SECURING WMN USING HYBRID HONEYPOT SYSTEM

Paramjeet Rawat[1], Sakshi Goel[2], Megha Agarwal[3] and Ruby Singh[4]
[1]Dept. of Computer Science' IIMT Engineering College, MTU, India;
paramjeet.rawat@gmail.com
[2]Dept. of Computer Science' IIMT Engineering College, MTU, India;
Sakshi.kur14@gmail.com
[3] Dept. of Computer Science' SRM University, Ghaziabad, India;
megha.80agr@gmail.com
[4] Dept. of Information Technology' SRM University, Ghaziabad, India;
rubysinghit@gmail.com

ABSTRACT

Wireless Mesh Network (WMN) has been a field of active research in the recent years. Lot of research has focused various routing mechanism but very little effort has been made towards attack detection or intrusion detection. In this paper, we propose an attack detection approach for wireless mesh network using Honeypot technique. A Honeypot is a security resource whose value lies in being probed, attacked or compromised. A honeypot is designed to interact with attackers to collect their attack techniques and behaviors. A collection of such Honeypots laid to effectively trap the attacker is called a Honeynet. In our paper, we propose a honeynet, that is able to trap the attackers by analyzing their attacking techniques and thereby sending the logs to a centralized repository to analyze those logs so as to better understand the technique used for attacking.

KEYWORDS

Security, honeypot, honeynet, wireless, mesh, network.

1. INTRODUCTION

As various wireless networks evolve into the next generation to provide better services, a key technology, Wireless Mesh Networks, has emerged recently, which is being adopted as the wireless internetworking solution for the near future. WMN has characteristics such as rapid deployment and self configuration. Unlike traditional wireless networks, WMNs do not rely on any fixed infrastructure, it can be various forms like (i) Client WMN (ii) Infrastructural WMN and (iii) Hybrid WMN. Typical Wireless Mesh Networks (WMNs) consist of mesh routers and mesh clients [1]. Fixed or static Mesh routers, forms a wireless backbone of the WMNs and interwork with the wired networks to provide multi-hop wireless Internet connectivity to the mesh clients. Mesh clients access the network through mesh routers.

Wireless ISP's are choosing WMNs to offer Internet connectivity, as it allows a fast, easy and inexpensive network deployment. Wireless mesh networks can easily, effectively and wirelessly connect entire cities using inexpensive, existing technology. Traditional networks rely on a small number of wired access points or wireless hotspots to connect users. In a wireless mesh network, the network connection is spread out among dozens or even hundreds

DOI : 10.5121/ijdps.2012.3304

of wireless mesh nodes that "talk" to each other to share the network connection across a large area.

The development of this technology has to deal with the challenging security, architecture and protocol design issues. The emergence of new applications of WMN's necessitates the need for strong privacy protection and security mechanisms against attacks. Amongst the several security attacks, intrusion detection has been the most common and challenging attack. Traditionally intrusion detection involved a defensive approach where systems were either dedicated computers like firewalls or host based detection systems aimed at detecting attacks or preventing them. These systems existed as a part of the commercial/in-use networks and used techniques like pattern matching or anomaly detection. Another type of security systems are system integrity checkers, which are, typically host based. The problem that these systems face is that they are running on computers, which are in use on a daily basis. These systems usually have to deal with large number of connections and data transfers which results in huge log files and also makes it difficult to differentiate between normal traffic and intrusion attempts accurately. A proactive approach would be to discover these malware before they cause any damage, or at least, before their damage progresses. Such an approach is a Honeypot technique. A Honeypot is a technique used to trap the attacker by monitoring and analyzing the techniques used by the attacker to a attack a system. Almost any software or packet captured by this Honeypot is malicious, as Honeypot do not run any real software but works as a simulator that pretends to be a real node.

1.1 Type of interaction level of honeypots

The level of interaction[2] of Honeypots defines the range of attacks possible through the Honeypot. On the basis of the possible range of attack the Honeypots are categorized into two:

a. Low interaction Honeypot
b. High interaction Honeypot

1.1.1 Low interaction Honeypots

In low interaction Honeypots there is no operating system that an attacker can operate on. Instead operating system emulators are installed which interacts with the attacker. It offers limited interaction level to the attackers. It will be used to scan the port and generates attack signatures.

