

Honey
Pot

Mr. Manfred Hung

Mr. Alan Lam

Venue sponsored by

Chinese University of Hong Kong



Honeypot Hacker Tracking and Computer Forensics

Manfred Hung

manfred.hung@pisa.org.hk



Agenda

- Honeygot History
- Value of Honeygot
- Honeygot Technology
- Common Honeygot products/solutions
- Honeygot deployment tips
- Future of Honeygot



Intruders' Motivation

- High speed network/Internet bandwidth
- High speed processors
- Enormous disk storage
- Value of information stored
- Business/Political issues
- Script Kiddies, Worms ...

Value of Honeypot

- Research based Honeypot
 - Research the threats organization may face
 - Detect new threat
 - Deploy in University, Government, IT Vendor
- Production based Honeypot
 - Secure product environment by detect attack
 - Waste intruders' time on honeypot system
 - Deploy in customers' production net



History of Honeypot

- Real System
 - 1990 or before
- Network services simulation
 - 1998
- Virtual System
 - 1999



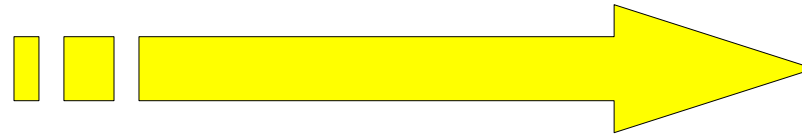
Honeypot Technology

- Real System
 - Unprotected network
 - Wide open firewall rules
 - Insecure system
 - Default install, no patch
 - Insecure setting

Honeypot Technology

- Network Service Simulation
 - NFR BOF, Sting, Honeyd
 - Unable to capture details information
 - Easy to deploy
 - Low interaction, lower risk, no real OS
- Virtual System
 - Honeynet, Decoy Server (ManTrap)
 - Able to capture details information
 - Complex to deploy
 - High interaction, higher risk, virtual OS

Honeypot Technology



NFR BOF
CyberCop Sting
Honeyd

Lower interaction

Lower risk

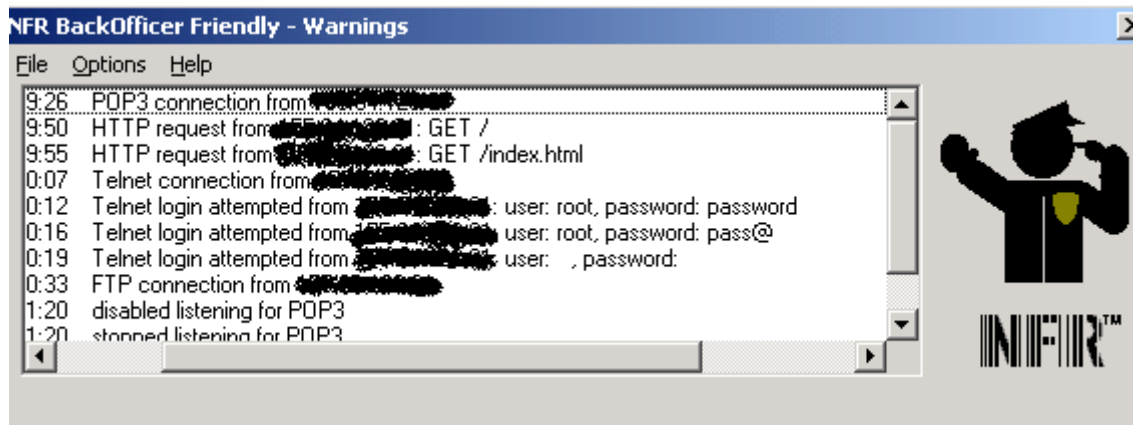
Symantec Decoy Server
HoneyNet

Higher interaction

Higher risk

NFR BackOfficer Friendly (BOF)

- Network services simulation
- Slow-interaction honeypot



The screenshot shows a window titled "NFR BackOfficer Friendly - Warnings" with a menu bar containing "File", "Options", and "Help". The main area is a log of network events:

