ISSN 2395-1621

System Security with Deceptive Virtual Host

[#1]Kiran Shukla, [#2]Narendra Rakhapasare,[#3]Varsha Pawar ,
[#4]Prasad Patil

[1]Shuklakiran19@gmail.com
[2]friendkingnarendra@gmail.com
[3]varshvrush@gmail.com
[4]patilprasad1256@gmail.com

Computer Engineering Department, Savitribai Phule Pune University
TSSM's Bhivrabai Sawant College of Engineering and Research,India

ABSTRACT

In Existing system one honeypot is protect to the only one server. Therefore this system is increase the complexity between the hardware. Because each and every time dedicate the honeypot for the every server. If more than one server are available then each server having its own honeypot.HoneyD is a tool for simulating computer system on the network layer framework of HoneyD is introduced in this paper. The development concept and the way of working of HoneyD are analyzed. A virtual large scale network is constructed by HoneyD and which includes network delay, network packet loss. Thetopology of virtual network, host operating system and system services are tested. It shows that HoneyD can simulate large scale network successfully.As there are risk associated with deploying any type of honeypot.The design of Honeyd server starts with identifying the risk in the environment in which it will be deployed. Here multiple server is protecting using one virtual honeypot.

***Keywords*— Honey D, NIDS.**

ARTICLE INFO

Article History

Received :26th April 2016

Received in revised form :

28th April 2016

Accepted : 30th April 2016

Published online :

3rd May 2016

I. INTRODUCTION

HoneyD is an open source computer program that allows a user to setup and run multiple virtual hosts on a computer network. These virtual hosts can be configured to mimic several different types of server, allowing the user to simulate an infinite number of computer network configurations.
Three main functions of Honeypot are:

1. Collection: - Can provide us with small sets of high value data.
2. Detection: - Can provide us with early warnings of attacks.
3. Diversion:-Can provide us with more time to fix vulnerabilities by diverting attacks away from real hosts.

There are at least three issues with honeypots that

need to be addressed in order to make the approach effective against worm attacks: 1 . A traditional honeypot only sees the traffic that is directed at it. Ina class A network (e.g. 1 0.0.0.0), a single honeypot would be ineffective because the chance of it being found by a worm that is randomly scanning the network is extremely low. In order to increase our chances, we would need to deploy more honeypots. 2. Honeypots can be difficult and expensive to set-up and run.

If we multiply the effort and expense by the number of honeypots needed to be effective, we quickly find that the approach becomes impractical and prohibitively expensive. 3. Deploying a honeypot can be risky.

II. LITERATURE REVIEW

Intrusion detection system using advanced Honeypots.

As the number and size of the Network and Internet traffic increase and the need for the intrusion detection grows in step to reduce the overhead required for the intrusion detection and diagnosis, it has made public servers increasingly vulnerable to unauthorized accesses and incursion of intrusions. In addition to maintaining low latency and poor performance for the client, filtering unauthorized accesses has become one of the major concerns of a server administrator.
Design and efficient deployment of Honeypot and

Dynamic rule based live Network Intrusion collaborative system.

A Honeypot based Network Intrusion Collaboration System which is capable of generating dynamic rules during any anomalous behavior in the network or a possible intrusion is presented. The NICS designed is a collection of several existing Free and Open SourceSoftware's customized for the specific need that helps in implementing both preventive and detective mechanisms of network security.

Extended honeypot framework to detect old/new cyber Attacks.

To propose an approach to detect the new malicious objects with an optimal cost. Honeypots are generally used to detect the new malicious objects. The available honeypot frameworks are too costly to be afforded by an average organization. Therefore, we are proposing a low cost honeypot framework to detect malicious objects named extended honeypot. The approach is not only cost effective but also better than other approaches in some situations such as in the Intranet which is having more than one LANs and every LAN is having double honeypot.

Design and implementation of Linux based hybrid client honeypot incorporating multilayer detection.

In current global internet cyber space, the number of targeted client side attacks are increasing that lead users to adversaries' web sites and exploit web browser vulnerabilities is increasing, therefore there is requirement of strong mechanisms to fight against these kinds of attacks. In this paper, we present the design and implementation of a client honeypot which incorporate the functionality of both low and high interaction honey client solution and incorporate the multi layer detection mechanisms to fight against client side targeted attacks. Our developed Virtual Box powered Honey client is very useful for collection of internet malwares but it is having a limited capabilities or we can say that it is just a prototype. There is a requirement of integration of crawler as data acquirement, in present system, there is no such component in our developed module. Further there is also a possibility of addition of various client side applications such as Firefox, pdf etc. because currently we only using Internet Explorer for actively visiting the websites. And there is also a possibility of addition of automatically analysis of collected malwares. We can confirm that we cannot cover all the challenges such human user simulation, logic bomb, time triggered websites but we have developed a prototype solution to get better understanding of client honeypots.

