

University of New Brunswick

## Table of Contents

1. Honeypot Deployment .....	错误! 未定义书签。
2. Table of Contents .....	2
3. 1. An Introduction of Honeypots .....	3
4. 2. Choose a Deployment Plan .....	5
5. 3. Details of Honeyd Installation .....	7
6. root@ubuntu: cd /usr/local/bin .....	8

# Honeypot Deployment

Hanli Ren  
Member of UNB Honeynet Project  
Faculty of Computer Science  
University of New Brunswick, Fredericton, Canada

## 1. An Introduction of Honeypots

### 1.1 What is a Honeypot

“A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource.” This definition was developed by Lance Spitzner (founder of The Honeynet Project).

- The phrase information system resource is broadly defined intentionally, so that the honeypot can be any type of computer resource. It can be a workstation, file server, mail server, printer, router, any network device, or even an entire network.
- A honeypot is intentionally put in harm’s way to be compromised and has no legitimate production value beyond the honeypot goals.

### 1.2 Why use a Honeypot

- **Low False-Positives**

False-positives are very common in intrusion detection systems (IDSs) and firewalls, as are false-negatives, to a lesser extent. Much effort is spent trying to decrease noise coming from firewalls and IDSs. Often, the noise is so high that administrators give up reading and analyzing their logs, decreasing the value of the security device.

In comparison, honeypots have no legitimate production value and should never be accessed by anyone but the honeypot administrator. Any honeypot traffic, outside the expected administrative traffic, is probably malicious. Any traffic leaving the honeypot is malicious.

- **Know Your Enemy**

KnowYour Enemy is the name of a honeypot book by Lance Spitzner and is one of the many mantras of The Honeynet Project. There is no better tool for learning what hackers are up to than a honeypot. You can learn what hackers are doing in general, or you can discover specifically what particular hackers want to do with your information resources.

Honeypots can capture everything associated with the hacker, including all network packets, uploaded malware, chat communications, and typed commands. This allows the administrator to learn what the hackers are doing and how they are doing it.

➤ **Hacking Prevention**

Honeypots aren't normally promoted for their ability to prevent malicious activity. Most honeypots, by their very nature, are passive recording devices. But this is not always the case.

First, if hackers are spending time attacking a honeypot, you are distracting them from attacking a legitimate production target. This is preventing hacking.

Second, it is important to design your honeypot so that it cannot be used to attack other computers. It is very common for hackers to use a compromised system to attack other systems. While a properly designed honeypot will prevent the hacker from successfully attacking other machines.

### **1.3 Basic Components of HoneyPot**

- **Network device hardware:**  
firewalls, routers, and switches
- **Monitoring/logging tools:**  
Key to having a honeypot is monitoring and logging what the hacker is doing.
- **Management workstation:**  
A monitoring and logging workstation collects the data from the honeypot or honeynet. Great protection must be taken to prevent hackers from discovering the monitoring/logging workstation.

