HoneyPots and HoneyNets

Paulo Leal
Faculdade de Ciências
Universidade do Porto
up201101503@fc.up.pt

José Reisinho
Faculdade de Ciências
Universidade do Porto
up200901635@fc.up.pt

Carlos Fernando
Faculdade de Ciências
Universidade do Porto
up201002603@fc.up.pt

José Valente
Faculdade de Ciências
Universidade do Porto
up201102801@fc.up.pt

Abstract—**This document will deal with both HoneyPots and HoneyNets, in general as an application of the concept of electronic decoys to elude an attacker to a system or network and in particular their uses as to log the attack and serve as an Intrusion Detection System (IDS) of an ongoing attack, and in particular with an implementation of such a logging system analyzing the obtained results and commenting upon the extracted information.**

I. Introduction

In this day and age, when most of our personal lives can in some way be traced back to the virtual world, information and network security is gaining more attention and deservingly so. One of the technologies which saw itself being developed and increasingly used to achieve this goal has been the application of the concept of ”entrapment”[1] which led to the creation of what is referred to as HoneyPots (HPots) and HoneyNets (HNets)

II. Used Software

- HoneyD[2]: Daemon that allows for the virtualization of hosts in a network, it is released under the GNU General Public License. In particular the implementation which can be found in [3]
- Kippo[4]: A SSH medium interaction HoneyPot (HPot) designed to log Brute Force (BF) attacks and their interaction with the console.
- Kippo-Graph[5]: A visualization API for ease of access to the information logged by Kippo, capable of showing geo-locations, unique IPs, user/pass combinations all in easy to read graphics.

III. HoneyPots

A. What is a HoneyPot?

In computer terminology, a HPot is defined as a system that is set up to act as a decoy. This decoy can serve various purposes such as detecting, deflecting, studying, or even countering any unauthorized access to a network. Usually it consists of a virtual system containing bogus data and applications that try to mimic the behavior of a real system, in order to lure potential attackers into it. The HPot is thoroughly monitored and all communications with it are considered hostile, as there’s no reason for legitimate users to access a it. The activity logged by the system may provide an interesting insight about the type of attacks being directed at the network infrastructure, while distracting the attackers from valuable targets and information.

B. Types of HoneyPots

HPots can be labeled as:

- Research - The main objective of this kind of system is to enable close analysis of malicious activity and gain knowledge on how attacks are carried out, in order to learn how to more effectively protect systems against them. Data containing unique identifying properties can be placed inside the system in order to track stolen data and identify connection between attack participants.
- Production - These HPots are placed inside a production network amongst production servers, serving both as a decoy and as an intrusion detecting system (IDS). These are designed to look realistic and contain appealing resources to the attackers, in order to lure hackers. It’s an efficient way to waste the attacker’s time and resources while keeping the real assets safe, as well as gathering information about the attack, the attacker and his intentions towards the system or network.

C. Advantages of HoneyPots

- Data Collection - Common intrusion detection systems are able to generate a very large amount of alerts. This is classified as noise and turns the task of reviewing the data into a time-consuming process, requiring a lot of resources. HoneyPots, on the other hand, only log data when being interacted with, making it much easier to go through the information and detect malicious intent.
- Encryption - !
- False Positives / False Negatives - Any activity regarding a HoneyPot is classified as unauthorized and malicious, henceforth, this eliminates False Positives and False Negatives. IDS’s on the other hand, have trouble distinguishing interactions and classifying them accordingly.
- Resources - IDS’s require resource intensive hardware to be able to keep up with the heavy traffic. Increased network load demands increased hardware capacities. According to the founder of the HoneyPot Project, Lance Spitzner, millions of IP addresses can be monitorized using a single Pentium computer with 128MB of RAM.

IV. Testing

As part of the research on this subject, we have implemented a SSH HoneyPot running on a Pine64 (Similar to a RaspberryPi). This test allowed us to gather several days worth of data and further our knowlege regarding this technology.

A. Hardware

A PINE A64+ 1GB[6] was used as the hardware platform for the project. Specifications:

- 1.2 GHz Quad-Core ARM Cortex A53 64-Bit Processor. Executes both 64 and 32 Bit for scalable high performance.
- Dual Core Mali 400 MP2 Graphics card
- 1GB DDR3 Memory
- 10/100/1000Mbps Ethernet Port

B. Software

Kippo v1.5.1[4] was the selected HoneyPot software. It is a Python-based SSH HoneyPot that aims to log bruteforce attacks and any interaction done by the attacker.

