

STEPS TO INSTALLING HONEYD IN LINUX

1. Install the required libraries.
 - a. Libevent
 - b. Libdnet
 - c. Libpcap: used libpcap 0.7 and its dev file.
 - d. Libpcrc

2. Install a tool to direct traffic to the Honeyd virtual hosts.

An arpd ,also called farpd ,daemon can be installed which monitors the allocated IP address space and for any IP address that no host respond with MAC address , farpd will respond with the Honeyd physical machine MAC address. Alternatively one can just install arp if the arp proxy method is chosen to direct traffic to the virtual host, this statically assign the virtual host IP address the MAC address of the hosting machine hence the virtual IP address will be statically and permanently stored in the host arp cache.

3. Install Honeyd

Downloaded honeyd-1.5c from <http://www.citi.umich.edu/u/provos/honeyd/honeyd-1.5c.tar.gz>

4. Configuring and testing Honeyd locally

Start Honeyd by the simple command below, and check that it is running under list of running process, or check any other way.

```
# sudo ./honeyd -f config.sample 10.0.0.0/8
```

Test Honeyd locally

Started testing Honeyd locally , (i.e accessing virtual host from the hosting machine) using the sample configuration file "config.sample " by redirecting the traffic for the 10.0.0.0/8 network to the physical machines loopback interface. First add the route in the routing table to direct Honeyd traffic to the loopback.

```
#sudo route -n add -net 10.0.0.0/8 gw 127.0.0.1
```

```
#root@nsllinux07:/usr/local/bin# honeyd -i lo -f /usr/local/share/honeyd/config.sample 10.0.0.0/8
```

Ensure that ping, Nmap, telnet and traceroute are installed in the local machine.

Using this sample network configuration, I managed to successfully

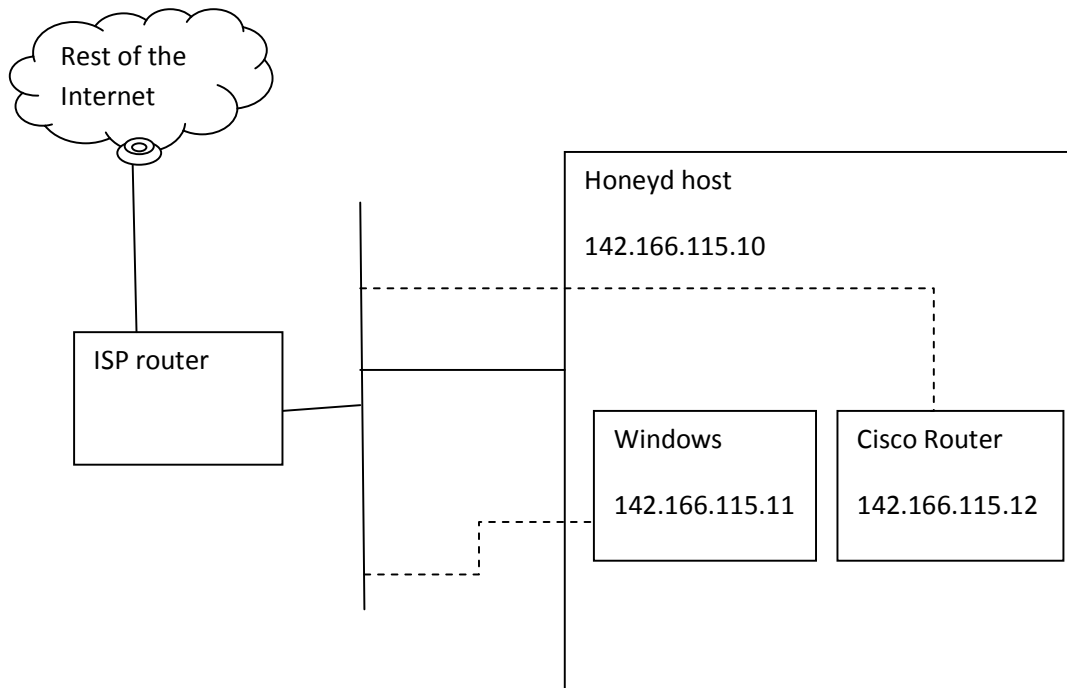
- Ping hosts in the virtual network.
- Make traceroute to virtual hosts which resolve to 1 hop with the physical machine's IP address.
- Perform an Nmap test, which so far can correctly detect the open ports in the virtual hosts and hop counts to hosts relative to the Honeyd entry router. However the test fails to return the emulate operating systems and reports unknown OS for all hosts.
- Telnet tests to routers were able to connect but with connections immediately closed, probably by security policy in the machine.

5. Logging in the Honeyd

Honeyd support packet level logging to the specified file with (-l), and service level logging with (-s). Packet level log file contains the timestamp on which the packet was logged, the protocol and ports used, Source IP address and port, Destination IP address and port .If a connection is successfully established then when it started, ended and the amount of bytes transmitted. Service level logs some details on the running emulated scripts like telnet, proxy etc.

```
#root@nsllinux07:/usr/local/bin# honeyd -i lo -f /usr/local/share/honeyd/honeyd.conf -l /var/log/honeyd/honeydlog1.log -s /var/log/honeyd/honeydlog2.log 10.0.0.0/8
```

6. Writing own configuration



The above figure shows the current network topology of our Honeyd.