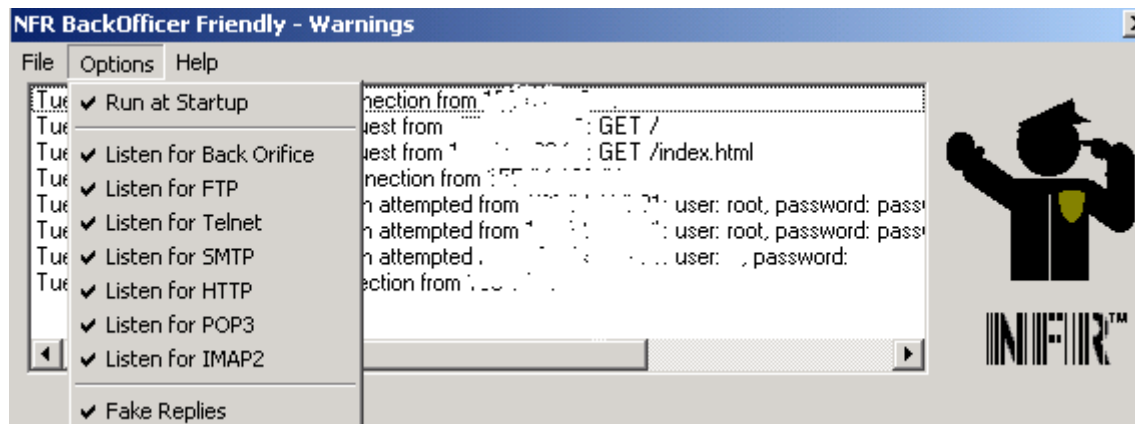
```
9:26 POP3 connection from [REDACTED]
9:50 HTTP request from [REDACTED]: GET /
9:55 HTTP request from [REDACTED]: GET /index.html
0:07 Telnet connection from [REDACTED]
0:12 Telnet login attempted from [REDACTED]: user: root, password: password
0:16 Telnet login attempted from [REDACTED]: user: root, password: pass@
0:19 Telnet login attempted from [REDACTED]: user: , password:
0:33 FTP connection from [REDACTED]
1:20 disabled listening for POP3
1:20 stopped listening for POP3
```

On the right side of the window, there is a logo for "NFR" featuring a stylized figure in a suit and hat, with the text "NFR™" below it.



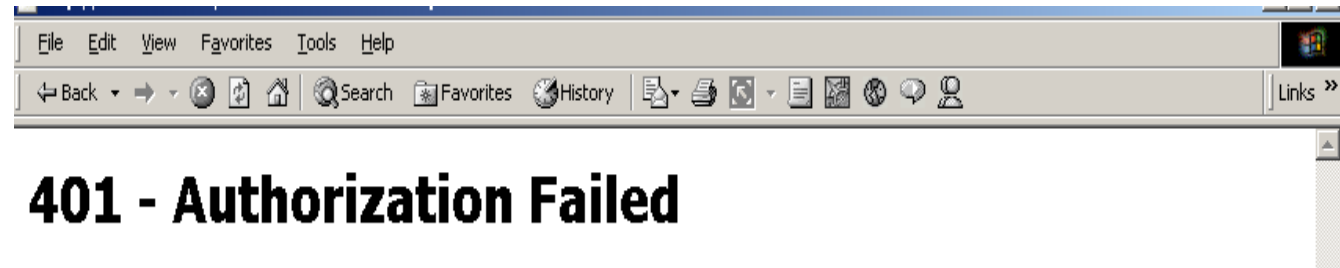
NFR BOF – Limited Services

- Limited Services
 - BO2K, FTP, Telnet, SMTP, HTTP, POP3, IMAP



NFR BOF – Slow Interaction

- Slow Interaction
 - HTTP – 401 Authorization Failed
 - FTP, SMTP, POP3 ... - 503 Service Unavailable



```
Connected to ...
503 Service Unavailable

Connection closed by remote host.
ftp> _
```



CyberCop Sting

- Network Service Simulation
- Low interaction honeypot

- Network device, Windows, Unix
- Limitation - Windows NT only

Honeyd

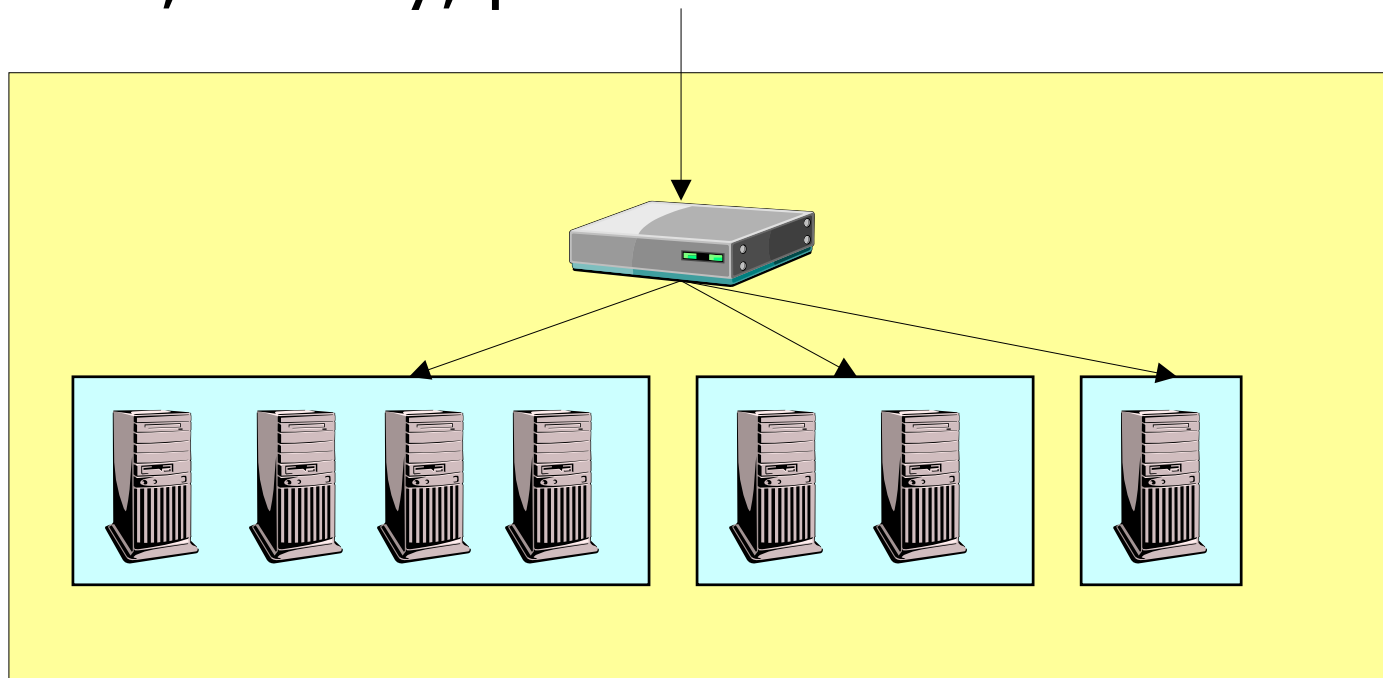
- Network Service Simulation
- Low interaction honeypot
- Network device, various OS system
- Resists fingerprinting