Kippo-Graph v1.5.1[5] was also used in order to use the collected data from Kippo and display it in a web interface, for easy understanding of the data.

C. Installation

The setup of the software can be done by simply following the instructions on both tools (Kippo and Kippo-Graph), however, the setup of a mysql database is required so Kippo can copy the logs to it, allowing Kippo-Graph to use them and generate the data displayed in the Web Application.

Following these instructions allows replication of our configuration:

1) Pre-Requisites:

```
sudo apt-get install python-mysqldb apache2
```

2) Python Twisted: The required version for this to work is 14.0.2. Some other versions are also compatible but anything newer than 15.1 will not work. apt-get install python-dev

```
cd /tmp

wget https://github.com/twisted/twisted/
archive/twisted-14.0.2.tar.gz

tar -zxvf twisted-14.0.2.tar.gz

cd twisted-twisted-14.0.2/

./setup.py install
```

3) MySql:

```
apt-get install mysql-server

apt-get install mysql-client
```

4) Creating the database and a user with all privileges:

```
service mysql start

mysql -h localhost -u root -p

create database kippo;

GRANT ALL ON kippo.* TO 'kippo'@'localhost'
IDENTIFIED BY '<password>';

exit
```

5) Cloning the Kippo repository:

```
cd /opt/

git clone https://github.com/desaster/kippo
```

6) Initialize MySql tables:

```
cd /opt/kippo/doc/sql

mysql -u kippo -p <password>

use kippo;

source mysql.sql;

exit
```

7) Editing Kippo Configuration:

```
cd /opt/kippo/

cp kippo.cfg.dist kippo.cfg
```

Edit the necessary information for MySql

```
nano kippo.cfg
```

8) Create an unprivileged user to start Kippo and give him access to the folder:

```
useradd -d /home/kippo -s /bin/bash -m kippo
-g sudo

chown -R kippo /opt/kippo
```

9) Install the packages required for Kippo-Graph:

```
sudo apt-get install libapache2-mod-php5
php5-cli php5-common php5-cgi php5-mysql php5-gd
```

10) Install Kippo-Graph's latest version:

```
cd /tmp

wget http://bruteforce.gr/wp-content
/uploads/kippo-graph-1.5.1.tar.gz

mv kippo-graph-1.5.1.tar.gz /var/www/html

cd /var/www/html

tar zxvf kippo-graph-1.5.1.tar.gz

mv kippo-graph-1.5.1 kippo-graph

cd kippo-graph

chmod 777 generated-graphs

cp config.php.dist config.php
```

Edit the necessary information for MySql

```
nano config.php
```

11) Check if the mysql service is running:

```
service mysql status
```

12) Check if the apache2 service is running:

```
/etc/init.d/apache2 start
```

13) Start Kippo:

```
cd /opt/kippo

su kippo

./start.sh
```

14) Optional: Setup port forwarding on your router and redirect port 22 to the HoneyPot's port 2222.

D. Results

Our HoneyPot is still online (0xcmf.ddns.net) and the SSH port is set to 22 as we found the default port to maximize intrusion attempts. Accessing the address with a web browser allows access to the data gathered (http://0xcmf.ddns.net/kippo-graph/). The HoneyPot's password is set to '123456'.

1) Online Time: The system was deployed at Wednesday, 17-May-2017, 03:40 AM. By 03:45 AM the first attack was captured. The last attack captured was on Sunday, 21-May-2017, 18:19 PM. This corresponds to a period of 4 days, 14 hours and 34 minutes.

2) Login Attempts: 9116 login attempts from 268 Distinct source IP addresses.

3) Top 10 passwords:

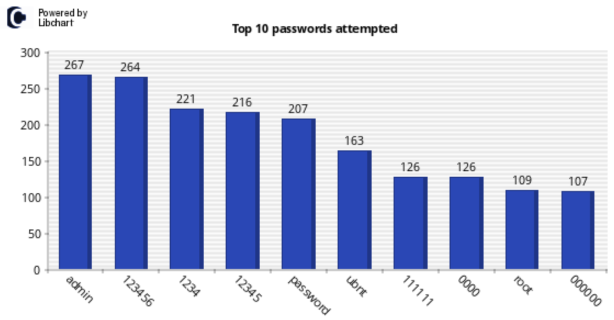


Fig. 1: Top 10 passwords that attackers try when attacking the system.

