Dynamic Honeypot Construction

2nd Annual Alaska Information Assurance Workshop

Christopher Hecker
U. of Alaska, Fairbanks
9-5-2006

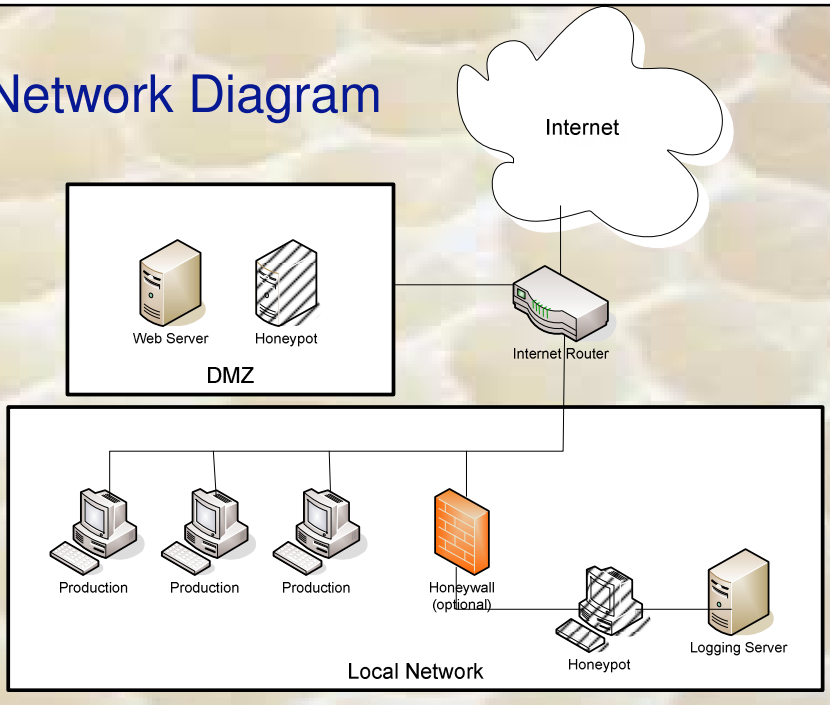
Presentation…

- Brief Introduction
- Project Overview
- Future Work
- References

Honeypots

- "A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource" [4] (basically, a computer/router, virtual or real, whose purpose is to be attacked and record as much information as possible).
- Since it has no production value, anything going to or from a honeypot is likely a probe, attack or compromise. This decreases the amount of logs to analyze.

Network Diagram



Types Of Honeypots

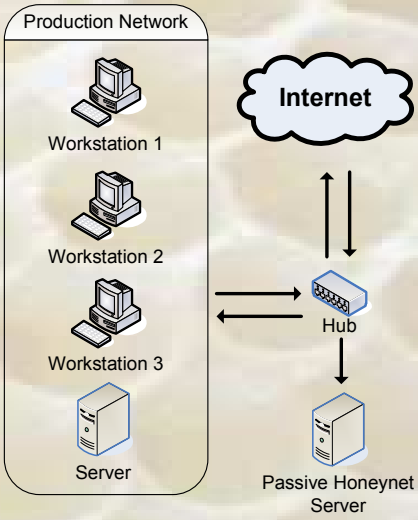
- Low-interaction
  - Scripts emulate services, applications, and OS.
  - Low risk and easy to deploy/maintain, but capture limited information.
  - Used by companies or individuals for IDS and network usage
- High-interaction
  - Real services, applications, and OS
  - Capture extensive information, but high risk and time intensive to maintain.
  - Used by researchers and security analysts for detecting new attacks

Dynamic Honeypots

- Two methods of mapping network – Passive and Active scanning
- Determines OS version, ports, and services running on scanned machine(s)
- Add new honeypots based on network configuration at scheduled times or as soon as a new machine is detected and scanned

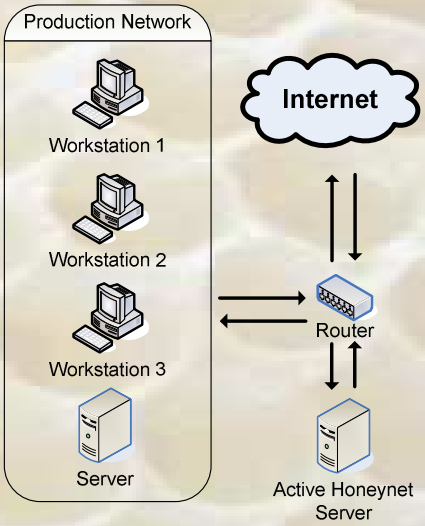
Passive Scan - Dynamic Honeypots

- Advantages
  - Determine network configuration without detection
  - Creates log of network usage and cycles
  - No extra bandwidth required
- Disadvantages
  - Takes a long time
  - Difficulty with routers/switches
  - Requires communication to determine open ports