Honeyd – Route traffic

Blackholing

- Route net traffic
- TTL, latency, packet loss





Honeyd – ARP request

ARP spoofing

- Arpd monitor arp traffic
- Map non-exist IP addr. to honeyd MAC addr.

ARP proxy

- Static map IP addr. to honeyd MAC addr.

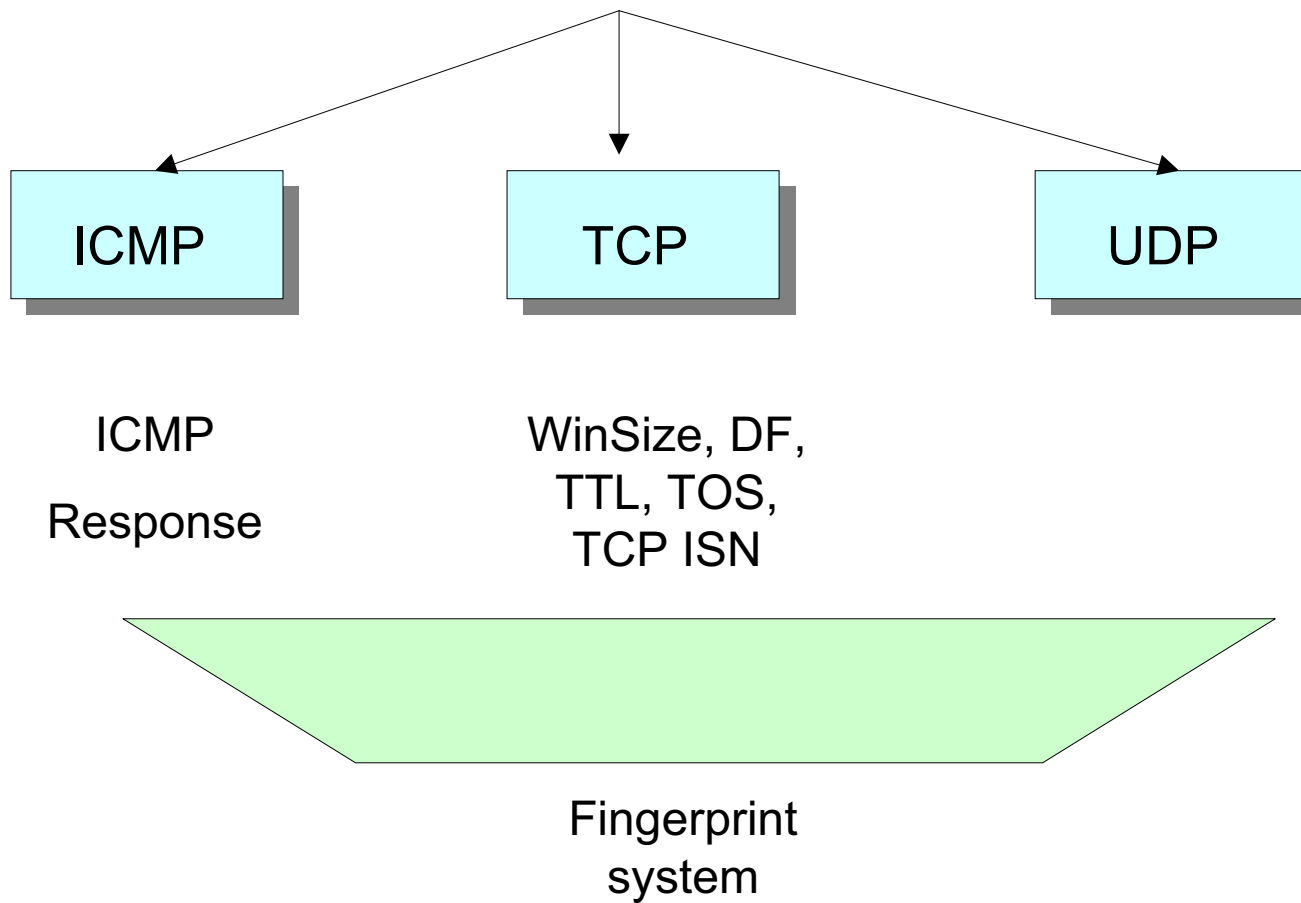
```
arp who-has 128.3.254.6 tell 128.3.254.68  
arp reply 128.3.254.6 is-at 02:07:01:00:01:c4
```

