Building a Honeypot Server And Analyzing the Data it Collects

ABSTRACT

This project aims to build a honeypot server and analyze the data it collects. The goal of the honeypot server is to create vulnerable server on the open internet which we expect will be attacked by malicious actors. These attacks will be logged by the server, then we will analyze the data collected to find common trends in the attacks.

1. INTRODUCTION

A honeypot is a server that is made intentionally vulnerable in order to attract the attention of malicious actors. The server then logs any attempts by attackers to exploit and gain access to it. The goal is for researchers to be able to analyze common trends in the kinds of attack used by attackers and even possibly discover new kinds of attacks that have never been seen before. Another use case for honeypots in an enterprise environment is to slow down attackers by allowing them to attack the non-critical honeypot instead critical infrastructure.

There are many different tools that can be used to create a honeypot. For example, the honeypot could emulate a vulnerable web app or a vulnerable end-user machine. The configuration of the honeypot depends on the goals of its creators whether that be research or protection as described above. Our goal is to research these various tools and gather enough data to be able provide an analysis of current trends in attacks.

For our approach, we will configure a honeypot as a vulnerable server using various tools which we have found to emulate common vulnerabilities and log attack data. The key insight of our project is dependent on the data which collect as there a numerous outcomes depending on the types of attacks we receive.

2. MOTIVATION

Motivation describes the most important of the related works. The ones that you either build on, prove/disprove, or in any way "extend".

Other related work, that is orthogonal to your approach but is in the same general problem-area, can be included in a separate related work section. One good place for that is at the end, so it doesn't disrupt the story here.

3. OUR ARCHITECTURE

We plan to host our honeypot on an Amazon Web Services server. The server will be an EC2 t2.micro instance. We will use an Ubuntu Server 18 image as our operating system.

To create the functionality of the honeypot, we have found three tools which we will configure and test. Kippo is a fully functional, medium interaction, SSH based honeypot. Dionaea captures malware and can simulate certain individual vulnerabilities. SNARE is a web application based honeypot.

For data analysis we will mainly use Python to graph trends. Tanner is another tool we can test which analyzes data specifically from SNARE, mentioned above.

4. CONSIDERATIONS

Since we are using AWS to host our honeypot, we must consider the configuration of the AWS instance itself. In particular, we will consider AWS firewall rules and security groups to allow all incoming connections and deny all outgoing connections. We must also consider that AWS might shutdown our server if they detect it is under attack. In that case, we will have a backup of the server prepared and it will be migrated to a private server which we control. Finally, if our honeypot does not collect enough attack data in the time period during which it is active, we will attack the server ourselves to demonstrate its capabilities.

If time permits, we have further plans to add more AWS honeypot servers and network them together with our original instance to hopefully see how an attack might traverse the network. We can also implement an intrusion detection system such as Snortif we have enough time.

5. RELATED WORK

Point out other important approaches in the problem area. For example, if you are proposing an architecture, maybe OS or PL approaches to this problem.

6. CONCLUSIONS

7. REFERENCES