

## Design and Implementation of Honeypots

Gagan Gujral<sup>\*</sup>, Erika Shehan<sup>§</sup>, Nina Tang<sup>§</sup>, Andrew Zeigler<sup>\*</sup>

{ggujral, shehan, tangn, azeigler} @purdue.edu

Project Advisor: Dr. Eugene Spafford

### **Introduction:**

We are working with honeypots, a type of computer security tool used for threat detection and analysis on a computer network. Honeypots are decoy systems that have no reason to be accessed in the day-to-day operations of a production network. Thus, any incoming traffic to a honeypot is inherently suspicious. The purpose of a honeypot is multifold: it can distract attackers from production machines, provide information about new types of attacks, and allow for study of the methods and motivations of computer attackers during and after an attack.

### **Problem:**

Honeypots, which can be physical computers or virtual hosts created by a daemon on a single computer, can be configured to blend in with production systems on the network, so that it is not obvious to an outsider whether a system is a honeypot. However, as production computers are added, removed, or updated on a network, manually keeping multiple honeypots, physical or virtual, up to date (so that they can continue to blend in with production systems) can be a daunting task.

### **Approach, Results, and Problems Encountered:**

We are researching methods to (1) ease initial configuration of virtual honeypots and (2) make virtual honeypots adapt to changes in the network around them without manual intervention. As a prototype of this method, we are building an add-on for honeyd, a popular honeypot daemon.

First, we incorporated p0f, a passive fingerprinting tool, into honeyd to analyze traffic within the network in order to gather information about the computers on the network. Initially, we intended to collect all operating system and service information using only passive fingerprinting, because it is far less invasive than active fingerprinting. Much to our disappointment, however, we discovered that p0f and other passive fingerprinting tools do not collect detailed service information. Adding support for service information to an existing passive fingerprinting tool was not possible given the time span of the project. To collect service information we are also using nmap, an active fingerprinting tool, on a limited scale. Although active fingerprinting may be less optimal for system administrators, it creates a more accurate system profile, which in turn leads to honeypot configurations that are more characteristic of the network.

Next, we developed a tool to create a “network snapshot” based on the information gathered. We also developed a tool to analyze the network snapshot and generate configurations of honeypots that blend in with the network. If changes in the network are detected, then the snapshot and configuration files are updated, and the daemon adjusts the characteristics of its virtual honeypots by adding, removing, or changing the services of one or more honeypots.

We are also incorporating a method for honeyd to collect operating system information about attackers for intelligence gathering purposes.

Our project is currently still in a state of development and testing.

### **Distribution of work:**

Gagan Gujral created start-up scripts which prompt the user for options, perform initial functionality, and launch honeyd. Erika Shehan integrated p0f into honeyd. Nina Tang developed a tool to develop the snapshot of the network. Andrew Zeigler developed a tool to create the honeyd configuration file from the network snapshot, and was the systems administrator for the computers in the test network. Each member of the group also conducted background research, helped to develop requirements/design, tested the system, and wrote documentation.

---

<sup>\*</sup> CS 490 Student

<sup>§</sup> CS 497 Student