Honeyd – fingerprint

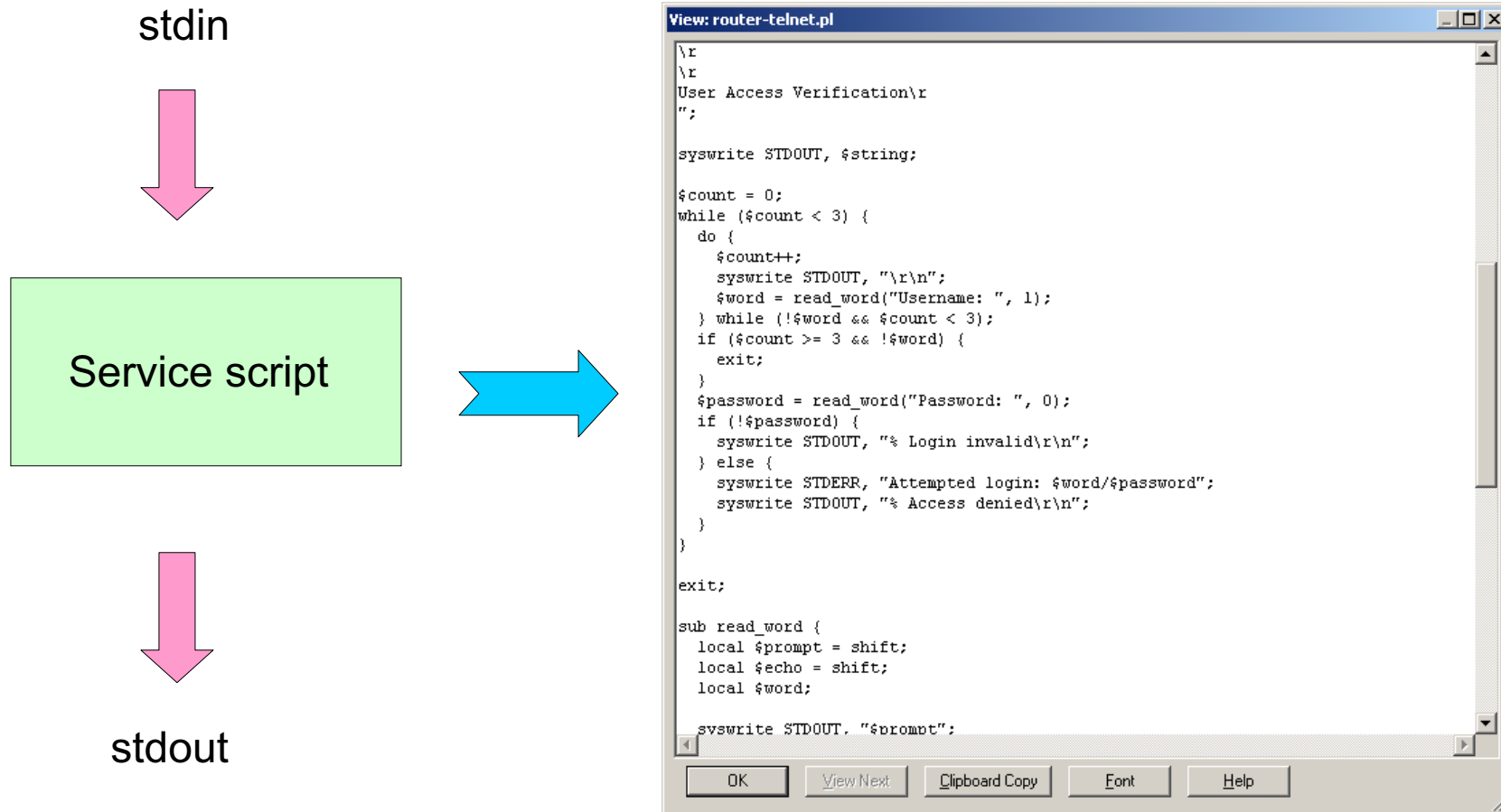
- Nmap, Xprobe
- Database of known OS signature

```
(The 1590 ports scanned but not shown below are in state: closed)
Port      State  Service
22/tcp    open  ssh
111/tcp   open  sunrpc
139/tcp   open  netbios-ssn
620/tcp   open  unknown
950/tcp   open  oftep-rpc
3306/tcp  open  mysql
5801/tcp  open  vnc-http-1
5901/tcp  open  vnc-1
6001/tcp  open  X11:1
10000/tcp open  snet-sensor-mgmt
32770/tcp open  sometimes-rpc3
Remote OS guesses: Linux Kernel 2.4.0 - 2.5.20, Linux 2.4.19-pre4 on Alpha
Nmap run completed -- 1 IP address (1 host up) scanned in 4 seconds
```