4) Top 10 usernames:

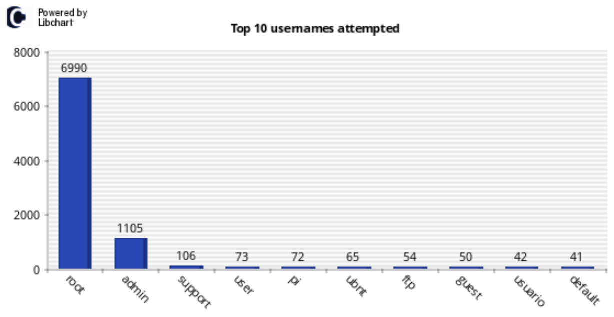


Fig. 2: Top 10 usernames that attackers try when attacking the system.

5) Success Ratio:

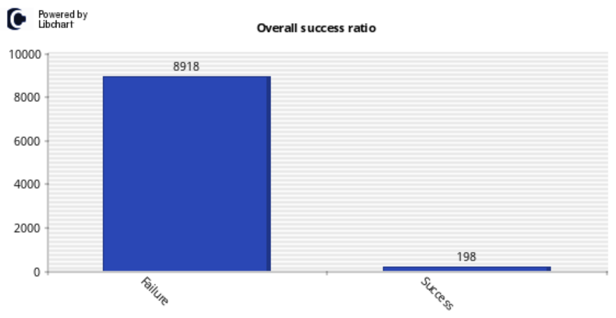


Fig. 3: Overall attack success ratio for the particular honeypot system.

6) Connections per IP:

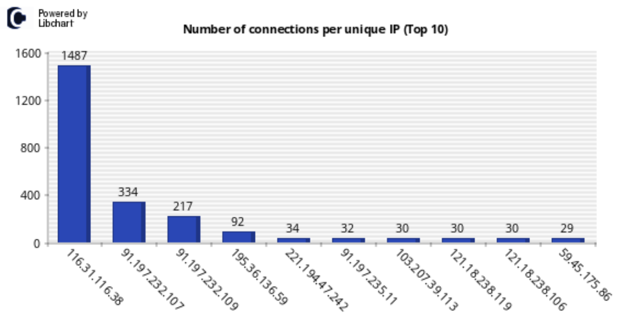


Fig. 4: Top 10 unique IPs ordered by the number of overall connections to the system.

7) Top 10 SSH clients:

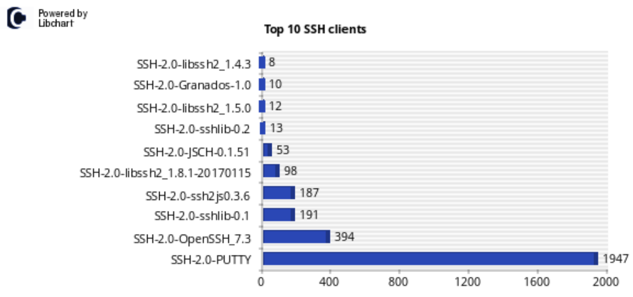


Fig. 5: Top 10 SSH clients used by attackers during their hacking attempts.

8) Top 10 Input:

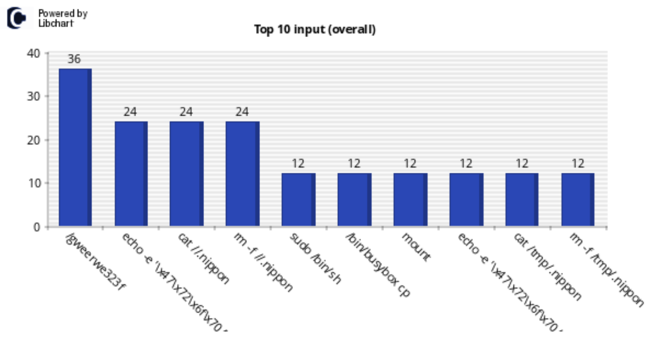


Fig. 6: Top 10 commands (overall) entered by attackers in the honeypot system.

