IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661 Volume 3, Issue 1 (July-Aug. 2012), PP 28-34
www.iosrjournals.org

Design & Implementation of Honeyd to Simulate Virtual Honeypots

[1]Bhumika, [2]Vivek Sharma
[1,2]*Department of Computer Science & Engineering*

Abstract:*Honeyd is an application which enables the setup of multiple virtual honeypots on a single machine, each with different characteristics and services. It is a nice little tool which can be perfectly used to setup a low to mid involvement honeypot. The possibility to generate different virtual honeypots on one machine with even different simulated operating systems enhances the usability of this tool even further. It's great for simulating victims and collecting a lot of interesting information. Honeyd could be used as an early warning system in a productive environment to catch some attacks and trigger an alert. Finding infected web or MS SQL servers can be achieved. The mechanism of attaching a script to a certain port allows a very flexible setup with unlimited capabilities and opportunities for tuning. Finally, the logs generated as well as attacks came on virtual systems can be analyzed graphically using scripts and configuration files.*
Keywords: *Honeyd, honeypots, snort, virtualbox,*

I. INTRODUCTION

Honeyd is an application which enables the setup of multiple virtual honeypots on a single machine, each with different characteristics and services. Honeyd is a small daemon that runs both on UNIX-like and Windows platforms. It is used to create multiple virtual honeypots on a single machine. Entire networks can be simulated using honeyd. Honeyd can be configured to run a range of services like FTP, HTTP, or SMTP. Furthermore, a personality can be configured to simulate a certain operating system. Honeyd allows a single host to claim as many as 65536 IP addresses. Honeyd emulates operating systems by responding with appropriate packets to Nmap and Xprobe fingerprinting packets. Thus the list of operating systems that honeyd emulates can be found in *nmap.prints* and *xprobe2.conf*. But let's start from the beginning i.e. the honeypot technology before going to the details of honeyd.

II. HONEYPOT TECHNOLOGY

Honeypots aren't something that new – the basic idea of a honeypot is quite old and was used already for quite a long time. Although, the word "Honeypot" is a new one and the
technology is getting more and more important. But let's first have a look at a possible definition of what a honeypot is
"A honeypot is a resource which pretends to be a real target. A honeypot is expected to be attacked or compromised.

A. *Level of Interaction*

The first category is installation and configuration effort, which defines the time and effort in installing and configuring the honeypot. In general, if the level of interaction between the user and the honeypot is more than the effort required to install and configure the honeypot is also significant.

The second category is deployment and mainteance. This category defines the time and effort involved in deploying and maintaining the honeypot. Once again, the more functionality provided by the honeypot, the more is the effort required to deploy and maintain the honeypot.

The third category is information gathering which means how much information can the honeypot gain on attackers and their activities? High-interaction honeypots can gather vast amounts of information, whereas low-interaction honeypots are highly limited. Finally, level of interaction impacts the amount of risk introduced. The greater the level of interaction, the more functionality provided to the attacker and the greater the complexity. Combined, these elements can introduce a great deal of risk. On the other hand, low-interaction honeypots are very simple and offer a little interaction to attackers and thus a very little risk is associated with them.

Degree of involvement	Low	Medium	High
Installation and configuration effort	Easy	Medium	Difficult
Deployment and maintenance effort	Easy	Medium	Difficult
Information Gathering	Limited	Medium	Extensive
Level of Risk	Low	Medium	High

Table 1: Tradeoffs between Honeypot Levels of Interaction

III. HONEYD: A LOW INTERACTION HONEYPOT IN ACTION

Honeyd is a tool which simulates virtual hosts on a network, and is actively used in Honeynet research today; it's a thin daemon with lots of interesting features. It can assume the personality of any operating system, and can be configured to offer different TCP/IP "services" like HTTP, SMTP, SSH etc. Honeyd is used in honeynet research typically for setting up virtual honeypots to engage an attacker. One useful feature of Honeyd is its ability to simulate an entire network topology within one machine – with multiple hops, packet losses and latency. This lets us simulate complex networks in test labs; it could also present a make-believe network to an attacker who gets snared in a honeynet.

Some of the features available in Honeyd for simulating networks are:

- Simulation of large network topologies
- Configurable network characteristics like latency, loss and bandwidth
- Supports multiple entry routers to serve multiple networks
- Integrate physical machines into the network topology
- Asymmetric routing
- GRE tunneling for setting up distributed networks

Another new concept is that Honeyd emulates an operating system not only at the application level but at the IP stack level as well. For example, if one select his Honeyd system to emulate a Windows NT 4.0 Server SP5-SP6 server, it not only emulates the services, such as an IIS Web server, but also the IP stack. In this case, if one use Nmap to fingerprint the IP stack, the response will be that the IP stack is Windows NT 4.0 Server SP5-SP6.