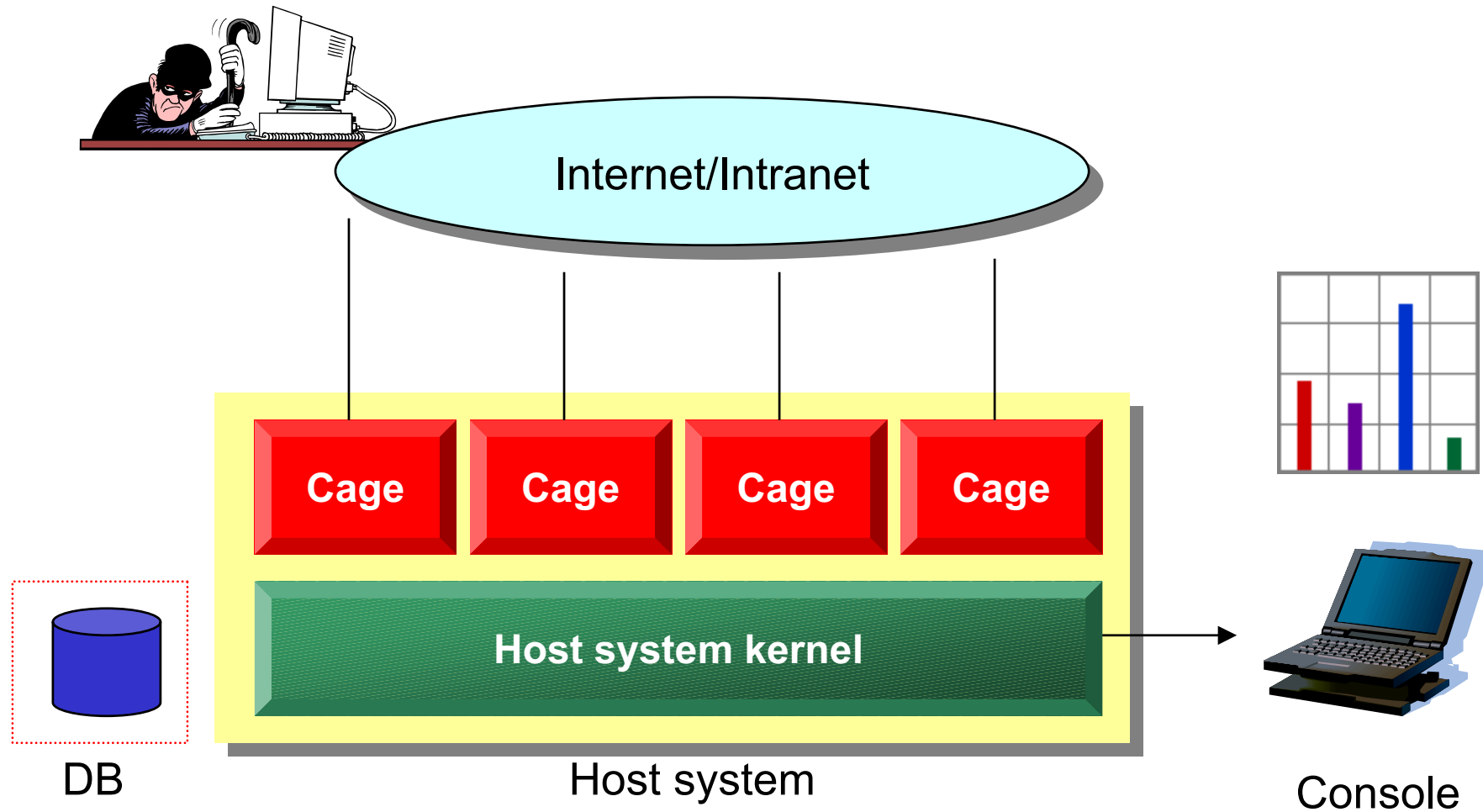
Honeyd – fingerprint



Honeyd – Interactive Response



Symantec Decoy Server (ManTrap)





Symantec Decoy Server

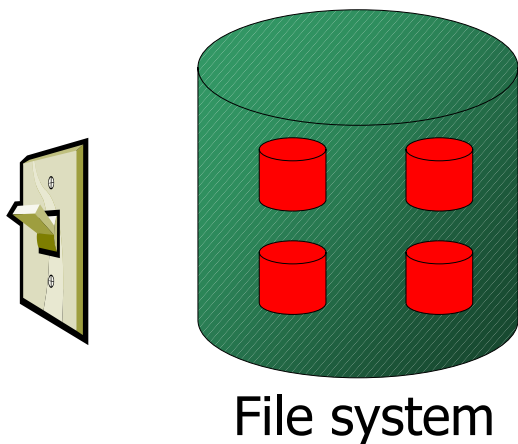
- Cage
- Content Management Module (CGM)
- Network sniffer
- Kernel – process create, File I/O
- Hardware token - iButton