9) *Connections per IP with Country Codes:*

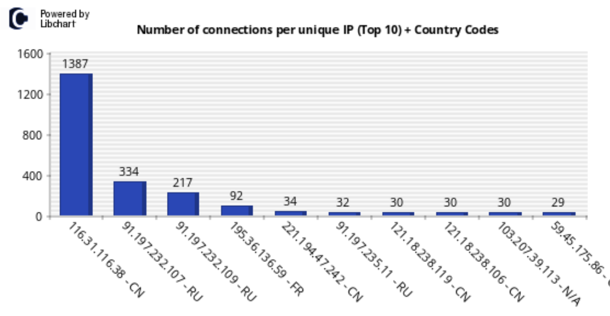


Fig. 7: Top 10 IPs ordered by the number of connections to the system with two-letter country code after each IP.

10) *Number of Connections per Country:*

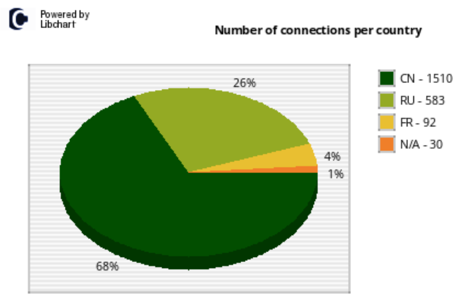
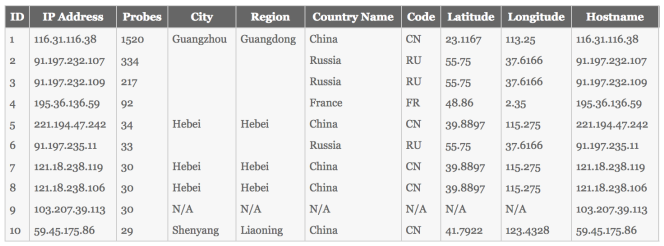


Fig. 8: Volume of attacks per country by summarising probes originating from the same nation, using the same IP or not.

11) *Attack Source Locations:*



ID	IP Address	Probes	City	Region	Country Name	Code	Latitude	Longitude	Hostname
1	116.31.116.38	1520	Guangzhou	Guangdong	China	CN	23.1167	113.25	116.31.116.38
2	91.197.232.107	334			Russia	RU	55.75	37.6166	91.197.232.107
3	91.197.232.109	217			Russia	RU	55.75	37.6166	91.197.232.109
4	195.36.136.59	92			France	FR	48.86	2.35	195.36.136.59
5	221.194.47.242	34	Hebei	Hebei	China	CN	39.8897	115.275	221.194.47.242
6	91.197.235.11	33			Russia	RU	55.75	37.6166	91.197.235.11
7	121.18.238.119	30	Hebei	Hebei	China	CN	39.8897	115.275	121.18.238.119
8	121.18.238.106	30	Hebei	Hebei	China	CN	39.8897	115.275	121.18.238.106
9	103.207.39.113	30	N/A	N/A	N/A	N/A	N/A	N/A	103.207.39.113
10	59.45.175.86	29	Shenyang	Liaoning	China	CN	41.7922	123.4328	59.45.175.86

Fig. 9: Top 10 IP addresses connected to the system.

V. HONEYD (HD)

Our assignment consisted of assembling a network and implementing the use of a HPot, with the purpose of testing its utilization and explaining how attackers attackers and detect and exploit HPots. We choose to use HD for this.

A. *What Is It?*

HoneyD is a daemon used to create virtual hosts in a network. It enables the hosts to assume multiple addresses and personalities. In this scenario a personality is the operating system that the host is recognized as. HoneyD also allows for simulated routing and networking, as well as setting up rules for ports, protocols and recording logs from requests made to the hosts. Custom scripts can also be used as actions for the rules that are set up.

B. *System Overview*

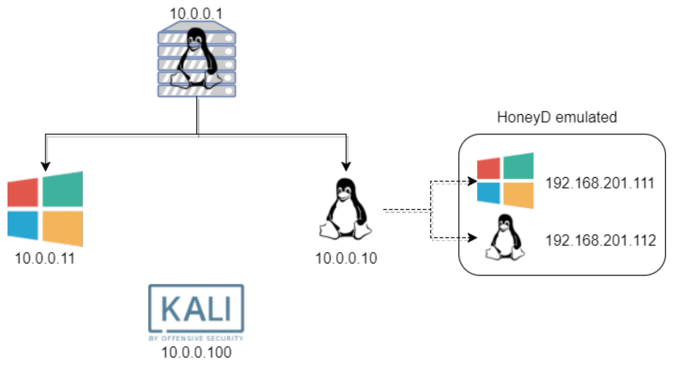


Fig. 10: Network Topology

The network was implemented using a Linux machine (10.0.0.1) as a Server, creating a connection between a Windows (10.0.0.11) and a Linux Desktop (10.0.0.10). The Desktop machine used the HD daemon and hosted a virtual network composed of another Windows (192..168.201.111) and Linux (192.168.201.112) machines. A Kali (10.0.0.100) machine was also employed outside of the network with the purpose of conducting the role of outside attacker in our tests, with the Windows one as inner attacker.

