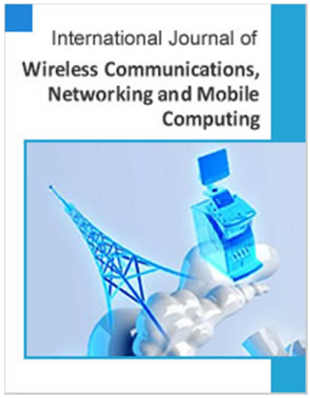
International Journal of Wireless Communications, Networking and Mobile Computing
2016; 3(4): 43-47
http://www.aascit.org/journal/wcnmc
ISSN: 2381-1137 (Print); ISSN: 2381-1145 (Online)

American Association for Science and Technology



Keywords
Snort,
Honeypot,
IPS,
IDS,
Heuristic

Received: May 1, 2016
Accepted: May 10, 2016
Published: September 9, 2016

A Novel Approach to Detect and Prevent Known and Unknown Attacks in Local Area Network

Mehak Mengi, Naveen Kumar Gondhi

School of Computer Science Engineering, Shri Mata Vaishno Devi University, Katra, India

Email address

mengimehak14@gmail.com (M. Mengi), naveen.gondhi@smvdu.ac.in (N. K. Gondhi)

Citation

Mehak Mengi, Naveen Kumar Gondhi. A Novel Approach to Detect and Prevent Known and Unknown Attacks in Local Area Network. *International Journal of Wireless Communications, Networking and Mobile Computing.* Vol. 3, No. 4, 2016, pp. 43-47.

Abstract

The most challenging task for organization in today's world is to meet security needs. Various Intrusion detection and prevention techniques have been evolving for many years. This paper present a solution to combine signature based IDS along with honeypot. Our system has been developed in the LAN environment keeping in view the analysis of requirements of software engineering framework, design, implementation and testing. For IPS system, we deploy Signature based IDS (snort base) along with low interaction honeypot in order to detect and prevent known and unknown attacks. Moreover, By means of using different vulnerability scanners, our proposed model is able to make signature based IDS to learn the critical behavior of network attacks in more closer investigation which will in turn enhance the performance of our system with a less number of false positives and high detection rate.

1. Introduction

With the increasing population of internet users, the network security is primary need to protect them from security threats, vulnerable risks and other hackers. A security framework need to be implemented to ensure the proper access rights to data and user privacy. But, almost all intrusion detection techniques like firewall or access control are not capable to completely secure our network against various network attacks. So, it is necessary to create a specialized defensive framework against various malicious activities. Over the past few decades, many Intrusion detection systems have been designed. Intrusion Detection Systems [1] monitor the network for all user activities. If any malicious activity or access rights violation occurs in the network, IDS alerts to the network administrator and the process is aborted or denied access. In this whole process, rulesare automatically generated based on process analysis which helps in decision making in future (called as Heuristics) [2]. Snort [3] is very popular IDS and quite powerful in IT security. Snort was widely used and distributed as open-source. This research is based on Honeypot technology. Spitzner [4] gives the definitions:" A honeypot is a resource whose value is in being attacked or compromised", Honeypot is a fake system that is used to trap hackers or intruders and keep them away from production system. Honeypot are of two types: Low interaction honeypot and High interaction honeypot. In this study, we used low interaction honeypot (honeyd) along with IDS system. Rules on honeyd will be generated automatically based on the logs that were captured by the honeyd itself. These automatically generated rules will automatically transfer to the IDS system (snort), which as result will enhance the security on the IDS server. So, this proposed system can mitigate the new attack pattern. This automation

will help the IDS sever to instantly stop the attacker to access the production server.

Northwestern University (USA) and Tsinghua University (China) [5] conducted a research based on detection of zero-day attacks and generate their rules to the honeynet inorder to prevent the attacks from propagating at their earlier stage.