Active Scan - Dynamic Honeypots

- Advantages
  - Short time frame
  - Determines OS and all open ports and services (to the best of its ability)
  - Few network hardware limitations (routers/switches)
- Disadvantages
  - Only recognizes ports open to network traffic (not individual machines)
  - All computers and ports are scanned (uses bandwidth)
  - Only recognizes new machines when scanning is initiated



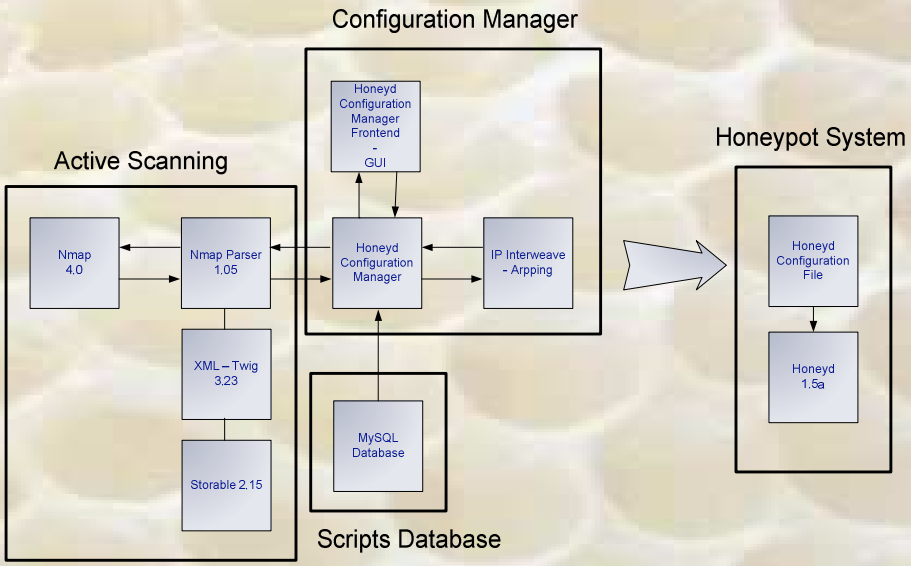
Project Overview – Active Scan using Honeyd

- Project - implement an low interaction, active scan dynamic honeypot system using Honeyd.
- Honeyd is a small daemon that runs both on UNIX-like and Windows platforms.
  - It is used to create multiple virtual honeypots on a single machine. Entire networks can be simulated using Honeyd.

Project Overview – Honeyd

- Honeyd emulates operating systems by responding with appropriate packets to Nmap and Xprobe fingerprinting packets. Thus the list of operating systems that Honeyd emulates can be found in *nmap.prints* and *xprobe2.conf*.
- Honeyd can be configured to run a range of services like FTP, HTTP, or SMTP.
- Add scripts to Honeyd which can provide a more in depth emulation of a given service. The depth of the emulation is only determined by the script provided.
- Honeyd allows a single host to claim as many as 65536 IP addresses

Project Overview



Project - Honeyd Configuration Manager

- Developed under Linux – Centos 4
- Programmed using Perl
- Development of Dynamic Honeypot system using Honeyd and Nmap
  - Nmap scans IP address or range
  - Nmap information is used real-time to make honeyd.conf file
    - MySQL is queried to find relevant emulation scripts
    - honeypots IP address is configured according to system administrator's specification
  - Honeyd is used to make Honeynet comparable to the network scanned
- GUI User Interface - Glade

