Research Article Impact Factor: 0.621 ISSN: 2319-507X
Sumaiyya Khan, IJPRET, 2014; Volume 2 (9): 465-471 IJPRET

INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

A SYSTEM FOR CREATION OF DYNAMIC VIRTUAL HONEYPOTS USING HADOOP

MS. SUMAIYYA Z. KHAN[1], PROF. D. M. DAKHANE[2], PROF. R. L. PARDHI[3]

1. PG Student, I.T. Branch, Sipna COET.
2. Associate Professor, Department of Computer Science and Engineering, Sipna COET.
3. Assistant Professor, Department of Information Technology, Sipna COET.

Accepted Date: 27/02/2014 ; Published Date: 01/05/2014

Abstract: System security personnel fight a seemingly unending battle to secure their digital assets against an ever-increasing onslaught of attacks. Honey pots- A security resource whose value lies in being probed, attacked, or compromised, provides a valuable tool to collect information about the behaviors of attackers in order to design and implement better defenses. Any commander will often tell his soldiers that to secure yourself against the enemy, you have to first know who your enemy is. This military doctrine readily applies to the world of network security. Just like the military, you have resources that you are trying to protect. To help protect these resources, you need to know who is your threat and how they are going to attack. On demand allocation of honey pots at right places on the network and at right time would considerably make the network more secure and harder to sneak into. This review paper is based on an idea of dynamically creating, modifying and managing virtual honey pots. This system combines the concept of honey pots and uses big data analyzer, Hadoop for quick information retrieval and analysis. The goal of this proposed system is to create evanescent honey pots at right places and times, on demand.

Keywords: Honey pots, Virtual Honey pots, Hadoop, Dynamic Honey pot Construction.

Corresponding Author: MS. SUMAIYYA Z. KHAN

Access Online On:

www.ijpret.com

How to Cite This Article:

Sumaiyya Khan, IJPRET, 2014; Volume 2 (9): 465-471

PAPER-QR CODE

INTRODUCTION

"A honey pot is an information system resource whose value lies in unauthorized or illicit use of that resource [1].” This means that whatever we designate as a honeypot, it is our expectation and goal to have the system probed, attacked, and potentially exploited. The honeypot contains no data or applications critical to the company but has enough interesting data to lure a cracker- a programmer who cracks (gains unauthorized access to) computers, typically to do malicious things.

Most current configurations are static setups consisting of either low interaction or high-interaction environments. Low-interaction honeypots have limited interaction, they normally work by emulating services and operating systems. High-interaction honeypots are different, they are usually complex solutions as they involve real operating systems and applications. Nothing is emulated, we give attackers the real thing. It is unfeasible to maintain honeypots pertaining to the entire network.

To solve this problem, dynamic honeypots came to rescue. Dynamic Honeypot is a solution, you simply plug into your network, it learns the environment, deploys the proper number and configuration of honeypots, and adapts to any changes in your networks [7]. Although there are some dynamic Honeypots, deployment of right number of virtual Honeypots at right places and at right time on demand is the need of the hour.

A physical honeypot is a real machine with its own IP address. A virtual honeypot is a simulated machine with modeled behaviors, one of which is the ability to respond to network traffic. Multiple virtual honeypots can be simulated on a single system [10].

Hadoop is a “flexible and available architecture for large scale computation and data processing on a network of commodity hardware” [9]. It is an open source framework for processing, storing and analyzing massive amounts of distributed unstructured data. It was designed to handle petabytes and Exabyte’s of data distributed over multiple nodes in parallel.

ANALYSIS OF PROBLEM

Consider two systems A and B in some network (See Fig. 1) System B was found to be important and had its equivalent honeypot B'. System A did not have its equivalent honeypot. If an attacker tries to exploit A without falling for honeypot B', the main purpose of having a honeypot in the network is unused. It is expensive to maintain honeypots that yield us no information whatsoever. It is imperative to maintain only those honeypots that could be potential targets for the attacker. Had there been a honeypot for A, it could have provided us a great deal of information.