A. Setup details and Implementation

In our implementation, we have implemented the personality of various OS as virtual Honeypots. Using honeyd, we can also simulate the large network topologies by writing scripts. In order to catch and categorize attacks, we installed Snort in its latest version as well as Mysql for logging all snort events to the database for easier and less time consuming analysis of collected attacks.

In this section, we discuss the design and implementation of Honeyd. Honeyd, developed by Neil Provis, is implemented as a Unix daemon that runs on a workstation and listens to network traffic.

B. Proposed Design Overview

To implement the virtual Honeypot approach, we need the components listed below.

- An active probing tool, such as Nmap.
- A passive fingerprinting tool, such as P0f and Snort.
- A low interaction honeypot used to simulate networks, such as Honeyd.
- A collection of physical honeypots to receive redirected traffic. These honeypots are considered as a small but representative sample of the most used operating systems on the network.
- A Database, containing hosts' description, and log information.
- A Virtual software to implement redhat Linux.
- An administrator's interface to configure dynamic honeypot server in real time and view reports.

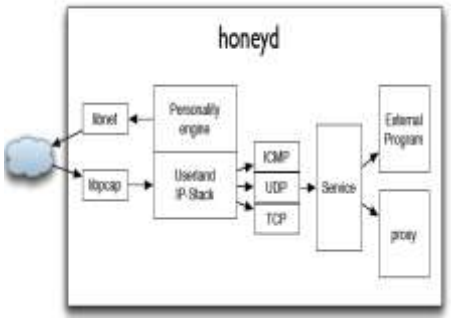

Fig. 1 Honeyd: A Low Interaction honeypot

A. Experimental Details

The experimental setup of our work is shown in the figure 2. Initially we assign a static IP addresses to the window OS, virtual machine. We assign two unused IP address to two virtual honeypots respectively. We assign 192.168.1.243 IP address to virtual honeypot1 and 192.168.1.244 IP address to virtual honeypot2.These virtual honeypots corresponds to two different personalities as described in configuration file of honeyd named honeyd.conf. The honeypot designed is proposed as a Low Interaction Honeypot which is integrated with Snort and MySQL.

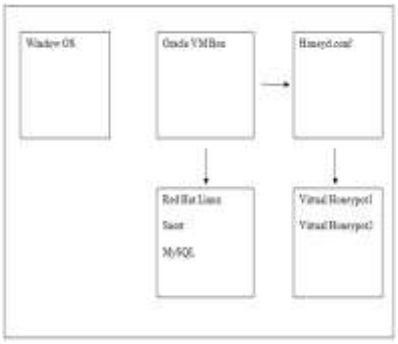

Fig.2 Experimental Setup

B. Oracle VM Virtual Box

Virtual Box is a cross-platform virtualization application. It can be installed on existing Intel or AMD-based computers, whether they are running Windows, Mac, Linux or Solaris operating systems. It also extends the capabilities of an existing computer so that it can run multiple operating systems (inside multiple virtual machines) at the same time. For example, you can run Windows and Linux on your Mac, run Windows Server 2008 on your Linux server, run Linux on your Windows PC, and so on, all alongside your existing applications. Virtual Box is deceptively simple yet also very powerful. It can run everywhere from small embedded systems or desktop class machines all the way up to data center deployments and even Cloud environments.

C. Procedure

- We start our work by writing personalities of virtual honeypots. We can write as many personalities in honeyd.conf file. We have written two different personalities in our configuration file.
- Then we write the command: *arpd –d –i eth0 "IP addresses in configuration file of honeyd"*
  Here we use the -d so that it does not run as a daemon or as a background process. Arpd daemon uses the free address space and allocate the Mac address of the machine with each IP address.
- After this honeyd is started by writing following command: *honeyd -l /log/test –f honeyd.conf (IP addresses in configuration file of honeyd).* -f is used so that honeyd runs in foreground. By this command honeyd is started as background process.
- Then the MySQL database is started. In new tab snort is started by the command: *snort -i eth0 –vc /root/Desktop/honeyd/Latest_snort etc/snort.conf –l/var/log*
- Honeydsum.pl is used to generate a report from the packets logged by the honeyd. This is done by following command: *./honeydsum.pl –c honeydsum.conf/root/Desktop/honeyd/log/test>report*
- Snortalog.pl is used to generate a report (graphically) from the alerts generated by snort. This is done by the following command: *perl snortalog.pl –file /var/log/alert –r N 30 – report*

IV. HONEYD: IMPLEMENTATION DETAILS

First install the required libraries.

a. Libevent
b. Libdnet
c. Libpcap: used libpcap 0.7 and its dev file.
d. Libpcre

A "Honeyd.conf" file is configured which contains virtual operating systems with emulated services. Each system is designed with the following commands.

