ISSN 2395-1621

Dynamic Honeypot: Cyber Security System For Industrial Control Networks

[#1]Aishwarya Katkar, [#2]Pooja Awsarmol, [#3]Nikhil Mehta

[1]aishk18@yahoo.com
[2]awsarmol1234@gmail.com
[3]nikhilmehta766251@gmail.com

[#123]Department of Information Technology

Nutan Maharashtra Institute of Engineering and Technology,
India, Pune.

ABSTRACT

Beginning with the prevalence of the Internet, security of the networks became an important concern. Many security measures like firewalls, intrusion detection systems, content filters have been developed for this purpose. Honeypots are also another concept for network security. Honeypot can be defined as "a security resource whose value lies in being probed, attacked, or compromised"[5]. Honeypots have been widely used as research mechanisms for the hacker methods. This way, a thorough understanding of the attack could be made, questions like how the hacker penetrated into the system, and which vulnerability he has used can be answered (Those attacks could also be zero-day attacks). Then, honeypots started to be used for identifying the attacks in a production environment. Since honeypots are not a part of the production environment, all accesses to them has a malicious will. Therefore, this malicious traffic is monitored and logged. This way, honeypots add value to the security, but it should be noted that it cannot assure security only by itself, like many other security measures.

ARTICLE INFO

Article History

Received: 25th January 2017

Received in revised form :

25th January 2017

Accepted: 1st February 2017

Published online :

7th February 2017

Keywords: NIDS; Honeyd; Cybercrime; Investigation.

I. INTRODUCTION

Honeyd is a program that allows the user to run virtual hosts on a machine on the network, in essence its basically solely for deploying honeypots. Honeyd's functions allow it to emulate almost any known operating system at the IP stack level, instead of service level, and multitude of services. It is under license of the GNU General Public License, and is freely available to be downloaded and deployed by everyone. The senior staff engineer of Google Inc., NielsProvos, created honeyd and has written highly detailed documentation on how it works. Very surprisingly for how simple it is, honeyd is an extremely powerful program for creating virtual honeypots. Evaluations of honeyd show a 1.1 GHz Pentium III processor sustaining over 2,000 TCP transactions per second with a total bandwidth usage of 30 MBit/s. (Provos, 2004) Of course, the reason the program is so lightweight is because it is low-interaction. Instead of simulating every aspect of the OS, honeyd simply copies it at the network stack.

With the ability to create over 65,000 hosts at one time, an admin has the ability to completely obfuscate their entire network with an entire spoofed network. The creator, Provos, says that more could be tested, but that is as far as he was able to go. (65,536). Honeyd is able to create this massive 'army' of hosts by teaming with Arpd to give distinct IP addresses to each one. ARPd listens for ARP requests on the host network, and answer for IP addresses that are unallocated. This way, honeyd can be supplied with unallocated IP's from ARPd so it doesn't conflict with the normal traffic of the network. This allows honeyd to be run in a production environment, and still be able to use the unallocated address space within the network to host virtual honeypots.(Costa, 2008)

II. CLASSIFICATION OF HONEYPOT

1) Low interaction:-

Low interaction honeypot provides the limited interaction between the intruders and the attack methods[5]. This type of honeypot has a simple design and easy to deploy and monitor. This type of honeypot is provided the interaction with the real time operating system and is only program that simulates services and logs any connections to them.

2) High interaction:-

This type of honeypot is real time systems. This real systems provides the real time applications and network traffic[5]. Intruders who break into the high level operates at a real time systems. This type of honeypot provides the maximum amount of information.

Low interaction honeypot	**High interaction honeypot**
It provides limited interaction between the intruders and attack methods.	It provides more interaction between the intruders and attack methods.
It is not a real time systems.	It is a real time systems.
It gathered limited information.	It gathered maximum amount of information.

