Identifying the Misbehaving User in a Network and Trapping them using Honeypot

Praveen J U, P Jayarekha

Abstract— In the IT world, information is considered to be the most valuable asset for any organization. The ability to secure this asset is the critical factor and the art of securing this asset is known as information security. In today's competing IT business, network administrator must be always available to protect the network and the information on the network with extreme measures. One of them is honeypot. Honeypot reduces the overhead of the network administrator to always be on the network and always monitoring it. Honeypot is a setup to imitate a real network. The idea is to make the attacker believe that the honeypot is a legitimate system. This Paper proposes the methodology to identify and trap the misbehaving user.

Index Terms— PM(pseudonym manager), (NM)nymble manager, NIDS(Network Intrusion Detection System), TG(Ticket Generator), TM(Ticket Manager).

I. INTRODUCTION

In today's world the data that an organization holds is very much valuable and any compromise regarding the data with the unauthorized usage is intolerable by the organization. This may result in huge damage to the same it may be in terms of economy, research, etc.

This paper initially carries discussion about various successful methods about how to secure data. Later section III provides the proposed method of how the misbehaving user is identified and data is secured.

Securing the data on the file server is of main concern these days. Many organizations have some kind of sensitive data, which are used to develop the market competitive products. These are the data which are frequently in the radar of the intruders. There are different kinds of attacks that an intruder tries to pour on the file server of the targeted organization and fetch the data of their interest. The different kinds of attacks could be any one of the following: 1) The intruder pretends to be the genuine user and send the message to the fileserver, collect all the data on file server or causes damages the data on the fileserver, 2) The intruder tries to capture the data units transmitted from genuine user to file server this unauthorized access may result in data leakage, 3) The intruder tries to collect the login privileges and get access to vital information's in file server.

II. LITRATURE SURVEY

As per [1], system proposes blocking misbehaving users on anonymizing networks, as shown in Fig. 1. The system uses

pseudonym manager whose task is to blacklist anonymous misbehavior users without knowledge of users IP addresses while allowing genuine users to connect anonymously and access the service. The system also ensures that users are aware of their blacklist status before they present a nymble, and disconnect immediately, if they are blacklisted. Patrick P. Tsang and Apu Kapadia's [1] research work has two major components pseudonym manager and nymbel manager, where in pseudonym manager is used to grant access to resources, and nymble manager is used to get access to the particular server. Further, in order to gain access to resources the user first requests PM, Where the PM generates the ticket or pseudonym to the requested resource where the generated pseudonym ticket for a particular resource is always same. In nymble manager where it uses the user's nymble and server's identity and generates another ticket or nymble which is specific to user-server pair. Here when the user connects and misbehaves with the server at any given time, the server detects the ongoing user's misbehaving activity and updates the NM about the misbehaving user and the further activity of this user is blacklisted in the NM and blocks the user from accessing the server.

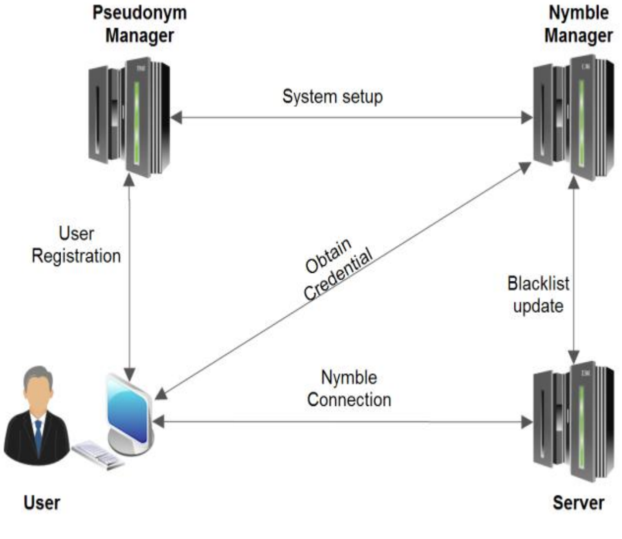

Fig.1.The Nymble System Architecture

A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource [3]. Honeypot is a program, machine or system that works beyond the routine intrusion detection[4]. The goal of honeypot is to create an environment for the attacker that the honeypot is a legitimate system. The honeypot contains services which replicates the legitimate services. There are two types of honeypots: (a) Production honeypot, and (b) Research honeypot.