Honeycomb, designed by Kreibich and Crowcroft. [6]

The purpose of this approach is to generate the automated signatures of attacks. To implement the Longest Common Substring algorithm (LCS), Honeycomb has used suffix tree. This algorithm was designed inorder to search for similarities in packet payloads.

Intrusion Detection Systems monitor the network for various malicious activities. If any malicious activity occurs in the network, IDS gives alerts to the network administrator but the problem arises here 99% of the IDS could generate false alarms. They provide high false positives and false negatives [7].

False positives: any benign activity is considered as malicious.

False negatives: any malicious activity is considered as benign.

The reason behind this problem is most of the Intrusion Detection Systems make use of Signature Based IDS to detect known attacks and anomaly based Detection to detect unknown attacks.

Signature Based IDS is also known as Rule Based IDS [8]. They look for specific byte sequence of each packet. They compare them with the signature stored in the database. If there is any match found, it will be identified as malicious. All the signatures needed to be stored in the signature pool. As the signature pool becomes larger, it becomes difficult to compare the new signature with the existing one. Moreover, it can detect only those attacks whose signatures are present in the signature pool. In other words, this system is be able to detect known attacks but cannot detect unknown attacks.

Anomaly Based IDS observes the traffic statistics and host behavior. [9]. It isolates the normal traffic and suspicious traffic. In order to understand the normal traffic behavior, the dataset undergoes through a training phase and learn the nature of normal or non-attacked state. If there is any deviation in normal state, it generates alerts but there are many factors that leads to deviation in normal state for e.g.: If network administrator add more users to network When Anomaly Based IDS detects that there are more users as compared to the users in the normal state it generates false alarms or false positives. This method is effective in detecting unknown attacks but we cannot ignore this fact that it generates high false alarms.

To address these problems, our research attempts to provide a systematic solution. Our work is based on Honeypot technology. Honeypot are of two types: Low interaction honeypot and High interaction honeypot: Low interaction honeypots [10] generally emulate vulnerabilities instead of presenting real vulnerable systems. Due to this, the attacker is not able to interact with it. So, they are safer. On the other hand, High interaction honeypot allows the attacker to gain full access of system. The reason for choosing the low interaction honeypot (honeyd) in our work is that it can create a simulated network of more than 60,000 virtual hosts to engage attackers and can be configured to offer different services like TCP, UDP, SMTP, SSH, FTP. From these emulated services we can easily determine what the attacker is intending to do and what they are looking for. The rest of this paper will be explained as below: Section II will describe the design and implementation; Section III presents result and discussion, and the last is Section IV concludes our paper.

2. Design and Implementation

We proposed a two phase framework. Fig.1 represents the first phase of our proposed model, which comprises of Signature Based IDS (snort base), Low interaction honeypot (honeyd), Signature rule generator. When user sends the packet over the network, Signature Based IDS(Snort Base) firstly analyses the byte sequence of each packet and compares it to the attack patterns stored in the signature based repository which in turn will help us to detect the known attacks by the Snort Base. As known attacks are easily detectable by the Signature Based IDS, the unknown attacks come into the picture whose detection by Snort Base is not possible. Our focus is on unknown attack and for detection of those attacks we have used Honeypot (honeyd). All the traffic coming from the Snort Base will be analyzed and if the DEST_IP matches to the HP_IP (honeyd IP), the traffic will be enrouted towards the Honeypot (honeyd). When this traffic reaches the honeyd all the activity will be logged and saved. Then the saved log will be further analyzed for collecting some specific information. Information that is collected is in the form of source IP, destination IP, destination port, and protocol used by the number of requests that are sent. Source IP of all the traffic will be blacklisted so that the attacker will not be able to attack in future.