III.EXISTING SYSTEM ARCHITECTURE

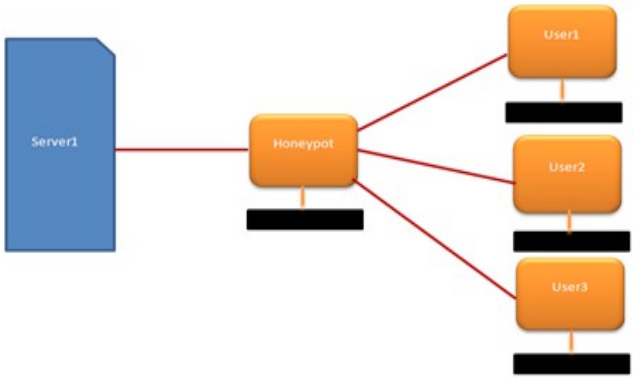

Fig: Existing system

In this system each server have its own honeypot.that means the multiple server having its multiple honeypot. A honeypot is a non-production system, design to interact with cyber-attackers to collect intelligence on attack techniques and behaviors. There has been great amount of work done in the field of network intrusion detection over the past three decades. With networks getting faster and with the increasing dependence on the Internet both at the personal and commercial level, intrusion detection becomes a challenging process. The challenge here is not only to be able to actively monitor large numbers of systems, but also to be able to react quickly to different events. Before deploying a honeypot it is advisable to have a clear idea of what the honeypot should and should not do. There should be clear understanding of the operating systems to be used and services (like a web server, ftp server etc) a honeypot will run.The risks involved should be taken into consideration and methods to tackle or reduce these risks should be understood. It is also advisable to have a plan on what to do should the honeypot be compromised. In case of production honeypots, a honeypot policy addressing security issues should be documented. Any legal issues with respect to the honeypots or their functioning should also be taken into consideration. In this paper we explain the relatively new concept of "honeypot." Honeypots are a computer specifically designed to help learn the motives, skills and techniques of the hacker community and also describes in depth the concepts of honeypots and their contribution to the field of network security. The paper then proposes and designs an intrusion detection tool based on some of the existing intrusion detection techniques and the concept of honeypots.

IV. PROPOSED SYSTEM ARCHITECTURE

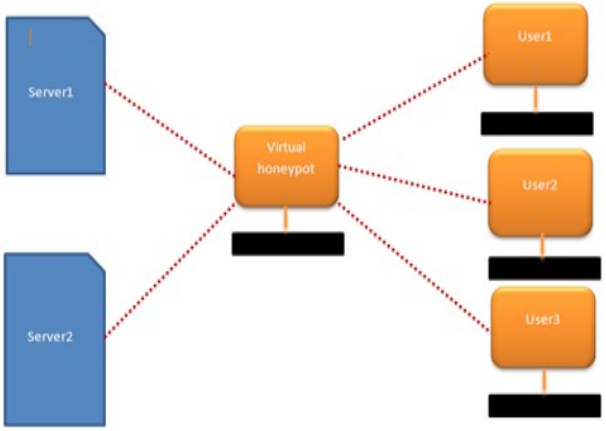

Fig: Deceptive virtual host i.e Virtual Honeypot

In proposed system one virtual honeypot is protect the multiple server. Here complexity between the hardware is minimum. In above fig. one virtual honeypot is prtotecting the sever1 and server2. Also here user1 , user2,user3 are communicat with these two server. Virtual honeypot is work like deceptive system. Which is protecting the multiple server . also it is help to detecting the attacker & hacker. It also crete the log of user. In log user ip address, time ,date & mac address are identify.

