IMPLEMENTATION OF HONEYPOT AS AN INTRUSION DETECTION SYSTEM FOR WIRELESS NETWORK

PUSHPA¹, YASHPAL SINGH², & S. NIRANJAN³

¹Student of M.Tech, PDM College of Engineering for Women, B’Garh, Bahadurgarh, Hjajjar, Haryana, India
²Assistant Professor, PDM College of Technology & Management, Bahadurgarh, Hjajjar, Haryana, India
³Professor, PDM College of Technology & Management, Bahadurgarh, Hjajjar, Haryana, India

ABSTRACT

Nowadays the there are so many security systems yet the network is not secured due to upcoming different threats and ideas of attackers to harm the network. We are facing with network threats that cause damage to the internet community day by day. So more and more people try to secure their network by using different traditional mechanisms including firewalls, Intrusion Detection System, etc. As compare to other technologies or mechanisms, Honeypot is one of the versatile tool or the system, meant specially for the attacker to interact with it. In this paper, we proposed a honeypot system which consist of many different websites which reacts according to the response of the attacker. In the honeypot system we have not emulated the tool instead we are providing a real operating system, services and programs for the attacker to attack.

KEYWORDS: Honeypot, Network, Security, Wireless

INTRODUCTION

An intrusion detection system (IDS) is a device or software application that monitors network or system activities for malicious activities or policy violations and produces reports to a management station.

Some systems may attempt to stop an intrusion attempt but this is neither required nor expected of a monitoring system. Intrusion detection and prevention systems (IDPS) are primarily focused on identifying possible incidents, logging information about them, and reporting attempts.[1]. Currently there are a lot of Security techniques which includes firewalls and different IDS but due to limitations many unknown attacks could not be traced and the system get effected. Since the firewalls and IDSs are designed mostly for network level attacks, they are incapable of defending application level attacks.[6].

A honeypot is a system that is built and set up in order to be hacked. Honeypot can be used in a different scenario as intrusion detection facility (burglar alarm), defense or response mechanism.

Moreover, Honeypot can be deployed in order to consume the resources of the attacker or distract him from the valuable targets and slow him down that wastes his time on the honeypot instead of attacking production systems[2].

Types of Honeypots

- 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.
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.[3].

Advantages of Honeypot are

- **Data Collection**

  Honeypots collect very little data, and what they do collect is normally of high value. This cuts the noise level down, make it much easier to collect and archive data. One of the greatest problems in security is wading through gigabytes of data to find the data you need. Honeypots can give you the exactly the information you need in a quick and easy to understand format. For example, the Honeynet Project, a group researching honeypots, collects on average only 1-5MB of data per day. This information is normally of high value also, as not only can you show network activity, but also what the attacker does once he or she gets on the system.

- **Reduction of False Positives and False Negatives**

  This has already been covered under Security and Honeypots in Section 2.

- **Simplicity**

  The very simplicity of design, implementation and use makes a honeypot a desirable method to enhance security conditions in any organization.

- **Resources**

  Many security tools can be overwhelmed by bandwidth or activity. Network Intrusion Detection Devices may not be able to keep up with network activity, dropping packets, and potentially attacks. Centralized log servers may not be able to collect all the system events, potentially dropping some events. Honeypots do not have this problem, they only capture that which comes to them.

- **Honeypots are a great training environment for security professionals.**

  In this we have proposed a honeypot model in the practical implementation with the clear outputs were also shown. The Honeypot system makes the network safe from the attackers / intruders. The Honeypot system was kept in the same network the other systems and IP address allocated was also from the network only. It does not have any unique identity like special IP address or anything. This is because the attacker could not differentiate between the honeypot system and the normal system.

  The Paper is organized as follows: Section II presents the Work on honeypot which was already done and performed. Section III shows the implementation work of the proposed work.

**RELATED WORK**

Related Work/Papers are Summarized Below

- “Design Considerations for a Honeypot for SQL Injection Attacks”, Thomas M.Chen and John Buford, 5th Worshop on Security in Communications Networks, October 2009. (In this Paper the Honeypot was created to
emulate the appearance of common defence against SQL injection and they described considerations to implement an experimental honeypot with honeyd).

- “High Interaction Honeypot System For SQL Injection Analysis”, Wei Huang, Jiao Ma and Kun Chai, 2011 International Conference of Information Technology. (In this Paper they proposed a high-interaction web honeypot system for SQL injection analysis. By (i)modifying PHP extension for MySQL to intercept data-base requests and (ii)adopting exception based and signature based detection techniques).