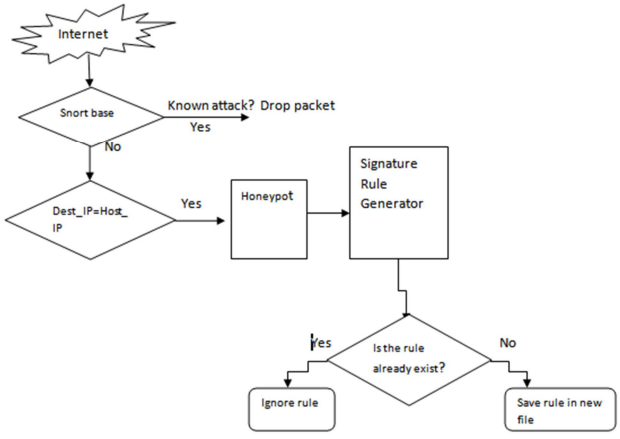


Fig. 1. *First phase of our proposed system.*

In the design of this system, we created a script that is able to retrieve the needed information from existing logs and then creating the automatic rules on honeyd. In our system both known and unknown attacks (Zero Day attacks [11]) are detected.

2.1. Snort in Learning Phase

Figure 2 represents the second phase of our proposed model in which we employ different vulnerability scanners (accunetix, ftp master, netsparker) to perform various attacks on honeyd Vulnerability scanners are responsible for generating malicious traffic like accunetix is a window based scanner that perform various web attacks (XSS, SQL injection, Directory Traversal attack and many more) [6], ftpmaster is an WebCrawler that performs various ftp attacks. In this phase, we deliberately attacked the honeyd by using different vulnerability scanners, further the honeyd will automatically generate their rules. In our system, we designed a script that will automatically transfer the rules generated by honeyd into the snort base library, this will let the snort to improve the detection efficiency in case when attacker tries to bypass the snort rules. Our system will be able to make snort base learn the critical behavior of various attacks which in turn enhance the performance of our system.

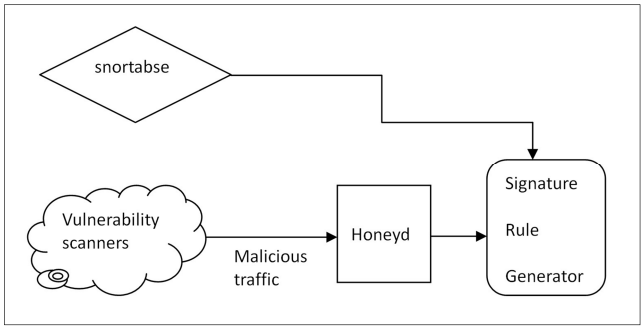


Fig. 2. *Snort in Leaning phase.*

2.2. Implementation

To implement the design, we outline the Hardware and Software requirements as shown in table 1.

Table 1. *Hardware and software requirements.*

S.no	Item	Hardware description	Software description
1	Attacker	Processor Core Intel i3, RAM 4 GB	Windows (Accunetix)Kali Linux (ftp master)
2	Honeyd	Processor Core Intel i3, RAM 4 GB	Backtrack5,VMWare,Honeyd version1.5 c
3	Snort base	Processor Core Intel i3,RAM 2 GB	Ubuntu 12.04,Virtual Box, Snort version2.9.7

When we meet the hardware and software requirements then we proceed our work by configuring both the servers (Signature Based IDS and honeyd). Honeyd provides a directory in which all logging information about the attacker is stored. In our experiment, we make use of three log files named as msg.log, web.log, ssh executed commands.log. These log files are used by different service scripts. For automatic generation of rules, we create different scripts based on which service was attacked at a specific port.

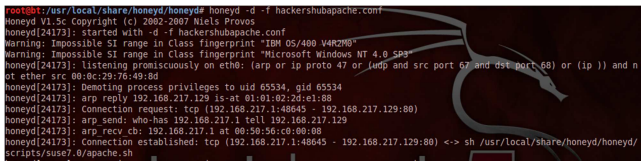


Fig. 3. *Showing the connection establishment between attacker and honeyd.*

The connection establishment between the attacker and the honeyd shows that our honeyd configuration file named as (hackershubapache.conf) works well. Once the connection is established between them, honeyd gathers valuable information regarding the attacker activities or what the attacker is intending to do. On the basis of the gathered information, it generates the logs and automatically generated their rules.