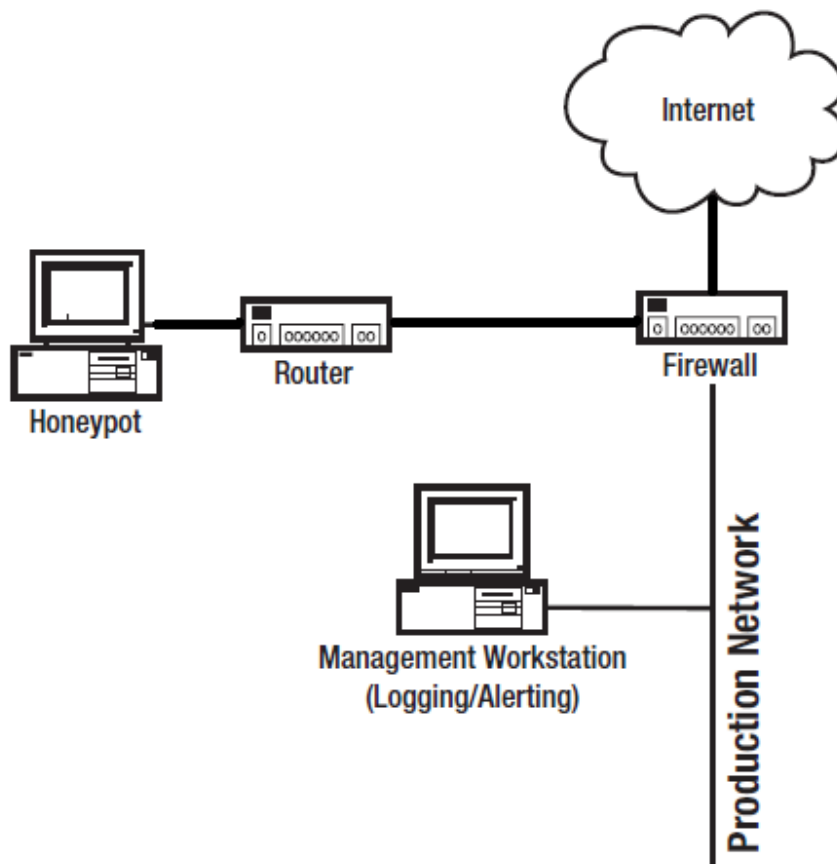
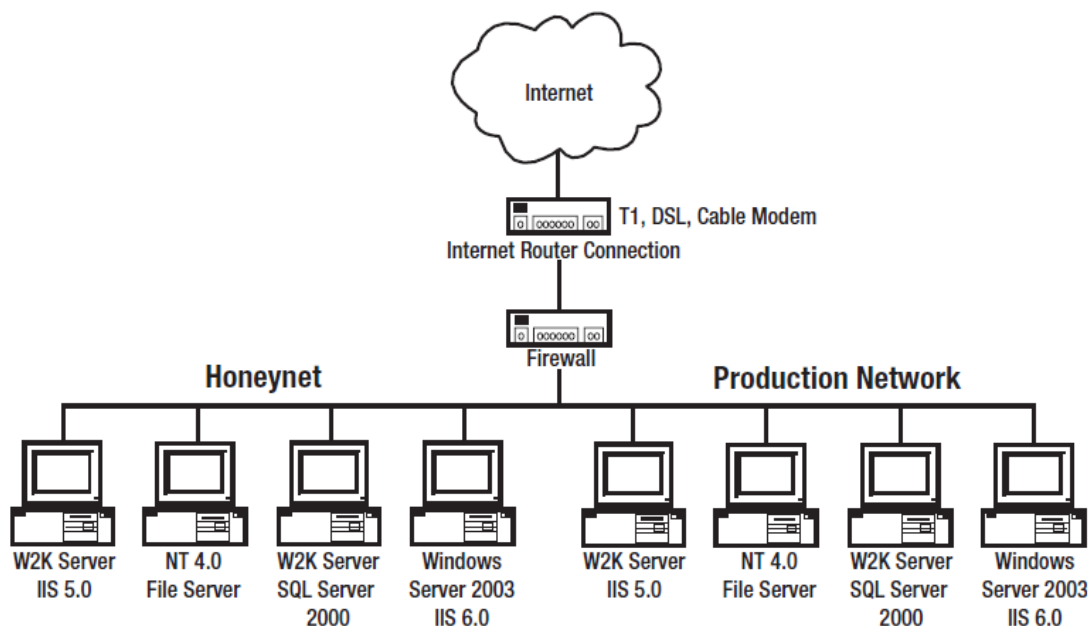


Figure 1: A sample honeypot deployment

## 2. Choose a Deployment Plan

### 2.1 Production or Research?

**Production honeypots** exist to protect your network and computers. It is their intent to lure hackers away from legitimate targets, document malicious activity, and mimic production assets.



**Figure 2: Example of a production honeynet**

**Research honeypots** are easier to set up and maintain than production honeypots. They can be placed on separate network segments, and honeypot administrators don't need to exert the extra effort involved with emulating their production environment.