Decoy Server – Data Control

- Intruder unable to access host system
- Multiple cage are under separate network
- Cage shutdown

```

mung@fw1:~$ df -k
Filesystem            1K-blocks      Used Available Use% Mounted on
/dev/hda2              75782272    48687040 23245660  68% /
/dev/hda1              101089       26655     69215   28% /boot
none                  119576       0         119576   0% /dev/shm
192.168.48.90:/home   74746460    65124596 5824904  92% /home
192.168.48.90:/pub1  38464340    33544472 2965964  92% /home/upload/pub1
mung@fw1:~$
  
```





Decoy Server – Data Capture

Kernel level data capture

- Network (sniffer)
- Process (PID, EUID, EGID)
- File I/O
- Device I/O



Decoy Server – Data Capture

Network capture

- Kernel level capture
- Session playback

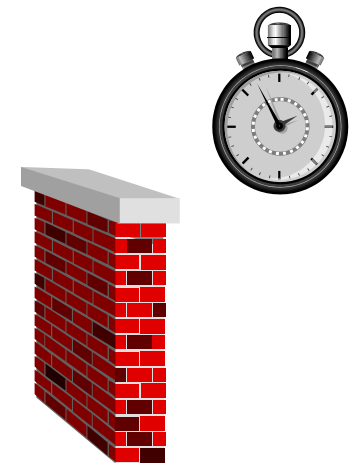


Decoy Server - Data Collect

- Content Generation Management (CGM)
- Alerts (e-mail, SNMP ...)
- Collected data integrity protection – iButton
- Secure copy (SCP)

Honeynet – Data Control

- Firewall – control inbound traffic
 - Limited access to pre-defined service
- Firewall – control outbound traffic
 - Allow outgoing traffic if not excess pre-defined no. of connection count
- Firewall – manual control
 - Allow console or remote control traffic



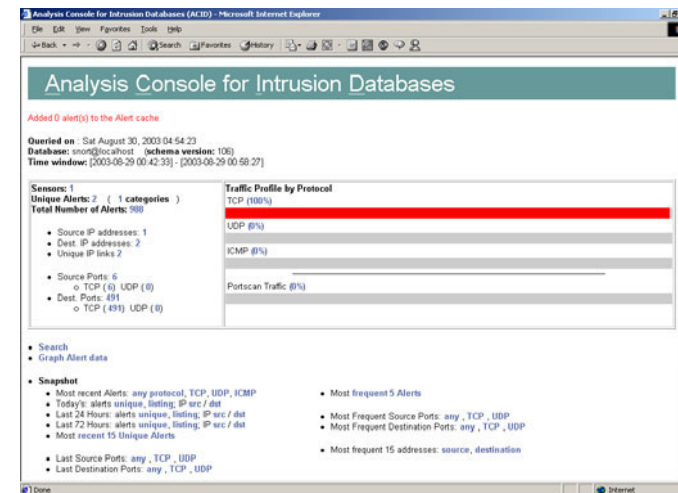


Honeynet – Data Capture

- Network level
 - Firewall
 - Network IDS
- System level
 - Trojan shell
 - System/Application log (ie. Syslog)
 - File I/O

Honeynet – Data Collect

- Firewall log
- IDS log (ie ACID)
- Trojan shell
- Syslog





Honeynet GenI Problem

- Fingerprint
 - Connection Limit
 - TTL
- Risk
 - Low connection limit not equal to safe
 - High connection limit not equal to unsafe

