The SCADA Honeynet Experience at INL

Seth Bromberger
Manager, Information Security
Pacific Gas and Electric Company

v1.2, 12 November 2008

The use of a "Type 1" honeynet[1] in combination with arp manipulation proved to be a deciding factor in the blue team victory in the recent INL Test Bed training exercise. This document provides details on configuration and setup of both components on the "corporate" network (172.1.2.0/24). (Note that the IP address ranges have been changed so as not to compromise future INL testing exercises.)

I. Hardware / OS

The hardware used in the exercise was an IBM T30 Thinkpad laptop with 256 MB RAM and a 30 gigabyte hard drive running Ubuntu Linux 8.10. No additional hardware was used; the integrated ethernet interface was the point of connection into the network. The network port was a standard 100Base-T ethernet switch connection; no span port was used (though better results might have been achieved had a span port been available).

II. Software

Honeyd 1.5b (http://www.honeyd.org) and farpd (the Ubuntu / Linux version of arpd, via apt-get) were installed. No other software was required.

III. Configuration

The ethernet interface was put into promiscuous mode

```
sudo ifconfig eth0 promisc
```

and was assigned a static IP address:

```
sudo ifconfig eth0 172.1.2.200 netmask 255.255.255.0
```

farpd was run as root via the following command:

```
sudo farpd -d 172.1.2.0/24
```

This ran farpd in the foreground, and specified that the entire address range was in scope for arp responses. (Note: farpd does its best to claim only unused IP addresses on the network; it does this via various methods, but some diligence is still required - especially immediately after startup.)

---

[1] The creators of the soon-to-be-released Firepot managed honeynet service have designated three types of honeypots. Type 1 honeypots are designed to detect suspicious activity, but provide minimal simulation of production environments; Type 2 honeypots typically mirror production servers and implement real network services; and Type 3 honeypots simulate real vulnerabilities on what appear to be production servers. The Firepot project is expected to be released in 2009.

Disclosure: the author is involved in the Firepot project.

Finally, honeyd was run as root with the following command:

```
sudo honeyd -d -f ./honeyd.conf 172.1.2.0/24
```

This runs honeyd in the foreground, uses the honeyd.conf file in the current directory for configuration, and specifies that it will apply to the entire subnet (see “default template”, below).

The honeyd configuration file contained the following:

```
1   create default
2   set default personality "Microsoft Windows 2000 Server SP2"
3   set default default tcp action reset
4   set default default udp action reset
5   set default default icmp action open
6   add default tcp port 139 open
7   add default tcp port 137 open
8   add default udp port 137 open
9   add default udp port 135 open
10  add default tcp port 0 open
11  add default tcp port 20000 open
12
13  create Linux
14  set Linux personality "Linux Kernel 2.4.0 - 2.5.20"
15  set Linux default tcp action reset
16  set Linux default udp action reset
17  set Linux default icmp action open
18  add Linux tcp port 22 "sh honeyscripts/fakessh.sh"
19
20  bind 172.1.2.3 Linux
```

Lines 1 - 11 create a default configuration that is used whenever a specific configuration is not configured. Lines 13 - 20 provide an example of a specific configuration that replaces the default behavior for a single host. The default configuration mimics a standard Windows 2000 SP2 server running a DNP-based application on port 20000. A scan of a honeypot configured in this manner would suggest that the device is an HMI.

The Linux configuration mimics a standard Linux server running SSH with an IP address of 172.1.2.3. Upon connection to port 22 (SSH), an attacker would be presented with the output of the script in line 18, which would simulate a real SSH banner. However, that is the limit of imitation; the script sends a protocol error message to the attacker’s client after the identification banner. It is possible to modify the script to perform any function required, so simulating different protocols, at least superficially, is relatively easy.

IV. Operation and Monitoring

Operation of the honeypot yielded several different types of messages. Ones that are significant indicated hosts on the network that were attempting to communicate to the honeypot. These messages were promptly investigated, as hosts that were functioning normally would had no need to communicate with devices that didn't exist. In general, it's OK to ignore UDP broadcasts.

Here are some examples of honeyd logs that are stored via SYSLOG:

```
Nov  1 17:41:03 SYSTEM honeyd[15775]: Connection to closed port: udp
(172.1.2.15:138 - 172.1.2.255:138)
```

This log entry is common on Microsoft Windows networks and can safely be ignored. It indicates that the device at 172.1.2.15 has attempted a UDP connection to all hosts on the network on port 138.

```
Nov  1 17:41:05 SYSTEM honeyd[15775]: Sending ICMP Echo Reply: 172.1.2.20 ->
172.1.2.19
```

This log entry is suspicious. It indicates that the device at 172.1.2.19 has tried to ping the honeypot at 172.1.2.20. Because the honeypot is configured to respond to ICMP (see line 5 in the configuration above), the honeypot logs a successful reply.

```
Nov  1 17:36:38 SYSTEM honeyd[15775]: Killing unknown connection: tcp
(172.1.2.20:3389 - 172.1.2.15:2400)
```

This log entry is also suspicious. It shows a connection attempt from 172.1.2.15 to the honeypot at 172.1.2.20 on port 3389 (Windows Remote Desktop Protocol). Port 3389 isn't configured in the honeyd.conf file for special handling, so the default tcp rule at line 3 dictates that a RST packet is sent to the source indicating that the service isn't present.

V. Results

In the exercise, the honeypot successfully detected intrusion attempts by the red team and provided enough information for the white team network engineers to filter the red team's traffic at the perimeter firewall. Deployment of a honeypot within a control or corporate network will require some baselining to differentiate traffic from misconfigured hosts from possible malicious activity, and will also require continuous scrutiny (or the establishment of alarms/alerts upon detection of anomalous traffic).