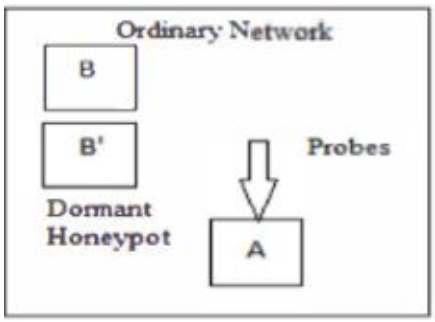

Figure 1 : Honeypot deployed in an Ordinary Network.

PROPOSED WORK

The problem mentioned above can be solved in the following manner. In this endeavor, no honeypots are deployed beforehand. The honeypots are generated 'on- demand', as per the needs generated by the network. This will not only solve the above problem but it is also an efficient way to do so.

The attacker will experience an obscured network and will be redirected to the newly created honeypot on trying to connect the victim machine. Thus the machine will remain secure from present as well as such other malicious attacks in the future. The proposed system will enable on demand allocation of Honeypots over the network emulating some of the actual running processes. Dynamic honeypots radically revolutionize the deployment and maintenance of honeypots. By learning and monitoring our networks in real time, they become a fire-and-forget solution. Not only do they become cost-effective to deploy and maintain, but they have better integration into our network. Dynamic and evanescent deployment of Honeypots at runtime will only serve to support and strengthen the current available defenses.

In addition to this, by using Hadoop in the proposed system we can store enormous data sets across distributed clusters of servers and then run “distributed” analysis applications in each cluster. It’s designed to be robust, in this Big Data applications will continue to run even when individual servers – or clusters – fail. It makes the proposed system more efficient, because it doesn’t require our system to shuttle huge volumes of data across network.

System Modules

The system can be broadly divided into four main modules (See Fig. 2) viz. Intrusion detector, Auditor, Enumerator and Honeypot Manager. As shown in the figure, when a suspicious intrusion attempt to a system present in a network is detected, it triggers a process to determine system characteristics and related data carried out by the enumerator. The

enumerator passes the requisite information to the honeypot manager which generates a honeypot. The attacker is then redirected to the newly created honeypot.

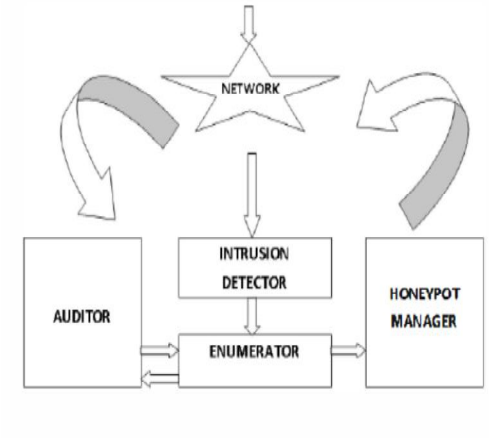

Figure 2: System Modules

1. Intrusion Detector

The main goal of this module is to capture any suspicious activity on the network. It constantly monitors the network, its incoming and outgoing traffic and in case of any discrepancy, it logs the appropriate alerts onto the designated database. The auditor can be used to get an overview of attacking methods encountered in the system over a period of time.

2. Enumerator

The Enumerator module is used to get thorough details of the system being probed. The system's IP address is obtained, which is queried to the auditor for getting its details.

3. Auditor

The Auditor is one of the main backbones of the system. The proposed system includes a python script running on all systems of the network. This script periodically updates the data generated by means of assigning timestamps to every entry. Apache Hadoop is used to satisfy this requirement. The script periodically logs its entries in HBase. Auditor is also used for periodical analysis of the network. Every intrusion attempt carried out on the system creates an equivalent log file. These logs files are analyzed to determine attacking patterns and update IDS registry. Moreover, passively, Auditor also plays a major role in analysing the logs files created at the virtual honeypots running on Honeypot with Hadoop. These analyses help in understanding the attacking mentality and improving the network's current defenses.

4. Virtual Honeypot Manager

Honeypot Manager controls the setup, configuration and deletion of virtual honeypots on the network (See Fig. 3). On receiving the information from the enumerator; it sets up a virtual honeypot by modifying its pre created template. It assigns IP address, open or closed ports, running processes, OS as well as such other subtle features to the honeypot. The attacker then, without his/her knowledge is redirected to the newly created honeypot. The VHM admin keeps track of all its created systems and in case of no responses over a long time i.e. created honeypot lying dormant over a period of time or if an attempt to sabotage a honeypot itself is detected, it deletes it.