In our research, web attacks are performed on honeyd. When attacker attacks on port21 or port 22, honeyd logs the information about the ftp attacks. All the logging information regarding ftp attacks are stored in msg.log.

In var/log/honeyd directory, we created a script for various attacks including ftp, web, ssh. The screenshot shown in figure 4 shows our scripts that will automatically generate the rules of various attacks occurring at port 21 or 22 (ftp attacks).This script is stored in /var/log/honeyd directory named as ftpsnortrulegenerator.sh

Fig. 4. *Showing the procedure for creating rules for ftp attack.*

Generations of rules for snort base are illustrated in fig 5. These rules were automatically generated in honeyd and will be transferred to the snort base IDS.

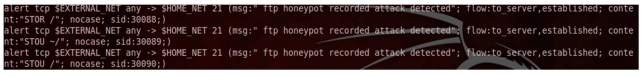


Fig. 5. *Showing the generation of rules for snort base.*

Similarly, when attacker attacks the web server, honeyd logs information about the web attacks. All the logging information is stored in file web.log. Based on the logging information, in our research we have tried to automatically generate the rules for the attacks occurred at port 80.

The generation of rules for web attacks using a script is

illustrated in fig 6.

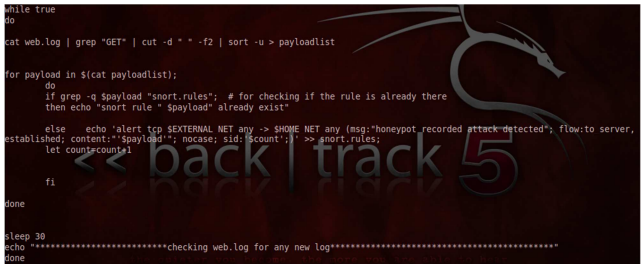


Fig. 6. *Showing the created script for generating rules of web attacks.*

Our scripts firstly remove the irrelevant data present in the log file and then sort the file in alphabetic order and then verify that whether the new rule generated by honeyd were already present or not in snort. rule file. If the rule is present then that newly generated rules are discarded, else are saved in the snort. rule file. Our scripts will automatically check log files after every 30 seconds, making our scripts more effective.

We have succeeded in completing our first phase by detecting both known and unknown attacks at a high coverage and second phase, by enhancing the efficiency of the snort base system and making the snort work in learning phase.

For enhancing the efficiency, these newly generated rules are transferred into the snort base using a script (autoruletransfer.sh) that will automatically transfer the rules from honeyd repository to the snort base repository.

For learning phase, attacks are performed deliberately by the different vulnerability scanners in order to the snort base to understand the behavior of various attacks.

3. Results and Discussion

In order to measure the performance of our proposed model, a sequence of experiments were carried out, in the research lab. Fig 7 presents the identification of intrusion over the existing and proposed system. To what extent the intrusion identified is based on the false positive and false negative rate that vary in different cases:.

Detection with basic security mechanism (firewall and antivirus enabled) system, the results that we obtained are99% of the alerts generate false negatives, 45% of the them generate false positives when we perform detection with basic security mechanism (firewall, antivirus enabled)

Detection with snort installed, the results that we obtained are 75% of the alerts generate false negatives, 22% of them are false positives.

Detection of attacks in an integrated system (combination of snort and honeyd), the results obtained are 55% of alerts generate false negatives, 20% were false positives.

Detection of attacks in a system when the snort is in learning phase, the results that we obtained areimproved as 15% of the alerts are false positives and 21% are false negatives. The rate of false positives and negatives as shown in fig 7 is least in case of our proposed system in which snort is in learning phase, this enhances the efficiency of a system.

4. Conclusion

The proposed system was developed in live LANenvironment. In this paper,we present a novel approach which is a combination of signature based Ids along with honeypot. The main goal behind using low interaction honeypot is to confuse the intruders and gather their valuable information and finally block them. This framework architecture also has an advantage of reducing the false alarm rate of the IDS.