HoneyNet GenII - Improvement

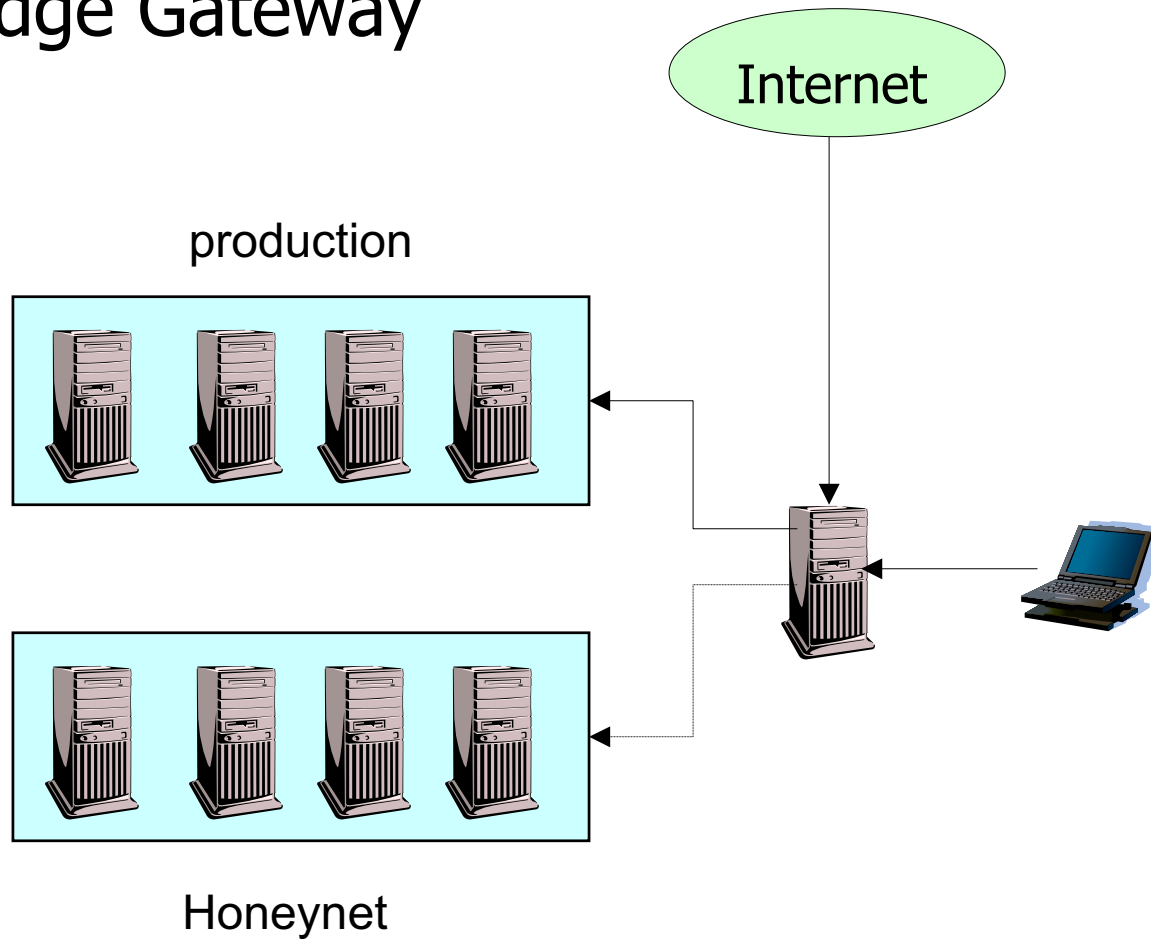
Data Control

- Connection Limitation
- Snort-Inline Drop
 - If intrusion occurs, drop
- Snort-Inline Replace
 - If intrusion occurs, replace attack action