- “Securing WMN using Hybrid Honeypot System”, Paramjeet Rawat, Sakshi Goel, Megha Agarwal and Ruby Singh, International Journal of Distributed and Parallel Systems (IJDPS), May 2012. (In this Paper, a Honeynet is proposed that is able to trap the attackers by analyzing their attack techniques and thereby sending the logs to a centralized repository to analyze those logs so as to better understand the technique used for attacking).

- “Hybrid Honeypot System for Network Security”, Kyi Lin Lin Kyaw, Department of Engineering Physics, Mandalay. (In this Paper, they described the usefulness of low-interaction honeypot and high-interaction honeypot and comparison between them. And then they proposed hybrid honeypot architecture that combines low and high interaction honeypot to mitigate the drawback).

PROPOSED WORK

This paper describes the important features of honeypot by attracting the attackers and saving the network in future. The System work as HoneyPot or the server which consist of fake websites and applications that attract the attacker/hacker. The websites have no connection with the real world, they were only meant for the attackers. If the user act according to the Honeypot then the particular user will be blacklisted. The system belongs to research honeypots and is initially deployed on real windows platform. It emulates vulnerable websites, tempting intruders to attack by providing large amount of attractive information and exposing vulnerabilities.

In this the experiment was established by taking example through different Case Studies and shown how the attacker came into trap and blacklisted, so that it could not access the network or any other system in the future.

It consists of log having two list of database-

First (Blacklist)

It contains the list of IP blocks from the database and generate the output scheme.

Second (Whitelist)

It consist of IP addresses which should never be added(either you own them or because they belong to somebody whom you trust a lot).

CASE STUDIES

(Fake bank website attracting the attacker) (Providing greed to attacker) (Capturing the intruder if he uses SQL Injection on Login page)

Figure 1: Different Case Studies
Case Study: 1 - Fake Bank Website Attracting the Attackers

A bank login page will be shown to the attacker, he uses hit and trail method and guesses ID and password. The Honeypot will provide fake detail about some user to show him that his trail or the guess was correct and now he can do the changes or transfer money from that account. Side by side the Honeypot was monitoring its activities and blacklist the user.

When the user enters the correct login and password the he would be directed to his account.

![Figure 2: Login Page](image1)

The valid user would get the real page of his account and to check whether it is the real page, one can see the URL of the page where the URL contains the ValidLogin.aspx page which is directly from the Server.

![Figure 3: Valid User Page](image2)

When the attacker try to guess the password and uses hit and trail method for more than 3 times then he would be directed to the page which is exactly the same as the original Valid page but it could be seen that the URL was not same as of the ValidLogin Page. The URL of this page contain hacker.aspx instead of ValidLogin.aspx.

![Figure 4: Hacker Page](image3)

Now the attacker can perform any action like Deposit as it was performed below and successful transaction would be shown but side by side the information regarding the attacker would be send to the server page of Bank.
Case Study: 2 - Providing Greed to the Attacker

In this a normal Google Page has been created with an attached message that if anybody want to know the password of others then click on the button. And after that some information is gathered from the attacker to show him that he was using a genuine site. In this way if the attacker comes in trap will be blacklisted.

A normal google page with a statement “to know the password of others”, when the user click on the button would be directed to a page to enter the ID.

The user enters the ID and would be directed to the error page.
Figure 9: ID Page

Error page would be shown to the attacker and side by side his Ip address would be blacklisted.

Figure 10: Error Page

Case Study: 3 - Capturing the Intruder if he Uses SQL Injection on the Login Page

A simple Login Page has been created and a database too. If any attacker uses the SQL injection or the special characters to go into the database then that particular user will be blacklisted.

Figure 11: System Structure of Honeypot System

If the user enters the valid Login and ID would be directed to the welcome as was not a hacker and can continue in his account.

Figure 12: Login Page

Welcome pages shown below for the authorised user.
If the attacker uses the SQL Injection or the special characters then would be directed to the block page.

Blocked page for the attacker/intruder.

CONCLUSIONS

This proposed work will make the IP address of attacker to be blocked for further access of any site in the network. It obtains complete attack vectors and grasping the intention of the intruders. Moreover, it can clearly found the attack by saving the time of security researchers on reading log files and tracking intruders. Such Honeypot could be valuable tool for securing web applications when web applications are certain to continue to be attractive targets. The proposed work helps to find the attacker very easily and make the network safer.

REFERENCES