C. *System Description*

Network Configuration

```
SERVER————————————
[auser@server ~]$ ifconfig
ens3:    inet 192.168.201.39   netmask
    255.255.255.0   broadcast 192.168.201.255
         inet6 fe80::2ce6:1ce0:653a:517e
             prefixlen 64   scopeid 0x20<link>
         ether 52:54:91:22:bc:48   txqueuelen
             1000   (Ethernet)
ens7:    inet 10.0.0.1   netmask 255.255.255.0
    broadcast 10.0.0.255
         inet6 fe80::5054:29ff:fe51:680f
             prefixlen 64   scopeid 0x20<link>
         ether 52:54:29:51:68:0f   txqueuelen
             1000   (Ethernet)

[auser@server ~]$ ip route show
default via 192.168.201.1 dev ens3 proto
    static metric 100
10.0.0.0/24 dev ens7 proto kernel scope link
    src 10.0.0.1 metric 100
192.168.201.0/24 dev ens3 proto kernel scope
    link src 192.168.201.39 metric 100

DESKTOP————————————
[auser@desktop honeydTest]$ ifconfig
ens3:    inet 192.168.201.40   netmask
    255.255.255.0   broadcast 192.168.201.255
         inet6 fe80::5054:a4ff:fed8:77ea
             prefixlen 64   scopeid 0x20<link>
         ether 52:54:a4:d8:77:ea   txqueuelen
             1000   (Ethernet)
```

```
ens8:   inet 10.0.0.10  netmask 255.255.255.0
     broadcast 10.0.0.255
        inet6 fe80::5054:9bff:fe7e:5a52
            prefixlen 64  scopeid 0x20<link>
        ether 52:54:9b:7e:5a:52  txqueuelen
            1000  (Ethernet)

[auser@desktop ~]$ ip route show
default via 192.168.201.1 dev ens3 proto
    static metric 100
10.0.0.0/24 dev ens8 proto kernel scope link
    src 10.0.0.10 metric 100
192.168.201.0/24 dev ens3 proto kernel scope
    link src 192.168.201.40 metric 100


KALI——————
auser@kali:~$ sudo ifconfig
eth0:   inet 192.168.201.42  netmask
    255.255.255.0  broadcast 192.168.201.255
        inet6 fe80::5054:86ff:fe60:8696
            prefixlen 64  scopeid 0x20<link>
        ether 52:54:86:60:86:96  txqueuelen
            1000  (Ethernet)

eth1:   inet 10.0.0.100  netmask 255.255.255.0
      broadcast 10.0.0.255
        inet6 fe80::5054:4bff:fec5:f450
            prefixlen 64  scopeid 0x20<link>
        ether 52:54:4b:c5:f4:50  txqueuelen
            1000  (Ethernet)

auser@kali:~$ ip route show
default via 10.0.0.1 dev eth1 onlink
10.0.0.0/24 dev eth1 proto kernel scope link
    src 10.0.0.100
192.168.201.0/24 dev eth0 proto kernel scope
    link src 192.168.201.42
```

HoneyD configuration

```
create default

set default default tcp action allow
set default default udp action allow
set default default icmp action allow

create windows

set windows personality "Microsoft Windows XP
    Professional SP1"
set windows default tcp action open
set windows default udp action open
set windows default icmp action open
add windows tcp port 135 open
add windows tcp port 139 open
add windows tcp port 445 close
add windows udp port 448 open

set windows ethernet "00:00:24:ab:8c:12"
bind 192.168.201.111 windows

create linux

set linux personality "Linux 2.4.18"
set linux default tcp action open
set linux default udp action open
set linux default icmp action open
add linux tcp port 22 "sh /home/auser/Desktop/
    honeyd_DataSoft/scripts/linux/ssh.sh"
add linux tcp port 21 "sh /home/auser/Desktop/
    honeyd_DataSoft/scripts/linux/ftp.sh"
add linux tcp port 137 open
add linux tcp port 141 open
add linux tcp port 446 close
add linux udp port 443 open

set linux ethernet "00:00:24:ac:8c:13"
bind 192.168.201.112 linux
```