V. WORKING

Honeyd operates one level above that by providing a framework to create virtual honeypots that can run any number of services. The Deception Toolkit could be one of the services running on a virtual honeypot. There are several areas of research in TCP/IP stack fingerprinting, among them: effective methods to classify the remote operating system either by active probing or by passive analysis of network traffic, and defeating TCP/IP stack fingerprinting by normalizing network traffic. Fyodor's Nmap uses TCP and UDP probes to determine the operating system of a host [9]. Nmap collects the responses of a network stack to different queries and matches them to a signature database to determine the operating systems of the queried host. Nmap's fingerprint database is extensive and we use it as the reference for operating system personalities in HoneyD. Instead of actively probing a remote host to determine its operating systems, it is possible to identify the remote operating system by passively analyzing its network packets. *P0f* is one such tool. The TCP/IP flags inspected by P0f are similar to the data collected in Nmap's fingerprint database. On the other hand, Smart *et al.* show how to defeat fingerprinting tools by scrubbing network packets so that artifacts identifying the remote operating system are removed. This approach is similar to Honey D's personality engine as both systems change network packets to influence fingerprinting tools. In contrast to the fingerprint scrubber that removes identifiable information, Honey D changes network packets to contain artifacts of the configured operating system. High-interaction virtual honeypots can be constructed using User Mode Linux (UML) or Vmware . One example is ReVirt which can reconstruct the state of the virtual machine for any point in time. This is helpful for forensic analysis after the virtual machine has been compromised. Although high-interaction virtual honeypots can be fully compromised, it is not easy to instrument thousands of high-interaction virtual machines due to their overhead. However, the Honey D framework allows us to instrument unallocated network space with thousands of virtual honeypots. Furthermore, we may use a combination of Honey D and virtual machines to get the benefit of both approaches. In this case, Honey D provides network facades and selectively proxy's connections to services to back ends provided by high-interaction virtual machines.

VI. APPLICATION

1. Network Decoys: The traditional role of a honeypot is that of a network decoy. Our framework can be used to instrument the unallocated addresses of a production network with virtual honeypots. Adversaries that scan the production network can potentially be con-fused and deterred by the virtual honeypots. In con-junction with a NIDS, the resulting network traffic may help in getting early warning of attacks.

2. Security for Control Network in Company System.

3.In Government project, especially by the MILITARY.

4.In the research field. (Knowing trends in the attacks domain & knowing one's enemies).

VII. ADVANTAGE

1. Small datasets of high value.
2. New tools and tactics.
3. Minimal Resource.
4. Encryption of IPv6.
5. Information.
6. Simplicity.
7. Reduced cost.
8. Easier maintenance.

VIII. DISADVANTAGES

1. Limited view.
2. Limitation of OS according to hardware and Virtualization software.

IX. CONCLUSION

An algorithm was proposed and demonstrated to automatically deploy deceptive virtual network entities in a control system network. Open source passive network-monitoring tools these are evaluated and Ettercap was chosen for host identification. This work has identified several areas of possible future research. The use of virtualized networks and devices derived from the automated system presented could subsequently be used as a standard test bed for a variety of IDS systems.

ACKNOWLEDGEMENT

We would like to acknowledge our heartfelt gratitude to our guide Prof. N. B. Pokale, BSCOER for his guidance and motivation for this system.

REFERENCES

1. Todd Vollmer, Senior Member, IEEE, and Milos Manic, Senior Member, IEEE Cyber-Physical System Security with Deceptive Virtual Hosts for Industrials Control Networks, MAY 2014, VOL. 10, NO. 2.

2. Atinder Pal Singh, Birinder Singh Design andImplementation of Linux Based Hybrid ClientHoneypotIncorporating Multi-Layer Detection,September- October 2012,.

3. Hemraj Saini, Extended honeypot framework to detect old/new cyber-attacks, March, 2011.

Renuka Prasad.B, Dr Annamma Abraham, & Abhas Abhinav, Design and efficient deployment of honeypot and dynamic rule based live network intrusion collaborative system, 2, March 2011 .

5. D. A. Shea, Critical infrastructure: Control systems and the terrorist threat, Libr. Congr., Rep. Congr. RL31534, Jan. 2004.

6. Y. Huang et al., Understanding the physical and economic consequences of attacks on control systems, Int. J. Crit. Infrastruct. Prot., vol. 2, no. 3,pp. 7383, Oct. 2009.

7. C. Rieger, D. Gertman, and M. McQueen, Resilient control systems: Next generation design research, in Proc. 2nd IEEE Conf. Human Syst. Interact., Catania, Italy, May 2009, pp. 632636.

8. G. Rueff, B. Wheeler, T. Vollmer, and T. McJunkin, INL control system situational awareness technology final report, INL, Idaho Falls, ID, USA, Rep. EXT-11-23408, Jan. 2013.

9. T. Iwao, K. Yamada, M. Yura, Y. Nakaya, A. Cardenas, S. Lee et al., Dynamic data forwarding in wireless mesh networks, in Proc. IEEE Smart Grid Comm. Gaithersburg, MD, USA, 2010, pp. 385390.

The Edison Foundation. (Apr. 2010). Utility-Scale Smart Meter Deployments, Plans Proposals [Online]. Available: http://www.ediso

