IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, **p**-ISSN: 2278-8719
Vol. 3, Issue 4 (April. 2013), ||V3 || PP 25-28

Network Forensic Based on Honeynet

Deepali Kansal, Navdeep Singh Sethi
*Research Scholar, Department of Computer Science, Adesh Institute of Engineering and Technology, Faridkot-151203
**Assistant Professor, Department of Computer Science, Adesh Institute of Engineering and Technology, Faridkot-151203

Abstract: In this paper we design a Honeynet Network Architecture which combines the both Low and High interaction Honeypots. In this architecture, low interaction honeypot play the role of a gateway to high interaction honeypot and filter out incoming traffic. This permitted us to provide comprehensive statistics about threats, collect high level information about attacks, and monitor the activities carried out by different kind attackers, capture network traffic and storing on our base operating system for further forensic.

I. INTRODUCTION

In the ever changing world of global data communications, inexpensive Internet connections, and fast-paced software development, security is becoming more and more of an issue. Security is now a basic requirement because global computing is inherently insecure. As your data goes from one computer to Next computer on the Internet, for example, it will pass through several other points along the way, giving other users the opportunity to intercept, and even alter it. So the security of data and the system is the main issues. It does nothing to protect your data center, other servers in your network, or a malicious user with physical access to your EnGarde system. By learning the about tools and techniques used by intruder, we can secure our IT assets and infrastructure. Honeypots provide information about black-hat techniques and tactics by which they have been able to gain illegitimate access to system resources. In last several years, the functionality and use of computer systems have increased vastly, which leads to an increasing complexity of these systems. An attacker can use such vulnerabilities to gain remote access, granting partial or full control over these computer systems. The attacker can initiate the attack from any arbitrarily system connected to the Internet that is already under the control of the attacker. The Honeypot system has no production value and has no authorized activity. Thus any interaction with the Honeypot is most likely the result of malicious intent. Honeypots do not provide security but provide data and knowledge that aids the system administrator in enhancing the overall security of their network. This knowledge about the threat can be used as input to any early warning systems. Over the years, researchers have successfully isolated and identified worms and exploits using Honeypots placed in specialized architectures called Honeynets. They give insight into attacks and attackers, their skill level, their organization as groups or individuals, their motives and tactics, and thus, almost every aspect is logged and can be made auditable.Virtualization software helps reducing the cost ownership of the IT infrastructure of organization. Virtualization Technique like VMware™ [1] provides the flexibility to create a specialized network of hosts on a single physical machine. In this experiment we use free and Open Source tools and technologies that run on a Linux platform. Linux based operating systems have been used as the host OS and for guest OS virtual machines. This includes a Linux based Honeypot and a Honeywall gateway. We also use sebek as a data capture tool [2]. We can enhance the overall security of our network resources after study the results.

II. BACKGROUND

A. Honeypots

Honeypots is a technology that is rapidly maturing and establishing this counter measure as viable and useful in modern network defense. Honeypot focus on the principles of detection, response and monitor. It is not useful to reducing the impact of internal cyber attacks in organizations. A Honeypot is generally defined as a network security resource whose value lies in it being scanned, attacked, compromised, controlled and misused by an attacker to achieve his malicious goals. Lance Spitzner defines Honeypots as "A Honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource" [3]. Honeypots can be classified into two categories based on their level of interaction:

Low-interaction Honeypot: Simulate the services frequently requested by attackers. Since they consume relatively resources, multiple virtual machines can easily be hosted on one physical system, the virtual systems have a short response time, and less code is required, reducing the complexity of the security of the virtual systems. Example: Honeyd [4].

High-interaction Honeypot: In High Interaction honeypot, the server and host system have real operating system and application. It Provide the freedom to Attacker to interact with a real operating system. When they do, they do not know they are within honeynet and their every activity is logged and monitor for further analysis. Honeynets can do this with the help of Honeywall gateway. This gateway allows inbound traffic to the victim systems, but controls the outbound traffic using intrusion prevention technologies.
The individuality of the Honeypot is decisive and we can observe that the learning curve (from the attacker) is directly proportional to the level of stealth for the Honeypot [3, 5, 6, &7].

B. Honeynet

Combination of one or more honeypots on a network form is called honeynet. A Honeynet is a special kind of high-interaction Honeypot. Honeynets increase the value of single honeypot to a highly controlled network of Honeypots. A honeynet can provide information to the system administrator with intelligence about vulnerabilities and compromises within the network. A honeynet is a highly controlled network where every packet entering or leaving is monitored, capture and analyzed.

Data Control: It involves protecting non-honeynet systems that an intruder might target from a compromised honeypot. It prevents the attacker to use the honeynet to attack other non-honeynet system either by mistake or intentionally, even when the honeynet itself is compromised.