D. Tests

1) PING from Server - ping 192.168.201.112
2) SSH from Server - ssh 192.168.201.112
3) FTP from Kali - ftp 192.168.201.112

E. Results

1) PING from Server - PING 192.168.201.122 (192.168.201.122) 56(84) bytes of data
   64 bytes from 192.168.201.122: icmp_seq=1 ttl=64 time=0.760 ms
   LOG - 2017-05-21-18:46:21.0084 icmp(1) - 192.168.201.39 192.168.201.112: 8(0): 84
2) SSH from Server - Connection timed out
   LOG - 2017-05-21-20:51:25.7028 ssh(1) - 192.168.201.39 193.168.201.112:22 [Linux 2.2 20-25]
3) FTP from Kali - Connection timed out
   LOG - 2017-05-21-20:54:29.7328 ftp(1) - 192.168.201.39 193.168.201.112:21 [Kali]

VI. Detecting HoneyNets - Attackers Point Of View

This section details how an attacker could determine if the machine he is attacking is in fact a HPot or HoneyNet (HNet), and is mostly based on available Symantec articles [7][8][9][10].

A. UML Honeypots

UML uses virtual devices instead of real hard disks to store data, which are mounted as /dev/ubd*, so running 'df -k' or 'mount' can be a simple way of checking this.

Later version of UML utilize a module to mask /dev/ubd* devices into real drives, this module can be accessed if UML is started with some options switched on, namely fakehd and fake idea.

Although there is one issue: The ID of the devices are not the same as used in a standard system, but instead the specific number of 98(0x62).

Utilizing the /proc tree is also another way of quickly telling if the access is inside of UML as 'cat /proc/cpuinfo/' shows that the model is UML and the mode is Tracing Thread.

B. VMWare Honeypots

VMWare suffers many of the problems that UML does, as it is also a virtual machine. Until version 4.5 most of the hardware could not be configured and most of the default values gave it away, for example:

- Video Card - VMWare Inc PCI Display Adapter
- Hard Disk - VMWare Virtual IDE Hard Drive

The MAC address used by VMWare interfaces also had the vendor part be always one of three values: 00-50-56 / 00-0C-29 / 00-05-69.

C. chroot environments

Running 'ls -lia' on the root directory can show if the attacker is inside of a chroot environment since the inodes for the '.' and '..' directories will be drastically different.

a) Standard: 2drwxr x x 21 root root:

b) chroot: 1553552 drwxr xr x 6 1000 100:

D. HoneyD Honeypots

One problem with HD is that if an attacker sends a network packet that is malformed to a HD host it will respond as if it has received a valid packet.

Another issue is that HD uses nmap to handle fingerprinting which can produce false positive results.

HD is also user configured, which means that errors in configuration (for example, semantics) can give away that the attacker is in a HD.

References

[1] Internet security glossary, version 2. [Online]. Available: https://tools.ietf.org/html/rfc4949

[2] Honeyd virtual honeypot. [Online]. Available: http://www.honeyd.org/

[3] Used implementation of honeyd. [Online]. Available: https://github.com/DataSoft/Honeyd/

[4] Kippo - ssh honeypot. [Online]. Available: https://github.com/desaster/kippo

[5] Kippo - visual tools. [Online]. Available: http://bruteforcelab.com/kippo-graph

[6] Pine64. [Online]. Available: https://www.pine64.org/?product=pine-a64-board-1gb

[7] T. Holz and F. Raynal, "Defeating honeypots: System issues, part 1," 2005. [Online]. Available: https://www.symantec.com/connect/articles/defeating-honeypots-system-issues-part-1

[8] ——, "Defeating honeypots: System issues, part 2," 2005. [Online]. Available: https://www.symantec.com/connect/articles/defeating-honeypots-system-issues-part-2

[9] ——, "Defeating honeypots: Network issues, part 1," 2005. [Online]. Available: https://www.symantec.com/connect/articles/defeating-honeypots-network-issues-part-1

[10] ——, "Defeating honeypots: Network issues, part 2," 2005. [Online]. Available: https://www.symantec.com/connect/articles/defeating-honeypots-network-issues-part-2