7. HOW TO RUN HONEYD WITH INTERNET

1. Start farpd

Currently I am using the farpd which monitors our address space and for any received IP address that no host reply with the MAC address , farpd respond with the hosting machines MAC address hence directing virtual host traffic to this machine.However the best approach in our scenario is to use the arp proxy with “arp -s hostname physical_MAC_addr pub “, this adds the

virtual addresses to the hosting machines arp cache and hence statically sharing the MAC address with the host machine , currently it fails.

```
root@nsllinux07:~# farpd 142.166.115.8/29
```

2. Start Honeyd

```
root@nsllinux07:~# honeyd -i eth0 -f /usr/local/share/honeyd/honeyd.conf -p /usr/local/share/honeyd/nmap.prints -x /usr/local/share/honeyd/xprobe2.conf -a /usr/local/share/honeyd/nmap.assoc -l /var/log/honeyd/honeydlog1.log -s /var/log/honeyd/honeydlog2.log 142.166.115.11-142.166.115.12
```

3. To view currently running IPTables(Note below is the list of IPTables that have been running since connected to Internet until 31/08/2008.I am currently making some modifications as shown in next update section)

```
root@nsllinux07:~# iptables -L
```

Chain INPUT (policy DROP)

target prot opt source destination

ACCEPT all -- anywhere anywhere

LOG tcp -- anywhere anywhere tcp dpt:ssh flags:FSH Connection: '

ACCEPT tcp -- 142.166.115.14 142.166.115.10 tcp dpt:ssh flags:F

ACCEPT tcp -- ic314m03.cs.unb.ca 142.166.115.10 tcp dpt:ssh flags:F

LOG tcp -- anywhere anywhere tcp dpts:ftp-data:f

ACCEPT tcp -- 142.166.115.14 142.166.115.10 tcp dpts:ftp-data:f

ACCEPT icmp -- 142.166.115.14 142.166.115.10

ACCEPT tcp -- anywhere 142.166.115.10 tcp dpt:www

ACCEPT tcp -- anywhere 142.166.115.10 tcp dpt:https

ACCEPT all -- anywhere 142.166.115.11

```
ACCEPT all -- anywhere 142.166.115.12
```

```
ACCEPT all -- anywhere anywhere state RELATED,ESTAB
```

```
ACCEPT icmp -- ic314m03.cs.unb.ca 142.166.115.10
```

```
Chain FORWARD (policy DROP)
```

```
target prot opt source destination
```

```
Chain OUTPUT (policy ACCEPT)
```

```
target prot opt source destination
```

```
root@nsllinux07:~#
```

```
IPTABLES UPDATE (updated 31/08/2008)
```

Below are the new modifications to IPTables. I experienced some problems with outgoing traffic, so changed the OUTPUT policy back to ACCEPT, but only the virtual hosts are reachable. So these are still being tested.

```
#Delete all currently stored rule chains
```

```
iptables -F
```

```
#Begin by setting all policies to drop then later accept what you want.
```

```
iptables -P INPUT DROP
```

```
iptables -P FORWARD DROP
```

```
iptables -P OUTPUT DROP
```

```
#Allow any loopback traffic
```

```
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
```

#Log any ssh and ftp attempts.Allow icmp,ssh and ftp to host machine only from #142.166.115.14 and my unb machine.(This is currently just for testing)

```
iptables -A INPUT -p tcp --tcp-flags ALL SYN --dport 22 -j LOG --log-prefix
"Inbound SSH Connection: "
```

```
iptables -A INPUT -p tcp --tcp-flags ALL SYN -s 142.166.115.14 -d
142.166.115.10 --dport 22 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -A INPUT -p tcp --tcp-flags ALL SYN -s 131.202.243.81 -d
142.166.115.10 --dport 22 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -A INPUT -p tcp --dport 20:21 -j LOG --log-prefix "Inbound FTP
Connection: "
```

```
iptables -A INPUT -p tcp -s 142.166.115.14 -d 142.166.115.10 --dport 20:21 -m
state --state RELATED,ESTABLISHED -j ACCEPT
```

#Allow icmp to hosting machine only from monitoring station.

```
iptables -A INPUT -p icmp -s 142.166.115.14 -d 142.166.115.10 -m state --
state RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -A INPUT -p icmp -s 131.202.243.81 -d 142.166.115.10 -m state --
state RELATED,ESTABLISHED -j ACCEPT
```

#Allow only www and http access from the monitoring station.

```
iptables -A INPUT -p tcp -s 142.116.115.14 -d 142.166.115.10 --dport 80 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -A INPUT -p tcp -s 142.116.115.14 -d 142.166.115.10 --dport 443 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

#Allow unlimited input traffic to virtual hosts

```
iptables -A INPUT -d 142.166.115.11 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -A INPUT -d 142.166.115.12 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

#Drop any traffic to hosting machine(Also possible that this duplicates the fact #that I am not allowing any input from virtual hosts to physical machine)

```
iptables -A OUTPUT -s 142.166.115.11 -d 142.166.115.10 -m state --state RELATED,ESTABLISHED -j DROP
```

```
iptables -A OUTPUT -s 142.166.115.12 -d 142.166.115.10 -m state --state RELATED,ESTABLISHED -j DROP
```

#Allow hosting machine to access the Internet.

```
iptables -A OUTPUT -p tcp -s 142.166.115.10 --dport 80 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -A OUTPUT -p tcp -s 142.166.115.10 --dport 443 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

#Allow any output from the virtual hosts.

```
iptables -A OUTPUT -s 142.166.115.11 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -A OUTPUT -s 142.166.115.12 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

INSTALLING AN IDS

Currently working on installing Snort IDS so as to capture more information and be able to generate alerts.