[Table 1: Difference between Low interaction and High interaction honeypot]

III. EXISTING SYSTEM

In existing system one honeypot protect only one server at a time so the drawback is each server required one seprate honeypot. In existing system there is a static node we give the manual IP to node so the data transfer done through this static node. And due to manual IP configuration IP address also limited and DDos attack occur randomly There is no alert if attacks occurs in our network .Passive attack can not be identified in this system easily

B. Problems in Existing System:-

Difficulty in analyzing the log records due to large data. Need to configure in each and every host inside the network. Huge number. of log files possess difficulty when they are processed and analyzed by security analysts as they consume a lot of time and resources. Existing system doesn't provide summarized information about each host based on protocols. Existing system provide static graphical representation. Packet information is stored in textual format which is not secured. Existing system is suffering from categorizing the normal and abnormal behavior of a system when Network environment is too complex.

IV. PROPOSED SYSTEM

A honeypot is a closely monitor network computing resource. It provides early warning about new intruders and attack methods. Basic purpose of honeypot is to gather information about the intruders and the attacks methods.

Parameter	Existing System	Proposed System
IP address	Static IP address	Dynamic IP address
Configuration	Manual IP configuration	Dynamic IP configuration
Cost	Expensive because each server requires one separate honeypot.	Inexpensive because each server does not requires one separate honeypot.
Maintenance	Difficult	Easy
Alerts	No alters are provided	Alters are provided
Passive attack	Passive attack cannot be easily identified	Passive attack can be easily identified

[Table 2: Difference between existing system and proposed system]

V. CLASSIFICATION OF HONEYPOT BASED ON IP ADDRESS

1) Physical honeypot:-

It runs on the physical honeypot. This type of honeypot are expensive to install and monitor[6].For large address space it is impossible to monitor and deploy for each IP address.For this reason the virtual honeypot is needs to deploy.

2) Virtual honeypot:-

Virtual honeypot is lightweight. We can additionally deploy one physical computer that hosts several virtual machines that act as honeypots[6]. This leads to more facile maintenance and lower physical requisites. Conventionally VMware or Utilizer-Mode Linux (UML) are acclimated to establish such virtual honeypots. These two implements sanction us to run multiple operating systems and their applications concurrently on a single physical machine, making it much more facile to accumulate data .For any honeypot to work, the external Internet needs to be able to reach it. Many of us are connected to the Internet via DSL or cable modems. These contrivances customarily employ network address translation (NAT). Albeit you might have a consummate network abaft the modem, your internal network is not reachable from the Internet. As such, you are not going to get valuable data by deploying a honeypot on a NATed network. Some NAT contrivances sanction you to transmute the port-fowarding configuration and at least sanction you get remotely exposure to the Internet. For more earnest experiments, you should find an ISP that provides you with authentic unfiltered IP connectivity.

VI. ACKNOWLEDGEMENTS

We Are Thankful To Our Project Guide Prof. Nitin Wankhade And Project Co-Ordinator Prof. Pramod Patil For Their Support. Also All The Staff Of It Department For Coordination.

VII. CONCLUSION

Honeypots gives alert about the new attacks that cannot be provided by the existing system. Honeypots provides easy maintenance, low cost and dynamic IP configuration. It helps to prevent system from external attacks. In the future, we

hope to develop more advanced honeypots that help us to gather information about threats.

REFERENCES

[1] R. Sommer and V. Paxson, Outside the closed world: On using machine learning for network intrusion detection, in Proc. IEEE Symp. Security Privacy. Oakland, CA, USA, May 2010, pp. 305316.

[2] A. Carcano et al., A multidimenisonal critical state analysis for detecting intrusions in SCADA systems, IEEE Trans. Ind. Informat., vol. 7, no. 2, pp. 179186, May 2011.

[3] J. Ousterhout, Is scale your enemy, or is scale your friend? Commun. ACM, vol. 54, no. 7, pp. 110111, Jul. 2011.

[4] Cyber-Physical System Security With Deceptive Virtual Hosts for Industrial Control Networks Todd Vollmer, Senior Member, IEEE, and Milos Manic, Senior Member, IEEE May. 2015.

[5] Detection and Analysis of Network & Application layer Attacks using Maya Honeypot Seema Sharma Computer Science and Engineering Amity University Uttar Pradesh Noida, India seema.modgil@gmail.com

[6] A Survey on Dynamic Honeypots Hamid Mohammadzadeh.e.n, Roza Honarbakhsh, and Omar,International Journal of Information and Electronics Engineering, Vol. 2, No. 2, March 2012 Zakaria.

