ISSN:0975-9646

A Review on Creation of Dynamic Virtual Honeypots Using Hadoop

Sumaiyya Z. Khan, Prof. D.M.Dakhane , Prof. R.L.Pardhi

Sipna College of Engineering and Technology,
Amravati, Maharashtra, India.

***Abstract*— System security personnel fight a seemingly unending battle to secure their digital assets against an ever-increasing onslaught of attacks. Honeypots- 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 honeypots 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 honeypots. This system combines the concept of honeypots and uses big data analyzer, Hadoop for quick information retrieval and analysis. The goal of this proposed system is to create evanescent honeypots at right places and times, on demand**
.

***Keywords*— Honeypots, Virtual Honeypots, Hadoop, Dynamic Honeypot Construction.**

I. INTRODUCTION

"A honeypot 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. Attacker activity is limited to the level of emulation by the honeypot. 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. That is, some of the vulnerable or important systems are identified beforehand, and their corresponding honeypots are maintained. 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. Deploying a physical honeypot is often time intensive and expensive as different operating systems require specialized hardware and every honeypot requires its own physical system. 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. Hadoop clusters run on inexpensive commodity hardware so projects can scale-out without breaking the bank.

II. LITERATURE REVIEW AND RELATED WORK

Security has one purpose: to protect assets. For most of history, this meant building strong walls to stop the enemy and establishing small, well-guarded doors to provide secure access. Malicious activities on the Web make use of compromised Web servers, because these servers often have high page ranks and provide free resources. Attackers are therefore constantly searching for vulnerable servers. How attacker’s find, compromise, and misuse vulnerable servers [3], characterized attacker behavior and develop simple techniques to identify attack traffic.

Honeypots are usually deployed with the intent of capturing interactions with unsuspecting adversaries. The captured interactions allow researchers to understand the patterns and behaviour of attackers. For example, honeypots have been used to automate the generation of new signatures for network intrusion detection systems [2], collect malicious binaries for analysis, and quantify malicious behaviour through measurement studies.

Research effort in Dynamic Honeypot Construction [3], a method to automatically and dynamically configure honeypots based on the results of network scans. These dynamically constructed honeypots then emulate a system or network of systems in order to collect information to better protect that network. This dynamic honeypots configuration methods has been implemented and tested, and can now enhance the ability of

system administrators to identify system vulnerabilities.

A book entitled "Honeypots: Tracking Hackers" by Lance Spitzner [4], is about keeping the bad guys in- about building computers you *want* to be hacked. Traditionally, security has been purely defensive. There has been little an organization could do to take the initiative and challenge the bad guys. Honeypots change the rules. They are a technology that allows organizations to take the offensive.

Honeypots come in a variety of shapes and sizes- everything from a simple Windows system emulating a few services to an entire network of productions systems waiting to be hacked. Honeypots also have a variety of values - everything from a burglar alarm that detects an intruder to a research tool that can be used to study the motives of the black hat (bad) community. Honeypots are unique in that they are not a single tool that solves a specific problem. Instead, they are a highly flexible technology that can fulfill a variety of different roles. It is up to you how you want to use and deploy these technologies.

The Bait and Switch Honeypot [5], is a multifaceted attempt to take honeypots out of the shadows of the network security model and to make them an active participant in system defense. To do this, we are creating a system that reacts to hostile intrusion attempts by redirecting all hostile traffic to a honeypot that is partially mirroring your production system. Once switched, the would-be hacker is unknowingly attacking your honeypot instead of the real data and your clients and/or users still safely accessing the real system. Life goes on, your data is safe, and you are learning about the bad guy as an added benefit.

The Honeynet Project [6] provides the tools, tactics and motives involved in computer and network attacks, and shares the lessons learned.

In a paper entitled "Dynamic Honeypot" by Lance Spitzner [7], he stated that, the dynamic honeypot is a plug-n-play solution. You simply plug it in and the honeypot does all the work for you. It automatically determines how many honeypots to deploy, how to deploy them, and what they should look like to blend in with your environment. Even better, the deployed honeypots change and adapt to your environment.

Another paper entitled "BAIT-TRAP" [8], proposes the design and implementation of BAIT-TRAP, a catering honeypot architecture. By carefully monitoring network activities, BAIT-TRAP dynamically identifies "bait" services and automatically composes "attractive" honeypots in order to capture the expected attacks. Within seconds, a newly composed honeypot will be automatically deployed and exposed to potential attackers.

III. 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.

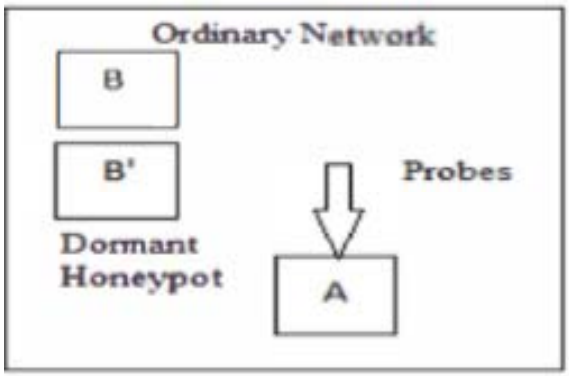

Fig. 1 : Honeypot deployed in an Ordinary Network.

IV. 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.

V. 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.

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

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

[3] 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.

[4] L. Spitzner, 2002, Honeypots tracking Hackers. lsted. Boston, MA, USA: Addison Wesley.

[5] The Bait and Switch Honeypot, http://www.violating.us/projects/baitnswitch/

[6] The Honeynet Project, http://www.honeynet.org.

[7] L. Spitzner, Dynamic Honeypots, http://www.securityfocus.com/infocus/1731, Sept. 2003.

[8] BAIT-TRAP, http://www.cs.purdue.edu/homes/jiangx/BaitTrap, Dec. 2003.

[9] Research paper on A Study on “Role of Hadoop in Information Technology era” by Vidyasagar S.D.

[10] A Virtual Honeypot Framework by Neils Provos, Google, Inc.