- Create command create a new system.
- Add command used to add services, therefore binding scripts to a certain port.
- Set command assigns personality to a created system.

Install Honeyd
Downloaded honeyd-1.5c from http://www.citi.umich.edu/u/provos/honeyd/honeyd-1.5c.tar.gz

1) *Running ARPD:* The fake IP addresses are created with the help of Arpd daemon, which further bind these IP addresses with different templates specified in "honeyd.conf" configuration file. In this file The templates are created which are nothing but completely fake script modules for the different operating systems. Some fake instances of different operating systems are created with fake IP addresses and then bind the templates accordingly.

An Arpd ,also called farpd ,daemon can be installed which monitors the allocated IP address space and for any IP address that no host respond with MAC address , farpd will respond with the Honeyd physical machine MAC address.

2) *Starting honeyd:* Honeyd is a small daemon that runs both on UNIX-like and Windows platforms. It is used to create multiple virtual honeypots on a single machine. Entire networks can be simulated using honeyd. Honeyd can be configured to run a range of services like FTP, HTTP, or SMTP. Furthermore, a personality can be configured to simulate a certain operating system.

Sample Honeyd Configuration File

```
create windows
set windows personality "Windows XP home edition"
add windows tcp port 80 "perl scripts/iis-0.95/iisemul8.pl"
add windows tcp port 139 open
add windows tcp port 137 open
add windows udp port 137 open
add windows udp port 135 open
set windows default tcp action reset
set windows default udp action reset
bind 192.168.1.243 windows
```

A sample honeyd configuration file named honeyd.conf is written above. In this file IP address 192.168.1.243 is assigned to virtual honeypot1.Similarly IP address 192.168.1.244 is assigned to the virtual honeypot2.Then the packets are logged in a file named test. The command is

Honeyd –l/log/test –f honeyd.conf 192.168.1.243-192.168.1.244

Fig. 3 Honeyd: A Low Interaction honeypot

3) *Starting MySQL services:* MySQL is an Open Source Standard Query Language (SQL) database that is fast, reliable, easy to use, and suitable for applications of any size. MySQL can easily be integrated into Perl programs by using the Perl DBI (Database Independent interface) module. DBI is an Application Program Interface (API) that allows Perl to connect to and query a number of SQL databases.MySQL is used here to store the alerts generated by Snort. The command used to start the MYSQL is

service mysqld start;

4) Starting Snort: Snort is a libpcap-based packet sniffer/logger which can be used as a lightweight network intrusion detection system. Snort has three primary uses: It can be used as a straight packet sniffer like tcp dump, a packet logger or as a full blown network Intrusion prevention system. Snort is used here to generate alerts and to capture more information about the attacks.

Command used to start snort is

Snort -i eth0 –vc etc/snort.conf –l/var/log.

After this command alerts generated by snort are stored in the file called log situated in var folder.

After initializing the snort we use sql query to display the number of alerts generated by snort in descending order. The query is

Select s.sig_name,count(*)as count from event e,signature s where e.signature=s.sig_id group by e.signature order by count desc;

```
mysql> select s.sig_name,count(*) as count from event e,signature s where e.signature=s.sig_id gr
oup by e.signature order by count desc;
+-----------------------------------------------------------------------------------+-------+
| sig_name                                                                          | count |
+-----------------------------------------------------------------------------------+-------+
| ICMP Echo Reply                                                                   |   586 |
| ICMP PING                                                                         |   157 |
| ICMP PING Windows                                                                 |   148 |
| ICMP Destination Unreachable Port Unreachable                                     |    63 |
| P2P BitTorrent transfer                                                           |    43 |
| (http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE            |    37 |
| Reset outside window                                                              |    19 |
| ICMP PING BSDtype                                                                 |    17 |
| ICMP PING *NIX                                                                    |    17 |
| (http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE                               |     2 |
| SPYWARE-PUT Trackware funwebproducts mywebsearchtoolbar-funtools runtime detection |     1 |
+-----------------------------------------------------------------------------------+-------+
```

Fig. 4 No. of alerts generated by snort

5) *Starting Honeydsum:* It is a script written in Perl designed to generate a text summary from Honeyd logs. The summaries may be produced using different parameters as filters, such as ports, protocols, IP addresses or networks. It shows the top source and port access and the number of connections per hour, and supports input from multiple log files. The script can also correlate events from several honeypots. The output of this script contains the information that provides several different summaries of all the traffic captured by the whole Honeyd environment. The command to generate a report from the honeydsum is:

./honeydsum.pl –c honeydsum.conf /root/Desktop/honeyd/log/test>report.