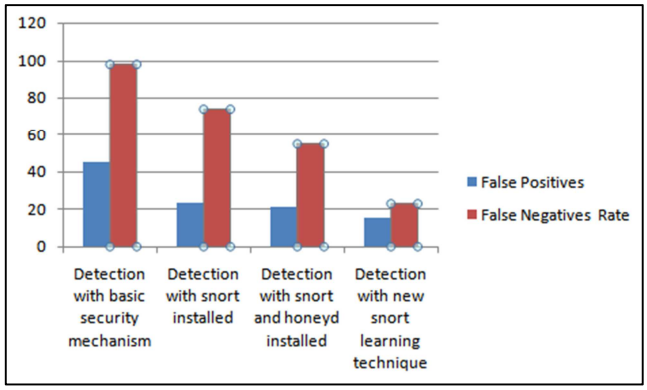


Fig. 7. *Showing the rate of false positives and negatives.*

On the other side, our proposed system is capable enough to detect zero day attacks with a high detection rate as shown in fig 8. The systems with basic security mechanism and the system with snort installed were incapable to detect zero day attacks.

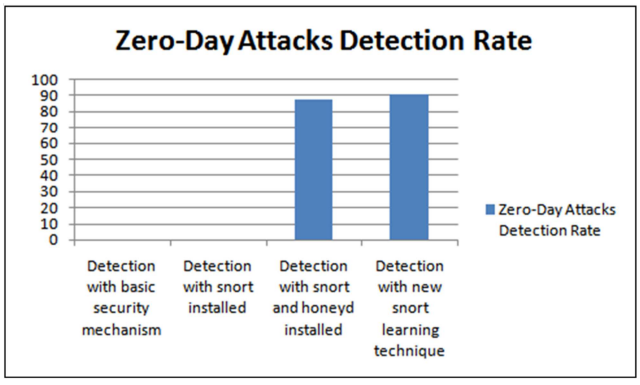


Fig. 8. *Showing detection rate of zero-day attacks.*

References

[1] James P. Anderson. Computer security threat monitoring and surveillance. Technical report, Fort Washington, 1980.

[2] Hyang-Ah Kim and Brad Karp. Autograph, toward automated, distributed worm signature detection.*USENIX Security Symposium*, pages 271–286, 2004.

[3] Snort – The defacto Standard for Intrusion Detection/Prevention, Available: http://www.snort.org, 14 February 2011.

[4] Lance Spitzner. *Honey pots: Tracking Hackers*. Addison-Wesley Professional, 2002.

[5] Zhichun Li, Lanjia Wang, Yan Chen and Zhi and "Network-based and attack-resilient length signature signature generation for zero-day polymorphicworms," http://www.cs.northwestern.edu 6

[6] Christian Kreibich and Jon Crowcroft. Honeycomb: creating intrusion detection signatures using honeypots. *ACM SIGCOMM Computer Communication Review*, 34: 51–56, 2004.

[7] Sumeet Singh, Cristian Estan, George Varghese, andStefanSavage. Automatedworm fingerprinting. *USENIX Security Symposium*, pages 45–60, 2004.

[8] Y. Tang and S. Chen. Defending against internet worms: A signature-based approach.*In Proceedings of the IEEE Infocom 2005*, 2005.

[9] A. Sundaram. An introduction to intrusion detection. *The ACM student magazine*, 1996.

[10] Niels Provos. A virtual honeypot framework. In *Proceedings of the 12th USENIXSecurity Symposium*, pages 1–14, August 2004.

[11] Portokalidis, G; Bos,H.'Eudaemon: Involuntary and On-Demand Emulation Against Zero-Day Exploit'.Third International Conference on ACM SIGOPS/ EuroSys Europen Conference on Computer Systems, Newyork-US, pp. 287-299(2008).