1.1.2 High interaction Honeypots

High interaction Honeypots have actual operating system and has tools which motivates the attacker to attack so that their attack strategies can be recorded and later analyzed. As high interaction Honeypot offers 24/7 internet connectivity, it attracts the attackers and to reduce the load of these high interaction Honeypots, only traffic filtered by low interaction Honeypots is passed to them. So high interaction Honeypots basically process the packets sent only by malicious users.

2. BACKGROUND AND RELATED WORK

Honeypot is a supplemented active defense system for network security [2,3]. It traps the hackers by recording all the activities of the hacker and thereby preventing attacks. Researchers have developed several methods and tools for malware sample collection based on honeypot techniques, among them the Nepenthes platform [4] that uses the principle of low-interaction honeypots: emulates the vulnerable parts of network services to attract and collect malware samples which attempt to infect the host by exploiting these vulnerable services.

Honeypots serve as a learning tool for system administrators and also involved studying issues[3] concerning intrusion detection systems the challenges that these systems faced. Various types of honeypots like: Virtual honeypots [1] that simulate different types of honeypots in a device, Distributed honeypot that consists of an set of honeypot systems in a network in order to trap the attacker with good success ratio. Honeypharm[5] collects and reports the malwares to a centralized repository in order to monitor all malicious activities but this was implemented on Wireless Sensor Network which have energy and power constraints. Honeycomb[6] technique, produces attack signatures automatically by analyzing traffic on a honeypot. This system produces good-quality signatures, it lays more emphasis on analyzing the attacking technique by exploring the signature rather than detecting the attacker. This approach took lot of time to detect attack on quiet nodes whereas it works well at busy nodes. Combination of [7] correlated logs and flow based attack gives high level of performance in detecting worm based attack.

Levine et al. [8] collected and analyzed rootkits manually using high-interaction honeypots. This paper is the first to introduce an automatic malware and other kinds of malware, in an automated manner. In another research, Portokalidis et al. introduce Argos [9], a containment high-interaction honeypot environment to study malware as well as human-generated attacks. Another area of research, HoneyBow [10] is based on the integrated honeynet comprising of both high-interaction and low interaction honeypots and has a capability of automatically collecting malware which propagates by exploiting new vulnerabilities. Our research is based upon this technique only but is basically made keeping in mind the Wireless Mesh Network.

3. PROPOSED WORK

We propose to create a HoneyPharm for trapping the activities of hacker in order to build a more secured WMN. Our proposal is based on a Clustered Honeypot approach where the entire network is divided into clusters. Each cluster consists of at least one Honeynet that comprises of two or more low interaction Honeypots (i.e. Honey mesh routers). These low interation Honeypot detects the attackers and traps all the activities of attacker. It then sends the attackers information to the high interaction Honeypot that are acting as a Remote Gateway(RG) which is a central place for collecting all the malwares. When the low interaction Honeypots encounters an attack, they activates a trigger on high interaction Honeypots. The high interaction Honeypot analyzes all the activities of the attacker and stores it in a log files. After the analysis, all these files are normalized and stored in a central database in the form of tables from where readable information can be presented in a proper way to the end users.

3.1 Preliminaries:

1) We consider a Infrastructural WMN [12] which is divided into several clusters where each cluster consists of a Honeypharm that comprises of two or more low interaction Honeypot acting as a Mesh Router and one or two high interaction Honeypots acting as a Gateway node.
2) There is a central repository in which all data is stored in form of tables.
3) Honeyd which is an open source software is used for creating low interaction Honeypots. Honeyd improves cyber security by providing threat detection and assessment.
4) We also assume that a Social Security Number(SSN)[11] which is used to identify the clients personal details containing his identity information (like name, father's name, address, passport number, blood group, phone number, gender, date of birth, etc.) is maintained globally in all the countries where internet is accessed. It is an entry ticket to access the Internet. So whenever any user wants to access the Internet, he has to first enter his SSN and password on a SSL encrypted page.

5) Any SSN can be used only by a single user i.e. simultaneously two persons cannot use the same SSN number. We propose that whenever a person wants to access our network, he has to enter his SSN number alongwith a password, which should be unique throughout.
6) Entering of SSN and password should be through a secured web page that is SSL encrypted.