Data Capture: The main goal of data capture is to gather data and details as possible from the attacker action by monitoring and logging of all threats and hackers activities within the Honeynet. All activities must be captured including activities on the network, against the honeynet host and also those activities that originate within the honeynet.

Data Collection: Collecting all the capture data by different honeynets in a distributed environment to a central location. This honeynet architecture makes a highly controlled network, in which one can control and monitor all kinds of system and network activity. Honeypots are then placed within this network..

C. Virtual Honeynet

Virtualization plays an important role in implementing virtual honeynet. Using Virtualization, we can run multiple virtual machines on a single physical machine. An Operating system that running inside of physical and virtual machines are referred to the host and guest virtual machine. In order to achieve virtualization, the host machine should share the CPU and memory resources with the guest virtual machines. Virtualization also reduces the cost of project hardware.

The name virtual is used because all different operating system have the appearance to be running as an independent system [8]. A virtual Honeynet is a complete Honeynet running on a single computer in virtual environment. In this project we have build and tested our own virtual honeynet for attracting the attackers to interact with real operating system, services and programs. The main reason of choosing virtual honeynet was lack of hardware resources thus we decide to use virtualbox [9] as virtualization software and honeywall Roo [10] to deploy the virtual honeynet. For our implementation we installed virtualbox version 3.0.12 and created three virtual machines upon it: honeywall, two honeypots (Windows, Ubuntu).

III. PROPOSED ARCHITECTURE

A. Problem Identification & Solution

During our literature review we decided to use [7] as the template for our work implementation. Using VMware, a bridged interface like vmnet0 has direct access to the physical network interface. The problem is to find malicious activity roaming in LAN environment. So if someone is browsing some web page which is malicious in production environment and if that malicious links drop some files which are malicious, then honeypot is able to detect that dropped files.Thus we developed a framework for capturing the data in production environment by deploying a honeynet system in production environment in which two interfaces will cause a bridge between the same LAN segments resulting in loops in the network.

B. Design Details and Discussion

Similar design problems were being faced and discussed by security researchers all over the globe who wanted to implement a similar virtual Honeynet. We implemented a honeynet architecture which combines the both low and high interaction honeypots. For the low-interaction part we used Nepenthes and for the high-interaction part we implemented a virtual honeynet architecture based on the Virtualbox [9] virtualization software. The low- and high-interaction honeypots are deployed separately, and the backup of the collected

attack data on each host machine of the low and high-interaction honeypots is stored on Base Operating system for further forensic of network packets. The implemented architecture is illustrated in Fig 1.

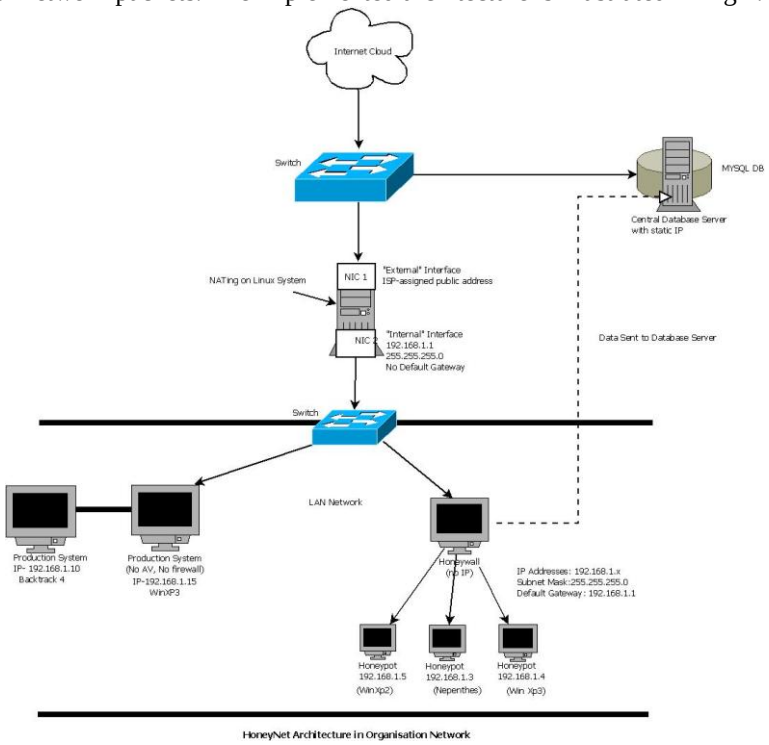

Fig 1: Proposed Design for Organization Network

In our implementation, we used only one physical machines which contain the virtual honeypots as guest OS and a baseOS machine to get all the collection of attack data and to monitor the activities and processes on the honeypots. All of the honeypots are deployed and configured on the virtual machines.
A small daemon that creates virtual hosts on a network is configured to make a virtual router and some virtual operating systems. The honeypot is configured to run arbitrary services also. It can also emulate any TCP or UDP port number with open, closed, or blocked states with simple port listeners. Port applications can be mimicked by installing service scripts and proxies. Anyone can ping and/or trace route the virtual machines simulated with any type of service.