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 53
(msg:"DNS EXPLOIT named";flags: A+;
content:"|CD80 EBD7 FFFFFFFF|/bin/sh";
replace:"|0000 EBD7 FFFFFFFF|/ben/sh");
```

Honeynet GenII - Improvement

Bridge Gateway





Honeynet GenII - Improvement

Data Capture

- Trojan shell
 - Fingerprint
- Capture at kernel module

Honeypot Deployment Tips

Placement

- outside or inside firewall
- Extensive probe

No. of honeypot system

- Map all unused IP addr., port to honeypot

Data in the honeypot system

- Network/System traffic, application data

Honeypot Deployment Tips

Security Data collected

- Modified activity log, wiping disk
- Syslog/SNMP Trap to other system
- Write-One Media (multi-session write)
- Content management
- Lookup source
- Collected data integrity
 - Hash checksum
 - Digital signed

Risk Associated with Honeypot

- Fingerprint
 - Fingerprint honeypot system
 - Based on OS, services running
 - Customization
 - Honey environment does not match real system
 - Victim standard system does not match honeypot
 - Apply similar production system setting
 - Match OS
 - Realism
 - Honeypot don't like real system
 - Monitor network/system activity
 - Real application and production data



Risk Associated with Honeypot

- Intruders' stepping stone
 - Firewall, NIDS, HIDS ...
- Response action
 - Automatically vs Manually

Honeypot Development

- Auto-build honeypot content
- Simulate real environment
 - Operation System
 - Application
 - Data
- GUI Management Console
- Centralized data collection
- User mode Linux (UML)



Thank You

Manfred Hung

manfred.hung@pisa.org.hk

Professional Information Security Association

www.pisa.org.hk



Honeynet

**A platform for studying Hacker Behaviors
and Computer Forensics**

Alan S H Lam

alan.lam@pisa.org.hk



Outlines

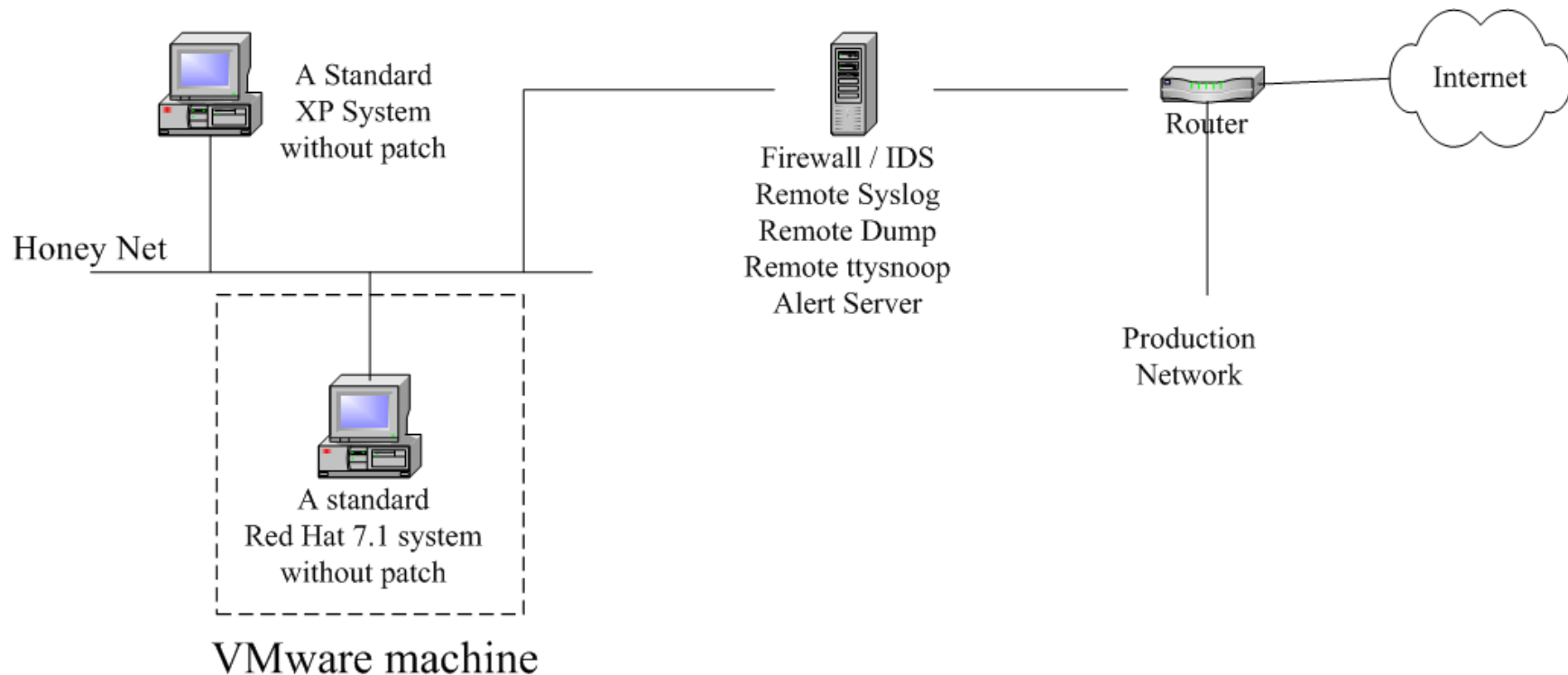
- Objectives of our Honeynet
- Implementation of our Honeynet
- Intruders' Activities and Forensics Techniques (with live demo)
- Deployment Tips
- Future Development
- Q & A



Objectives of our Honeynet

- To learn from the hackers
- To give early warning of potential attacks
- To collect research material for our computer forensic lab
- To improve our skill in security incident response

Existing Honeynet Network Infrastructure



Implementation

- Data Control
 - Egress filter rule
 - IPTable rule in firewall to drop or cut Honeypot traffic when
 - NIDS detects any attack originated from Honeypot
 - Packet rate higher than R
 - After N outbound connections from Honeypot
 - After M packets go through the HoneyNet
 - An alert message will be sent to the system admin when the connection is cut

Implementation (cont')

- Data Capture
 - Capture all network packets in/out the Honeynet
 - Capture hackers' keystroke by a trojaned login shell in Honeypot
 - Remote syslog
 - Dump backup
 - Firewall and SNORT NIDS log
 - All data captured are stored in the firewall host

Intruders' Activities

- Identify/locate the victim by some scanning tools
- Break-in the victim through system security holes. The following vulnerabilities were used by the hackers to break-in our Honeynet.
 - sshd CRC32 Overflow
 - Buffer overflow in openssl
 - WU-FTP RNFR ././ attack
 - execve/ptrace race condition
 - Microsoft's DCOM RPC (W32/BlasterA/D Worm)

Intruders' Activities (cont')

- After break-in, the hackers may
 - Install rootkit to setup backdoor, sniffer, IRC proxy, or streaming server
 - Use victim as a stepping stone to find and attack other victims
 - Fix the victim vulnerability and undo other hackers jobs
 - Send back the victim information through e-mail
 - Propagate the attack to other victims
 - Deface/remove victim web page

Forensic Tools

- scp, dd, tar, nc
- tcptrace, tcpdump, snort
- ps, netstat, lsof, fuser, kill -STOP, pcat, ltrace, strace, /dev/kmem, coreography
- /proc directory
- find, ldd, strings, gbd, od, bvi, icat, elfsh
- Coroner's Toolkit (TCT), Chkrootkit

Deployment Tips

- Do not deploy your Honeynet unless you are sure about your data control
- Start with tight data control first
- Capture data at different levels
- Make sure your Honeynet does not violate your company policy



Future Development

- Enhance the HoneyNet to include more other OS systems
- “Honey” the HoneyPots so as to attract different classes of hackers (e.g. building a web portal or on-line bank)
- Set up a forensic lab



Q & A

- Questions
- Comments
- Suggestions

Thank You

alan@ie.cuhk.edu.hk

Professional Information Security Association

www.pisa.org.hk