3.2 Outline Of The Algorithm

We propose to build a integrated honeynet comprising of both low and high interaction honeypots which is a system that manages, reports, and analyzes all distributed Honeypots. First of all, the low interaction Honeypots detects the attackers or malicious code, this will minimize the risk significantly. We propose to use honeyd at low interaction honeypots, as honeyd enables a single host to claim multiple addresses. Its typical work includes port scan identification, generation of attack signature and malware collection. After detecting the attack, these low-interaction Honeypots passes the control to high interaction Honeypots. Our high interaction honeypots comprises of three malware collecting tools: MwWatcher, MwFetcher and MwHunter, all of which implements different malware collection strategies. These high interaction Honeypots captures malware's on their systems and reports about the same to their central repository. This central repository gathers the data from various Honeypot's of various cluster's and analyze them and finally gives a report for the same in a web portal from where it can be viewed (by administrators managing the HoneyPharm or users working in the network). The web portal works in two modes : User mode and Administrator mode. In the Administrator mode, the Administrator's can view the information of security attacks of all clusters but in the User mode, the users can view the information about their cluster only.

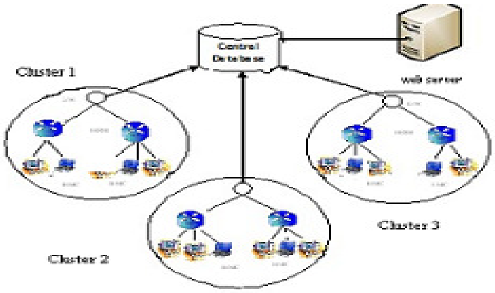

Figure 1: Clusered WMN containing Honeyclient and Honey Routers

The individual tools of high interaction Honeypot performs the following task:

- MwWatcher - It has no production activity, it just watches the file system for suspicious activity caused by malware activity by the attacker. The tool is executed on a high-interaction honeypot and exploits a characteristic feature of propagating malware: when some malicious user successfully exploits a vulnerable service and infects the honeypot, the malware sample will automatically transfer its copy to the victim and store it in the file system. Thereafter, the MwWatcher will detect this change of the filesystem and catch a binary copy of the malware sample. Now, this sample is sent to a hidden directory from where it is further collected by another tool called MwFetcher.
- MwFetcher is the second malware collection tool in the toolkit. This tool runs periodically on the host OS, issues a command to shutdown the honeypot OS and generates a listing of all files from the hard disk image of the honeypot system. Then this listing is compared to a file list generated formerly from the clean system. All

newly added or modified files are extracted since they could be object of successful infections. The samples collected in the previous step by MwWatcher are also extracted and added with the MwFetcher results. After sample extracting, MwFetcher will activate a restore procedure which reverts the honeypot OS to a clean state.

- MwHunter is the third malware collection tool in the toolkit and it is based on the PE Hunter [15] tool. MwHunter is implemented as a dynamic preprocessor plugin for Snort (an open source network intrusion detection system). MwHunter generates an alert containing a set of five informations (source IP, source port, IP protocol, destination IP, destination port) of the network stream, timestamp, and sha256sum of the captured sample. Along with these five information, we also get attacker's SSN which gives the complete information about the attacker, i.e. his name, address, gender, phone_number etc. This information will be helpful in catching or banning the attacker from further use of network.

ALGORITHM

```
Step 1.  Start honeyd on designated Honeypot in each cluster to trap the malware activities.
Step 2.  Let some attacker entered the Internet by entering his SSN and password.
Step 3.  If any honeyd detects any malwares, then
            (i) It redirect the same to designated high interaction Honeypot, alongwith his SSN information.
            (ii) At the high interaction Honeypot, MwWatcher detects the changes made to the filesystem where it silently transfers the malware samples to a hidden directory.
            (iii) Then, the MwFetcher compares these changes made to the file system with the originals stored. It then records all the added and modified files and restores the system back to original state.
            (iv) Afterwards, MwHunter generates an alert, comprising of the IP, port, clustered_id and SSN of the attacker, alongwith the sha256sum of samples captured. (This is done to alarm others about the attackers method of attack and identity of attacker. Thereafter, his SSN is banned from further accessing the network).
            (v) A copy of all the information given in Step (iv),  is also sent to the central repository for further analysis.
            (vi) The central repository normalizes these log files and store them to the database.
            (vii)  After analysis data is sent to a web portal from where it could be presented to the users to give them knowledge about various security attacks going on currently in various cluster's.
         Else
            (i)Keep running honeyd and keep checking periodically that it is working properly or not.
Step 4. End
```

Using this hybrid approach, we achieve a number of goals. First, we need to maintain a small number of high interaction Honeypot since the portion of the traffic that will be routed to them is limited. Secondly, the high interaction Honeypots are under strict monitors, so if the honeypot gets infected it will be very soon detected and also recovered . Thirdly, the information about the attacker and attacking techniques is also informed to the normal users, immediately. Also, the low interaction honeypots where honeyd is working, emulates different machines running in the network. So we can map several machines which run on the same operating system.

