tripwire

The fourth method of logging is to install Tripwire on the honeypot. Tripwire works by building a database of files on the system. When a tripwire check is run it compares the files on the system to a database and it generates alerts if there have been changes to any of the files. This is excellent for a honeypot as it gives the administrator an idea of what files the hacker has changed/trojaned after they hack the system. The database and the config file is saved to a floppy disk and removed from the system. This would prevent a hacker from modifying or corrupting the data.

The Fifth Layer -- Bash Keystroke Logging

The fifth and final method of logging is to patch the bash shell on the honeypot. The patch is applied to a clean version of bash shell. All the other shells are linked to bash. Once applied it then logs all keystroke commands issued on the honeypot to syslog which is then sent to the remote syslog server. This provides details of all the blackhats commands once he has compromised the system. This is very useful as it allows you to track commands even when the blackhat uses Ssh. Ethereal or Snort would not be able to record the commands if ssh was used as it encrypts the packets.

Five layers of logging might seem like overkill but it lets you build up a better picture of what's going on and introduces some redundancy. It's all about recording as much reliable information as possible to assist you in working out exactly how the honeypot was compromised and the activities after.

- **Recovery Module:**
  It checks for infected binaries and worms every hour and replaces infected binaries with original binaries. It also does reporting by generating alerts mails to the administrator

8. How the Modules Work Together

1. Snort monitors packets on the network.
2. On detection of an attack based on the snort-rules, the snort plug-in developed by us provides input (IP address) to the redirection module. This IP address is considered as hostile.
3. The redirection module redirects any further requests from the hostile IP address to the honeypot.

4. The honeypot provides services like FTP, Telnet, and HTTP, which are developed by us. These services give responses to the intruder, which appears to be authentic.
5. The honeypot is made intelligent enough to emulate all the IP addresses (in the internal LAN) for which connections are requested.
6. The honeypot also emulates the unused IP addresses in the internal LAN because all the connections to these addresses are considered as hostile activities
7. The Logging module does all types of logging as described above.
8. Honeypots are expected to be attacked and compromised. The general strategy of any intruder is to install rootkits (backdoors). The recovery module checks for system integrity by calculating the MD5 checksum and compares it with the original MD5 checksum. If the checksums differ, it indicates that system has been compromised. The Recovery module takes the system to a safe state by replacing the infected files by the original ones.

9. CONCLUSION

Honeypots are a new field in the sector of network security. Currently there is a lot of ongoing research and discussions all around the world. Several companies have already launched commercial products. A comparison of available products showed that there are some usable low- to high-involvement honeypots in the market. In the sector of research honeypots, self-made solutions have to be developed as only these solutions can provide a certain amount of freedom and flexibility, which is needed to cover a wide range of possible attacks and attackers. Each research honeypot normally has its own goals or different emphasis on the subject. Developing a self-made solution needs a good technical understanding as well as a time intensive development phase.

A honeypot is a valuable resource, especially to collect information about proceedings of attackers as well as their deployed tools. No other mechanism is comparable in the efficiency of a honeypot if gathering information is a primary goal, especially if the tools an attacker uses are of interest. But nevertheless, honeypots cannot be considered as a standard product with a fixed place in every security aware environment as firewalls or intrusion detection systems are today. Installing and running a honeypot is not just a matter of "buy and go". The involved risk and need for tight supervision as well as time intensive analysis makes them difficult to use. Honeypots are in their infancy and new ideas and technologies will surface in the next time. At the same time as honeypots are getting more advanced, hackers will also develop methods to detect such systems. A regular arms race could start between the good guys and the Blackhat community.

10. REFERENCES

[1] Lance Spitzner, Sun Microsystems, GESS security team, Honeypots, Value and Definitions, http://www.enteract.com/~lspitz, May 2002

[2] Reto Baumann and Christian Plattner, Honeypots, February 2002

[3] Marty Roesch and David Dittrich, Snort, An open source intrusion detection system, http://www.snort.org

[4] Maximum Security, Anonymous, Macmillan Publications, 3rd Edition.

[5] Honeypots, Hack Expo 2002

[6] Sweet Temptation, Eweek, September 2001

[7] Bill Cheswick, "An Evening With Berferd", January 1991

[8] The World of Honeypots, Rick Johnson, ITworld, November 2001

[9] Mark Cooper, member of Distributed Honeynet Project, Baby Steps with a Honeypot, http://www.lucidic.net/whitepapers/mcooper-4-2002.html, April 2002

[10] http://project.honeynet.org/papers/honeynet /bash.patch

[11] Stephen Holcroft, "Design of a Default RedHat Server 6.2 Honeypot", http://www.lucidic. net/whitepapers/sholcroft-4-2002.html, 2002

[12] The Honeynet Project http://www.project.honeynet.org