Whereas a production honeypot might be limited to mimicking your working environment, a research honeypot can be any environment. If you want to learn more about Linux and Linux hacks, you can set up a Linux honeypot. If you want to learn more about Windows, your honeypot can emulate every version of Microsoft Windows from 95 to Server 2003.

## 2.2 Real or Virtual?

One of the biggest questions is whether my honeypots will be real or virtual.

- **Real honeypots** are great for high interaction, but make data control more difficult and require a lot more work if you're going to set up a honeynet.
- **Virtual machine honeypots**, like VMware, share many of the same attributes as a real honeypot, but offer quick redeployment. But they can be identified by hackers with fingerprinting techniques and, because of their high interaction, can be used to attack other targets.
- **Emulated honeypots**, like Honeyd, can be easier to set up, especially for entire honeynets, but are limited to low to medium interaction. This means that you might not capture a month-long hacker conversation, but the honeypot won't allow innocent third-party systems to be attacked.

## 3. Details of Honeyd Installation

### 3.1 Introduction of Honeyd

Honeyd is a small daemon that creates virtual hosts on a network. The hosts can be configured to run arbitrary services, and their personality can be adapted so that they appear to be running certain operating systems. Honeyd enables a single host to claim multiple addresses on a LAN for network simulation. Honeyd improves cyber security by providing mechanisms for threat detection and assessment. It also deters adversaries by hiding real systems in the middle of virtual systems.

It is possible to ping the virtual machines, or to trace-route them. Any type of service on the virtual machine can be simulated according to a simple configuration file. Instead of simulating a service, it is also possible to proxy it to another machine.

Honeyd can be used to create a virtual honey net or for general network monitoring. It supports the creation of a virtual network topology including dedicated routes and routers. The routes can be attributed with latency and packet loss to make the topology seem more realistic.

### 3.2 Honeyd Install Steps

Before we start to install a honeyd on a Linux system, we need to install following lib to support honeyd:

- **Arpd:**  
Arpd is a daemon that listens to ARP (Address Resolution Protocol) requests and answers for IP addresses that are unallocated. Using Arpd in conjunction with Honeyd, it is possible to populate the unallocated address space in a production network with virtual honeypots. With DHCP allocated IP addresses, it is possible that Arpd interferes with the DHCP server by causing Honeyd to reply to pings that the DHCP server uses to determine if an address is free.
- **Libdnet:**  
libdnet provides a simplified, portable interface to several low-level networking routines, including:
  - \* network address manipulation
  - \* kernel arp(4) cache and route(4) table lookup and manipulation
  - \* network firewalling (IP filter, ipfw, ipchains, pf, PktFilter, ...)
  - \* network interface lookup and manipulation
  - \* IP tunnelling (BSD/Linux tun, Universal TUN/TAP device)
  - \* raw IP packet and Ethernet frame transmission
- **Libevent:**

The libevent API provides a mechanism to execute a callback function when a specific event occurs on a file descriptor or after a timeout has been reached. Furthermore, libevent also support callbacks due to signals or regular timeouts.

- **Libpcap:**  
libpcap is a system-independent interface for user-level packet capture. libpcap provides a portable framework for low-level network monitoring. Applications include network statistics collection, security monitoring, network debugging, etc.

The install step should be compiling and install libdnet, libevent and libpcap first. Then install arpd and honeyd.

Also there is an easier way to install the honeyd on a Linux system. We can use Honeyd Linux Kit to do the installation. The kit file honeyd-0.7a-beta4.tgz is available on <http://www.tracking-hackers.com/solutions/honeyd>.