Manuscript received on May 2014

Praveen J U, Information Science, BMSCE, Bangalore, India,
Dr. Jayarekha P, Information Science, BMSCE, Bangalore, India

Where production honeypot is used to help in transfer risk in an organization while the second category, is meant to gather as much information as possible [5]. Honeypots can run on any operating system and any number of services. The configured services determine the vectors available to an adversary for compromising or probing the system. A so-called "high-interaction honeypot" provides a real system with which the attacker can interact. In contrast, a "low-interaction honeypot" simulates only some parts; for example, the low-interaction honeypot "Honeyd" simulates the network stack of arbitrary systems[5]. The data collected by the honeypot can be used to understand and improve the overall network security honeypot can be used to for preventing the attacker from attacking the valuable resources.

The generic honeypot architecture look something like Fig. 2

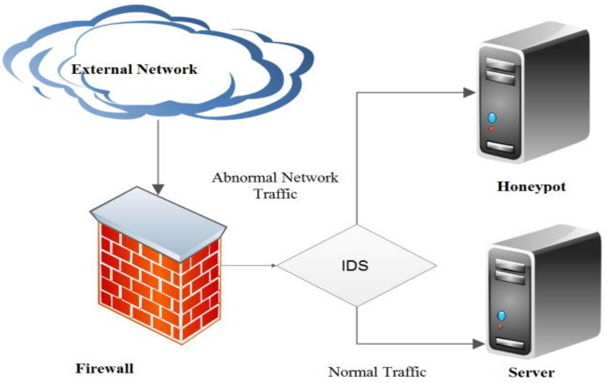

Fig.2. Basic Honeypot Architecture

The Intrusion Detection System (IDS) is the process of monitoring computers or network. Where IDS are used to detect unauthorized entrance, activity. IDS can also be used to monitor network traffic, thereby detecting of a system in being targeted by a network attack such as denial of service attack. There are two basic types of Intrusion Detection System (IDS): Host-based Intrusion Detection System (HIDS) and Network-based Intrusion Detection System (NIDS). Host-based IDs collects and analyze data that that originate on a computer that hosts a service, such as a web server. Once the data is aggregated for a given computer, it can either be analyzed locally or sent to a separate/central analysis system. Examples of Host-based IDS implementations include Windows NT/2000 Security Event logs, and UNIX Syslog. On other hand, NIDS analyze data packets that travel over the actual network. These packets are examined and sometimes compared with empirical data to verify whether they are of malicious or genuine nature [6]. The example of NIDS is snort [7], it is an open source network intrusion detection that performs real-time traffic analysis. Snort can be used to detect many kind of attacks and probes such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts and much more.

In Jiang Zhen and Zhenxiang Liu research work [8], the authors have combined the network intrusion detection using the IDS and abnormal intrusion detection using Host-based IDS which results in improved detection rate of malicious connection making the system security enhanced. The usage of seamless environment switching module with the honeypot system, which lets the impression on attacker is on real system. Resulting in collection of more attack information.

The attacks like Distributed Denial of Service (DDoS), the intruder basically makes his replica by using many computers to launch a coordinated DoS attack against one or more target. The research work[9] addresses mechanism to overcome DDoS by using Snort, where in the received packets are matched with the Snort database for matching the signature of DDoS packets. If match found then the control is handed over to honeypot that imitates the real system environment,

The survey carried out in [10] summarizes the report of detecting attacks in different protocols with the usage of IDS and combination of IDS with honeypot. Further the research work states that the IDS with honeypot is the efficient.

III. PROPOSED METHOD

The drawback of methodology as per [1] is that when dealing anonymous network the user ip address is hidden, where the blocked user can use different ip’s to gain access to server. If the user is blocked, the misbehaving user may become anger and launch a more hostile attack against the server. Anonymous networks were operated by small and friendly communities of developers. As interest in anonymous P2P increased and the user base grew, malicious users inevitably appeared and tried different attacks. This is similar to the Internet, where widespread use has been followed by waves of spam and distributed denial-of-service attacks. Such attacks may require different solutions in anonymous networks. For example, blacklisting of originator network addresses does not work because anonymous networks conceal this information. These networks are more vulnerable to DoS attacks as well due to the smaller bandwidth [2]. The benefit in [1] using the NM was a User could use the resources with the aid of a valid ticket from NM. The proposed work uses the concept of ticket to access a resource with the honeypot technology combines to form a new architecture that is used to trap the misbehavior user in honeypot instead of blocking. This helps in survey in what kind of data the misbehavior user is interested on server or what kind of attack the misbehavior user is trying to in order to damage the data. The honeypot contains all the duplicate files present in the file server but the contents of these duplicate files present on honeypot will have junk data. The architecture proposed to trap the misbehaving user is as depicted in Fig. 3. The architecture has certain assumptions that are listed as follows:

- Consider TG uses private key $Key_{ptg,pu}$ to decrypt the data received from User, User uses private key $Key_{pu,ptg}$ to decrypt the data received from TG, TM uses private key $Key_{ptm,ptg}$ and $Key_{ptm,pnids}$ to decrypt the data received from TG and NIDS respectively and NIDS uses private key $Keyp_{nids,pu}$, and $Key_{pnids,ptm}$ to decrypt the data received data from User and TM respsctively.
- Consider TG shares it public key $Key_{u,tg}$ with User, TM shares its public key $Key_{tg,tm}$ and $Key_{nids,tm}$ with TG and TM respectively and NIDS shares its public key $Key_{tm,nids}$ and $Key_{u,nids}$ with TM and User respectively.

- The user first registers with network intrusion detection system.
- For every service the user must request the TG to generate the ticket for specific service on the file server
- The TM keeps track of all the tickets generated by the TG.
- When the service request along with ticket from the user is received by NIDS, the NIDS contacts TM to verify if the received ticket for the requested service is valid or not.

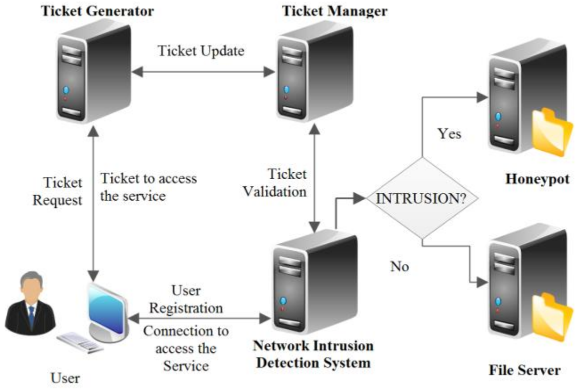

Fig. 3.The Architecture to Detect and Trap the Misbehaving User

Architecture Working: the work flow is as follows:

User → NIDS

First every user has to register with NIDS. During registration this system will get user details like (user id, password, mobile no, IP address & Mac address [Automatically Retrieved]) and it will store into database.

After registration user can login and to access a service on file server the user has to request TG to generate ticket for access the server.

User→ TG

User will send request to ticket Generator to generate the ticket. That user request consists as follows:

req : E(Key$_{u,tg}$, [uid|ADc|IPc||TS1||Service])
User→TG: req||uid||ADc||IPc||Service

TG→ User and TG→TM

Ticket Generator will generate ticket Tc to the user. The ticket consists of ticket no, Id of user, IP of Client, Mac address of user, time to generate ticket and Lifetime of the ticket, And this ticket will be in encrypted format. Ticket generator encrypts the ticket using Key$_t$. This encrypted ticket will sent to TM and user. The TG checks the *req* with the uid, ADc, IPc and Service. If this encrypted *req* matches with the uid, ADc, IPc and Service it sets the **Value** as **valid_user_req** else it sets the **Value** as **invalid_user_req.**

Tc: E(Key$_t$,[Tno||ADc||IPc||TS2||lifetime2||uid||Service])
TG →User: E (Key$_{tg,u}$,[Tc||Key$_d$])
TG →TM: E (Key$_{tg,tm}$,[Tc||uid||Tno||Key$_d$||Key$_t$||Value])

User → NIDS

If Let consider NIDS has two things. One is **valid_tpass** and **valid_ticket**.

After getting ticket from ticker generator user will try to access server through NIDS. User has to enter transaction password. If the transaction password is correct **valid_tpass** is consider as the **valid**.

If the transaction password is incorrect more than three times, **valid_tpass** consider as the **invalid**.

Once Password checking is PASS then user has to submit his ticket and service.

User→NIDS: E(Key$_{u,nids}$, [uid||ADc|IPc||Tc||Service])

NIDS←→ TM

When NIDS receives the ticket from the user, the NIDS has to send Ticket which is produced by the user along with the User id. Then Ticket Manager will decrypts ticket which is received from NIDS system and decrypt it, Once decryption is over it has to take the Ticket Number [Tno] and search its ticket database for the Ticket no. If it got the ticket with same Ticket Number then following checks are conducted.