It takes the packet logged by honeyd as input and generates the report.This report tells the total number of connections on the honeypot and the IP addresses of the sources which attacked on the virtual honeypots.

V. RESULTS

It is not easy and intuitive for everyone to query MySQL database in order to find some information about the honeypot's runs so to facilitate easy view and searching of data in the MySQL database.

Honeydsum is a tool written in Perl designed to generate a summary from honeyd logs. The summaries may be produced using different parameters as filters, such as ports, protocols, IP addresses or networks. This tool is basically used for processing of honeyd log files to get the reports. You can say it is being used for report generation by processing honeyd log files.

Further, the results produced by the two honepots are analyzed graphically.

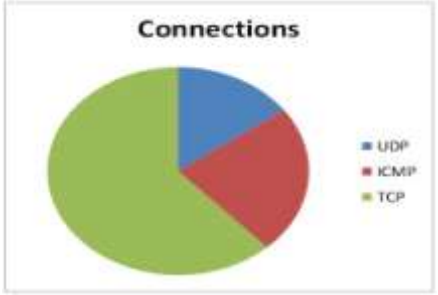

Fig 5. Total no of packets received (results generated from honeyd logs by honeydsum.pl)

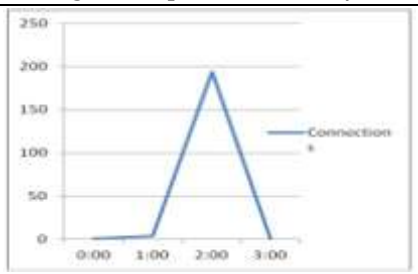

Fig 6 Total No. Of Connections established per hour (created from honeydsum.pl)

After performing the signature command using MYSQL as in fig 4, maximum alerts are ICMP attacks. This proves that out of top 10 source hosts, first attacker performs ICMP attack. This proves the result.

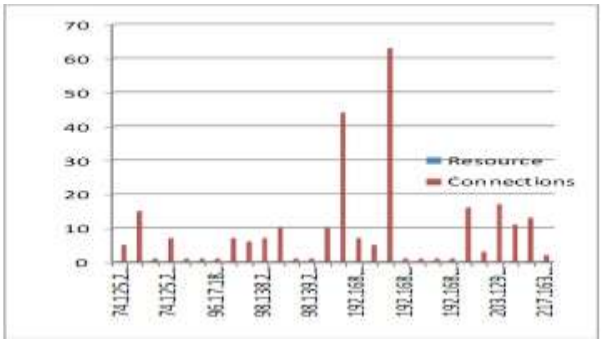

Fig7 Resources at Virtual Honeypot2 by different IPs

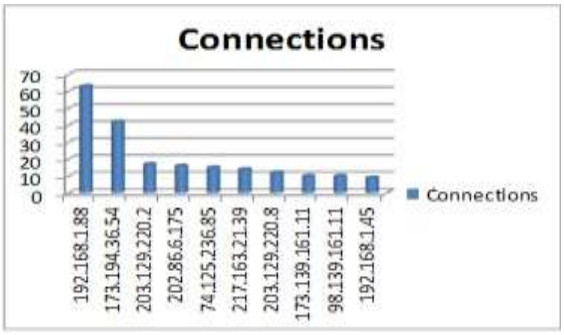

Fig8 Top 10 source hosts

VI. CONCLUSIONS

As per the research done so far and literature study, Honeypots are a current research field in the sector of network security. Currently there is a lot of ongoing research and discussions all around the world. Honeypots are resources which give full freedom to attackers and every activities of the attacker will get logged. Honeypots gives us the platform to gather the intelligent information about the attackers so that we can later study them and can take the remedial actions to tighten the network security. Honeypots are being classified into different categories such low interaction honeypots, high interaction honeypots etc. Low interaction honeypots provide the emulated environment to the attackers which is not a real environment therefore the volume of attacks may be less in case of low interaction honeypots whereas in case of high interaction honeypots which provide the real environment to the attackers to get logged. 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. The involved risk and need for tight supervision as well as time intensive analysis makes them difficult to use. Honeypots are in one's 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.