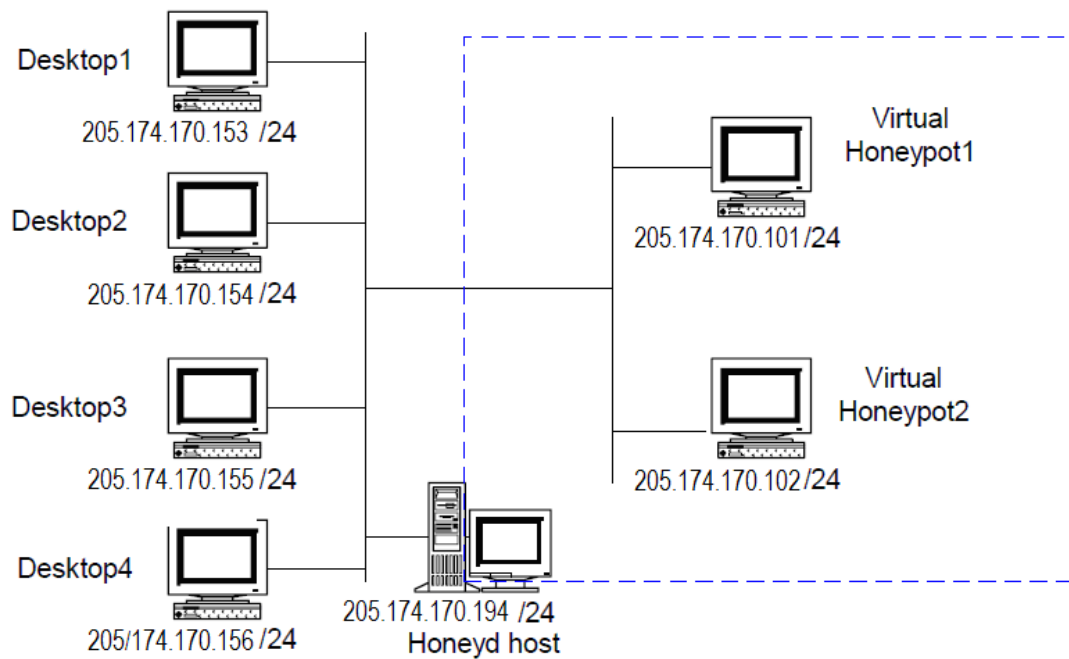
**Here are the honeyd installation steps:**

```
root@ubuntu: cd /usr/local/bin
root@ubuntu: tar -zxvf honeyd-0.7a-beta4.tgz
root@ubuntu: mv zxfv honeyd-0.7a honeyd
root@ubuntu: chown -R nobody honeyd
root@ubuntu: touch /var/log/honeyd
root@ubuntu: chown nobody /var/log/honeyd
```