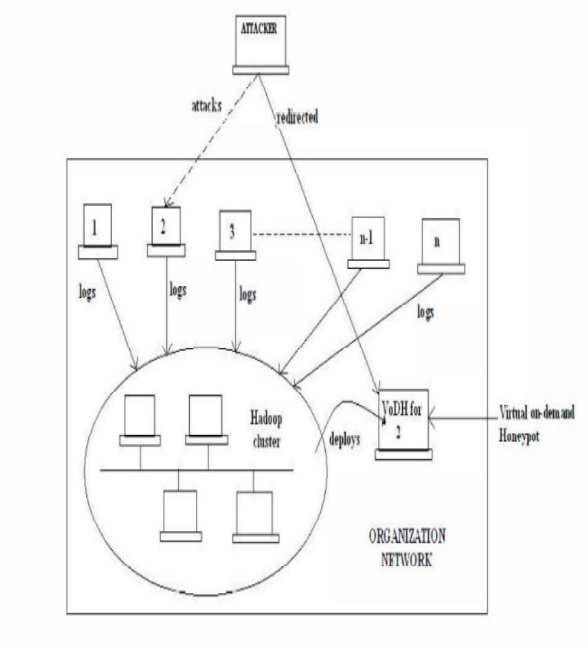

Figure 3: System at work

APPLICATIONS

1. **Intrusion Detection:** Intrusion Detection is the art of detecting inappropriate, incorrect, or anomalous activity. The proposed system can be used to determine if a computer network or server has experienced an unauthorized intrusion.
2. **Social Networking:** Web-based social systems enable new community-based opportunities for participants to engage, share, and interact. This community value and related services like search and advertising are threatened by spammers, content polluters, and malware disseminators. In an effort to preserve community value and ensure long-term success, we can use proposed for uncovering social spammers in online social systems.

3. **Network Forensics:** Network forensics deals with the capture, recording and analysis of network events in order to discover evidential information about the source of security attacks in a court of law. Using this system we can gather intelligence about the enemy and the tools and tactics of network intruders.
4. **Campus Net Security:** With the development of digital campus construction, the campus network size has been rapid growth, but there are also many network security problems. If this is applied to the campus network it can make the security of campus network unobstructed.

CONCLUSION

Honeypots with Hadoop can be found to be more efficient as compared to the conventional honeypot deployment. Standard honeypot deployment yields productive information only if it is explicitly probed or fiddled with by the attacker . This, on the other hand, promises useful data irrespective of the system on the network being targeted. This system would greatly benefit the entire computing community at large. Information security is an unending battle to safeguard our digital assets. No security mechanism can be classified as 'foolproof' as newer and stronger attacks are being discovered. Honeypots with Hadoop would enable us to get into the attacker's mind to some extent and bolster our defenses.

REFERENCES

1. Lance Spitzner, Honeypots: Definitions and value of Honeypots,

2. http://www.tracking-hackers.com.

3. John P. John, Fang Yuet et al., Heat-seeking Honeypots: Design and Experience. In Proceedings of WWW 2011-Session Web Security, 2011.

4. Christopher Hecker, Kara L. Nance, and Brian Hay, ASSERT Centre, University of Alaska Fairbanks. Dynamic Honeypot Construction. In proceedings of the 10th Colloquium for Information Systems Security Education University of Maryland, University College Adelphi, MD June 5-8, 2006.

5. L. Spitzner, 2002, Honeypots tracking Hackers. Isted. Boston, MA, USA: Addison Wesley.

6. The Bait and Switch Honeypot, http://www.violating.us/projects/baitnswitch/

7. The Honeynet Project, http://www.honeynet.org.

8. L. Spitzner, Dynamic Honeypots, http://www.securityfocus.com/infocus/1731, Sept. 2003. BAIT-TRAP,

9. http://www.cs.purdue.edu/homes/jiangx/BaitTrap, Dec. 2003.

10. Research paper on A Study on "Role of Hadoop in Information Technology era" by Vidyasagar S.D.

11. A Virtual Honeypot Framework by Neils Provos, Google, Inc.