- Check for Ticket No and User ID Validation
- Check for User IP address
- Check for MAC Address
- Check for Time Stamp
- Check for Service requested validation.

If all the checks are passed then it will send positive signal else it will send validation result along with what kind of misbehavior user done and also Key$_d$ which is used for encrypting and decrypting data that is being received from or transmitted to file server respectively.

NIDS→TM: E(Key$_{nids,tm}$, [UID||Tc||Service||IPc||ADc])
TM→NIDS: E(Key$_{tm,nids}$,[uid||ValueOfTicket||Key$_d$])

NIDS→Server

NIDS will check result received from TM. Based on the result **valid_ticket** will be desired. If ticket is valid **valid_ticket** is consider as the **valid** else **invalid.** If both the **valid_tpass** and **valid_ticket** is valid user will consider as the valid user else invalid user. If user is valid user it will connect to File Server else it will connect to HoneyPot.

File Server →It consists of all original file
HoneyPot → It consists of all duplicate file containing junk data

Tabele 1: Parameter and its Descriptions

Parameter	Description
uid	user ID.
ADc	Mac address of Client.
IPc	IP of Client.
TS1	Time Stamp, containing client time of request for ticket.
Key$_d$	It is a key used for encrypting and decrypting data that is being received from or transmitted to file server respectively.
Tno	Ticket number.
TS2	Time at generate ticket for user
Lifetime2	Expired time of the ticket
Key$_t$	This is the key which is used by TM later to decrypt the ticket.
Value	It contains the request received from the user was valid or not.
Service	Service Type (Read | Write).
ValueOfTicket	It contains the value of ticket. i.e, valid ticket or invalid ticket.

IV. CONCLUSION

With this proposed method the access to the file on file server can be secured as such, not only the identity but also the ticket place a role in securing the files on the file server, as all the connection to the file server has to go through the NIDS only. Since, NIDS keeps the complete information of the registered users, the file server can be accessed by the registered user with the valid ticket only. With this the system can secure the accessing of files from the file server because there is only one path to the file server that is through the NIDS which acts as a secured gateway.

ACKNOWLEDGMENT

The authors would like to acknowledge and thank Technical Education Quality Improvement Program [TEQIP] Phase 2, BMS College of Engineering and SPFU [State Project Facilitation Unit], Karnataka for supporting the research work.

REFERENCES

1. Patrick P. Tsang, Apu Kapadia, “Nymble: Blocking Misbehaving Users in Anonymizing Networks” IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 8, NO. 2, MARCH-APRIL 2011.
2. Anonymous P2P Available:http://wired4geeks.wordpress.com/2011/01/07/anonymous-p2p/
3. Lance Spitzner, Definitions Available:http://www.tracking-hackers.com/papers/honeypots.html.
4. Michael E.Whitman, Herbert J. Mattord,”Principles and Practices of Information Security”, 2009, pp.261.
5. Amandeep Singh, Satwinder Singh, Saab Singh M.Tech CE & Punjabi University Punjab, India"Review of Implementing a Working Honeypot System", ijarcsse,Volume 3, Issue 6, June 2013.
6. Niels Provos and Thorsten Holz, Available: http://www.eweek.com/c/a/Security/How-to-Use-Honeypots-to-Improve-Your-Network-Security/.
7. Snort website Available : http://www.snort.org/snort.
8. Jiang Zhen, Zhenxiang Liu, “New Honeypot System and its Application in Security of Employment Net Work” IEEE Symposium on Robotics and Application, 2012.
9. Nathalie Weiler ”Honeypots for Distributed Denial of Service Attacks” The Eleventh International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE’02), 2002.
10. Auttapon Pomsathit, “Effective of Unicast and Multicast IP Address Attack Over Intrusion Detection System with Honeypot”, 2012 Spring Congress on Engineering and Technology (S-CET), 2012.

AUTHORS PROFILE

Praveen J.U, is a PG Scholar in Computer Networks and Engineering at B.M.S College of Engineering, Bangalore. My research areas are Information Security, Network Security and Cloud Computing.

Dr. P Jayarekha, holds M.Tech (VTU Belgaum) in Computer Science securing second rank and Ph.D. degree in Computer Science . She has nearly two decades of experience in teaching field. She has published more than 15 research papers in referred International Journals and also few in national and international conferences. Research Scholars are working on various fields like cloud computing, WSN and related areas under her guidance. Currently, she is working as Associate Professor in the department of Information Science and Engineering at BMS College of Engineering, Bangalore, India.