C. Result and Discussion

Our honeypots were online for a period of approximately 90 days from March to May of 2012. During this period we received over 100,000 identified attack connections. These results provide the better insight to the readers about what was observed in our honeypot experiment. During this period our honeypot environment suffered different kind of attacks. Table 1 shows the number of attack connections during the observation period:

Protocol	Connections	Percentage
TCP	88365	95.42%
UDP	482	0.52%
ICMP	3759	4.03%
Total	92606	100%

Table 1: Total connections per protocol

By looking at Table 1, we can see that TCP is the most used protocol by attackers. This can be explained by the fact that multiple service and applications use TCP compare to other protocols. The number of established connections on two honeypots was more than on the other honeypots. It happened because of many open ports and real and emulated services run on those two honeypots. This indicates that those two machines have been continuously scanned by vulnerability scanners and the rate of the connection attempts on the network services HTTP (80), SMTP (25), POP (110), SSH (22) was higher.

Ports 100 and 25: POP and SMTP as emulated services was main target for spammers, and automatic programs in internet.
Ports 80 and 22: HTTP and SSH services with vulnerabilities several times were successfully exploited by attackers.

Attacked Port: The basic attacks we observed from the snort log files were TCP, UDP and ICMP port scans. These scans have been performed by different scanner tools such as NMAP, Portsweep in order to scan multiple hosts for a specific open port. A portscan is used to scan open ports on a single target host.UDP scanning is slower than TCP scanning.

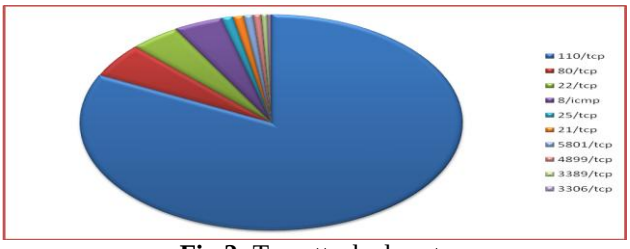

Fig 2: Top attacked ports

Top Attacked Countries: Out of these countries the highest number of attacks came from China and Europe followed by the US. The same proportion also stands for the highest attack frequencies.

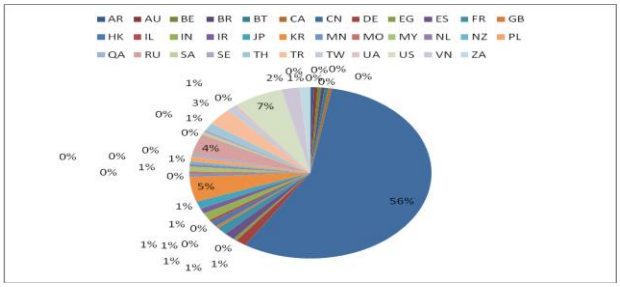

Figure 3: Top Attack Countries

IV. CONCLUSION

The implemented honeynet architecture is used for gathering attack data and tracking the activities carried out by the attackers. Then we have captured and classified the network packets. The aim was to study the attackers detail by using knowledge based on this analysis. It appears that most of the observed attacks were automated and carried out by script kiddies. This work will help organizations to select proper protection mechanism for their networks by evaluating the impact of detected attacks, and taking into consideration the attacker's skill and knowledge level.

REFERENCES

[1]. VMware. (2008) VMware Server 1.0.6 Free.Available:http://www.vmware.com/download/server/. Last accessed 20 Aug 2008.
[2]. Know Your Enemy: Sebek, A kernel based data capture tool, The Honeynet Project, http://www.honeynet.org, Last Modified: 17 November 2003
[3]. Spitzner L. (2002). Honeypots: Tracking Hackers.US: Addison Wesley. pp 1-430Honeyd, http://www.honeyd.org/
[4]. Stoll, C. The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage. Pocket Books, New York, 1990
[5]. Cheswick, B. (1991). "An Evening with Berferd, in Which a Cracker Is Lured, Endured, and Studied." Forum of Incident Response and Security Teams (FIRST).
[6]. T H Project, Know your Enemy. Addison-Wesley, 2nd ed., 2004.[8] Know Your Enemy: Defining Virtual Honeynets. http://old.honeynet.org/papers/virtual [9] VirtualBox. (2004)
[7]. Sun VirtualBox® User Manual. Available: http://www.virtualbox.org/manual/UserManual.html. The Honeynet Project. (2005). KnowYour Enemy: oneywall CDROM Roo. Available:http://old.honeynet.org/ papers/cdrom/Roo/index.html. Last accessed 5 May 2008.