MySQL Database

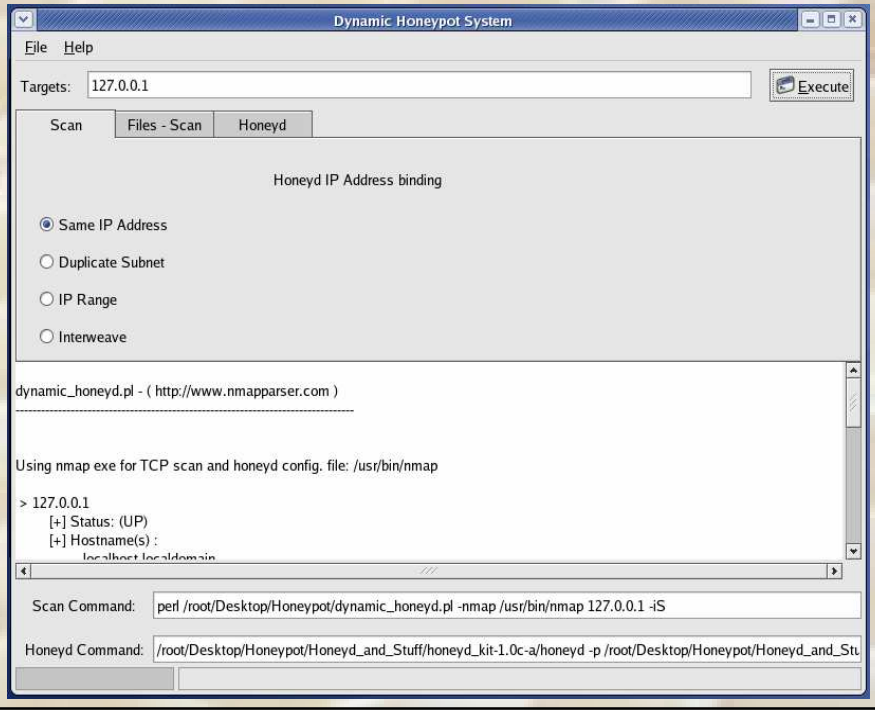
- Emulation script information stored in DB
  - allows user to add more scripts to database in specified format
  - allows Honeyd config. manager to query DB and find all relevant scripts for honeypots
  - Information:
    - OS Personality
    - Protocol
    - Port
    - Location

```
create Windows1
set Windows1 personality "Microsoft Windows 2003 Server or XP SP2"
set Windows1 default tcp action reset
set Windows1 default udp action reset
set Windows1 default icmp action open
add Windows1 tcp port 21 open
add Windows1 tcp port 23 "perl /root/Desktop/Honeypot/Honeyd_scripts/telnet-
emul/telnet/faketelnet.pl"
add Windows1 tcp port 25 open
add Windows1 tcp port 80 open
add Windows1 tcp port 110 open
add Windows1 tcp port 143 open
add Windows1 tcp port 5101 open
add Windows1 udp port 123 open
add Windows1 udp port 135 open
add Windows1 udp port 137 open
add Windows1 udp port 138 open
add Windows1 udp port 139 open
add Windows1 udp port 445 open
add Windows1 udp port 500 open
add Windows1 udp port 1067 open
add Windows1 udp port 1351 open
add Windows1 udp port 1900 open
add Windows1 udp port 4500 open
add Windows1 udp port 31337 open
set Windows1 ethernet "Intel Corporate"
bind 137.229.48.217 Windows1
```

Project Overview – GUI,

- Glade User Interface Builder for GTK+ and GNOME
- Programmed using C

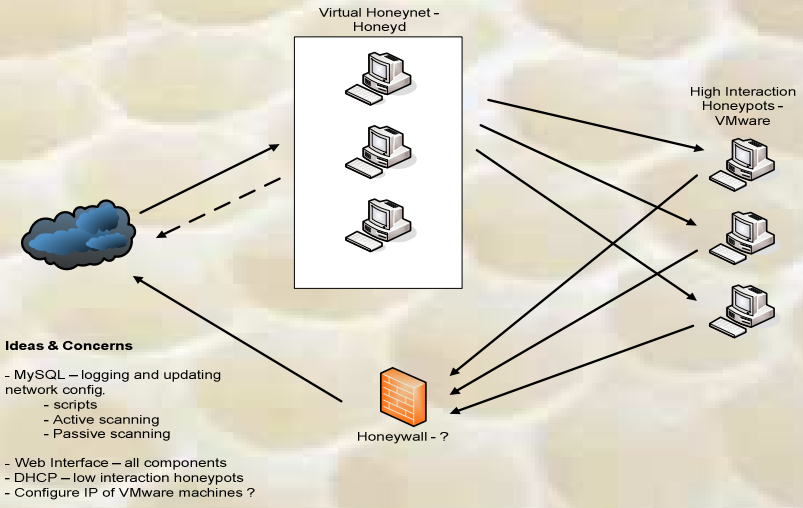
- Calls Honeyd Configuration Manager
- Allows user to setup up Honeyd honeypots from GUI



Future Work

- Integrate high interaction honeypots
- Incorporate network scanning/deployment with lag time
- Allow user to interact with database through the GUI – add/delete scripts
- Use combined Active and Passive Scanning methods
- Create standalone appliance with web interface

Possible Scenario



References

1) The Honeynet Project: Research Alliance - http://www.honeynet.org/index.html
2) http://www.honeyd.org/faq.php#list
3) http://www.insecure.org/nmap/nmap_documentation.html
4) http://www.securityfocus.com/infocus/1731
5) Honeypots for Windows by Roger A. Grimes, 2005
6) Honeypots: Tracking Hackers by Lance Spitzner, 2003

Questions?

http://assert.uaf.edu/

fscrh2@uaf.edu