REFERENCES

[1] "What Is the Difference: Viruses, Worms, and Trojans?", http://www.cisco.com/web/about/security/intelligence/virus-worm-diffs.html..

[2] Provos, N., "A Virtual Honeypot Framework", SSYM'04 Proceedings of the 13th conference on USENIX Security Symposium, Volume 13, 2004.

[3] Lance Spitzner, Tracking Hackers. Addison Wesley, September 2002.

[4] Zanoramy, W. and Zakaria, A., "Deploying Virtual Honeypots on Virtual Machine Monitor" IEEE Proceedings, vol. 4 , pp.1-5, 26 Aug 2008.

[5] Spitzner, L., "Honeypot: Definitions and Values", May 2002 http://www.spitzner.net.

[6] Levin, J. and Labella, R., "The Use of Honeynets to Detect Exploited Systems across Large Enterprise Networks", IEEE Proceedings, pp.92-99, 18 June 2003.

[7] Qassrawi, M. and Hongli, Z. "Deception methodology in virtual Honeypots", Second International Conference on Network Security, Wireless Communication and Trusted Computing, 2010.

[8] Bao, J. and Gao, M. "Research on network security of defense based on Honeypot", International Conference on Computer Applications and System Modeling, 2010.

[9] Levine, J. and Grizzard, J. "Using honeynets to protect large enterprise networks," Security & Privacy Magazine, IEEE, vol. 2, pp. 73-75, 2004.

[10] Ryan Talabis," Honeypots 101: What's in it for me?"http://www.philippinehoneynet.org/, Fetched 21/06/2011.

[11] Tang, X. "The Generation of Attack Signatures Based on Virtual Honeypots", International Conference on Parallel and Distributed Computing, Applications and Technologies, 2010, pp.435-439.

[12] V.Maheswari, Dr. P. E. Sankaranarayanan, "Honeypots: Deployment and Data Forensic Analysis", International Conference on Computational Intelligence and Multimedia Applications 2007, 0-7695-3050-8/07, 2007 IEEE.

[13] Chao-Hsi Yeh and Chung-Huang Yang, "Design and Implementation of Honeypot Systems Based on Open-Source Software", 1-4244-2415-3/08/. 2008 IEEE.

[14] Zhang Li-juan**, "**Honeypot-based Defense System Research and Design", 978-1-4244-4520-2/09/, 2009 IEEE.

[15] Xinyu Tang, "The Generation of Attack Signatures Based on Virtual Honeypots", The 11th International Conference on Parallel and Distributed Computing, Applications and Technologies, 978-0-7695-4287-4/10, 2010 IEEE.

[16] Abhay Nath Singh, R.C.Joshi, "A Honeypot System for Efficient Capture and Analysis of Network Attack Traffic", International Conference on Signal Processing, Communication, Computing and Networking Technologies, 978-1-61284-653-8/11, IEEE.

[17] Abdallah Ghourabi, Tarek Abbes, Adel Bouhoula, "Design and Implementation of Web Service Honeypot", 1078-0-8495-4287-4/11, 2011 IEEE.

[18] Meixing Le, Angelos Stavrou, Brent ByungHoon Kang, "DoubleGuard: Detecting Intrusions in Multitier Web Applications", IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 9, NO. 4, JULY/AUGUST 2012, 1545-5971/12/, IEEE.

[19] Liu Dongxia & Zhang Yongbo, "An Intrusion Detection System Based on Honeypot Technology", 2012 International Conference on Computer Science and Electronics Engineering, 978-0-7695-4647-6/12, 2012 IEEE

[20] "Snort", http://www.snort.org/assets/166/snort_manual.pdf.

[21] John E. Canavan," Fundamentals of Network Security" , http://www.artechhouse.com .

[22] Provos, N. "A Virtual Honeypot Framework", SSYM'04 Proceedings of the 13th conference on USENIX Security Symposium, Vol. 13, 2004.

[23] Lance Spitzner, "Definitions and Value of Honeypots", http://www.trackinghackers.com/papers/honeypots.html.

[24] Peter, E. et al., "A Practical Guide to Honeypots", http://www.cse.wustl.edu/~jain/cse571-09/ftp/honey/index.html, Fetched 20/06/2011.

[25] Spitzner, L." Know Your Enemy: Honeynets", http://www.honeynet.org/papers/honeynet.