4. CONCLUSION

In this paper we gave a novel approach for detecting the attack and understanding the attack technique so that a better security framework could be designed that enables user to work safely in a secured environment. Our technique not only allow us to analyze the attacking technique but also enable us to trap the attacker using his SSN entered at the time of accessing the Internet. In future we will give simulation results of using this approach.

REFERENCES

[1]. Feng .Zhang, Shijie Zhou. Zhiguang Qin, Jinde Liu,“Honeypot: a Supplemented Active Defense System for Network Security ,in the preceeding of International Conference on Parallel and Distributed Computing,DOI 0-7803-7840-7/03 2003 IEEE.

[2]. Kyi Lin Lin Kyaw, Pathein Gyi, Mandalay “Hybrid Honeypot System for Network Security”, World Academy of Science, Engineering and Technology 2008, pp 266-270.

[3]. Yogendra Kumar Jain , Surabhi Singh “Honeypot based Secure Network System”, International Journal on Computer Science and Engineering (IJCSE) Vol 3, Issue 02, pp 612-620 , 2011,

[4]. Portokalidis et al. “A containment high-interaction honeypot environment to study malware as well as human-generated attacks”, Laboratory for Dependable Distributed Systems, University of Mannheim, Germany. .

[5]. Ahmad Hassan, Majid Al Ali “Collecting Malware From Distributed Honeypots Honeypharm”, GCC Conference and Exhibition (GCC), Feb 2011 IEEE pp 351-352.

[6]. Christian Kreibich et al. “Honeycomb . Creating Intrusion Detection, Signatures Using Honeypots”, ACM SIGCOMM Computer Communication,Volume 34 Issue 1, January 2004.

[7]. Falko Dressler, Wolfgang Jaegers, Reinhard German, “Flow-based Worm Detection using Correlated Honeypot Logs”. Communication in Distributed Systems (KiVS), 2007 ITG-GI Conference IEEE Xplore.

[8]. Levine, J., Grizzard, J., Owen, H.” Application of a methodology to characterize rootkits retrieved from honeynets”. In: Proceedings of the 5th Information Assurance Workshop, pp. 15–21(2004)

[9]. Portokalidis, G., Slowinska, A., Bos, H.:” Argos: an emulator for fingerprinting zero-day attacks for advertised honeypots with automatic signature generation”, ACM SIGOPS Operating Systems Review - Proceedings of the 2006 EuroSys conference,40(4) 2006., doi - 10.1145/1217935.1217938

[10]. Jianwei Zhuge, Thorsten Holz, Xinhui Han, Chengyu Song, and Wei Zou “Collecting Autonomous Spreading Malware Using High-Interaction Honeypots”, ICICS 2007, LNCS 4861, pp. 438–451, Springer-Verlag Berlin Heidelberg.

[11]. Paramjeet Rawat, M.S.Aswal, “Integrated Security Framework For Hybrid Wireless Mesh Networks”, (IJCSE) International Journal on Computer Science and Engineering, Vol. 02, No. 04, 2010, 1136-1141.

[12]. Iyatiti Mokube, Michele Adams, “Honeypots: Concepts, Approaches, and Challenges”,GA 31419,
2010.

[13]. From Wikipedia en.wikipe,”en.wikipedia.org/wiki/Wireless_mesh_network

[14]. M.S. Aswal, Paramjeet Rawat, Tarun Kumar, “Threats and Vulnerabilities in wireless mesh networks”, International Journal Of Recent Trends in Engineering, November 2009.

[15] Tillmann Werner, “Honeytrap – Ein Meta-Honeypot zur Identifikation und Analyse neuer Angriffe”, In Preceedings of the 14th DFN-CERT Workshop Sicherheit in vernetzten Systemen, 2007. http://honeytrap.mwcollect.org.