### **3.3 Honeyd Configuration**

The configuration file of honeyd is [install dir]/honeyd/honeyd.conf

#### **❖ Setting up two honeypots**

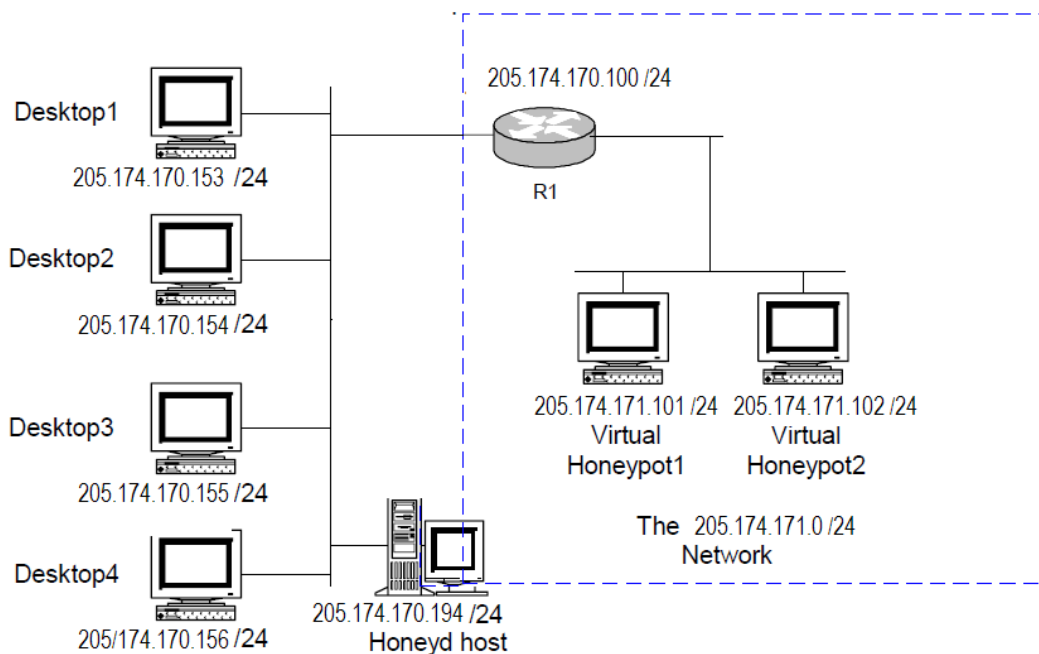


**Figure 3**

**If you want to simulate a network like Figure 3, the configuration file should be like this:**

```
### Windows computers
create windows
set windows personality "Windows NT 4.0 Server SP5-SP6"
add windows tcp port 80 "perl scripts/iis-0.95/iisemul8.pl"
add windows tcp port 139 open
add windows tcp port 137 open
add windows udp port 137 open
add windows udp port 135 open
set windows default tcp action reset
set windows default udp action reset
bind 205.174.170.101 windows
bind 205.174.170.102 windows
```

❖ **Setting up a router in the network**



**Figure 4**

**If you want to simulate a network like Figure 4, the configuration file should be like this:**

```

### Cisco router
create router
set router personality "Cisco IOS 11.3 - 12.0(11) "
set router default tcp action reset
set router default udp action reset
add router tcp port 23 "/usr/bin/perl scripts/router-
telnet.pl"
set router uid 32767 gid 32767
set router uptime 1327650
bind 205.174.170.100 router

### Windows computers
create windows
set windows personality "Windows NT 4.0 Server SP5-SP6"
add windows tcp port 80 "perl scripts/iis-
0.95/iisemul8.pl"
add windows tcp port 139 open
add windows tcp port 137 open
add windows udp port 137 open
add windows udp port 135 open
set windows default tcp action reset
set windows default udp action reset
bind 205.174.170.101 windows
bind 205.174.170.102 windows

route entry 205.174.170.100 network 205.174.170.0/16
route 205.174.170.100 link 205.174.171.0/24

```

### 3.3 Start Honeyd

❖ **Step 1 --- edit start-arpd.sh**

Example:

```
set -x
./arpd 205.174.170.0/24
```

❖ **Step 2 --- edit start-honeyd.sh**

Example:

```
set -x
./honeyd -f honeyd.conf -p nmap.prints -x xprobe2.conf
-a nmap.assoc -0 pf.os -l /var/log/honeyd
205.174.170.100-205.174.170.102
```

❖ **Step 3 --- Start arpd**

```
root@ubuntu: cd /usr/local/bin/honeyd
root@ubuntu: ./start-arpd.sh
```

❖ **Step 4 --- Start honeyd**

```
root@ubuntu: cd /usr/local/bin/honeyd
root@ubuntu: ./start-honeyd.sh
```

### 3.4 Test Honeyd

After we successfully installed and started the honeyd. Honeyd will start monitoring those virtual honeypots and writing log into the log file (/var/log/honeyd).

The log file will be look like this:

```
root@ubuntu: /var/log# tail -f honeyd
2008-09-10-20:06:28.0437 honeyd log started -----
```

We can also use some simple method to test whether our honeyd works. We can ping our new created Virtual Honeypot by another machine which also in the same VLAN.

For example, if we use machine 205.174.170.118 to ping the virtual honeypot 205.174.170.101. Honeyd will record such kind of action in the log file:

```
2008-09-10-20:07:48.0600 icmp(1) - 205.174.170.118 205.174.170.101: 8(0):
60
```