Cluster Security

- Security v performance and usability
- Aims:
  - Minimize susceptibility to outside attack
  - Minimize impact if hacker gains access to a node
- Approaches:
  - Stop unwanted packets before reach cluster
  - Stop unwanted packets on cluster

Packets on Cluster

- Locking down individual node software
  - Often distributions come with insecure parameters
- Disable unnecessary services
  - Default often runs almost everything - NFS, Httpd (WWW), other daemons
  - The more applications that are running with open ports, the more points of attack for hackers
  - Step 1
    - Check services using *ps* and ports used using *netstat*
    - *nmap* shows open ports
  - Step 2
    - Examine service startup scheme (often GUI for this)
    - Understand how to disable
  - Step 3
    - Configure all nodes identically

Securing Remaining Services

- Limit domains/machines which may connect to services
  - Default usually allows access from anywhere
  - Various services have various methods of restricting access
- Example: NFS access is controlled by /etc/exports

Service	Description	Allow
ssh	Secure remote login	Internet
nfsd	Share file system over network	Cluster Nodes
named	DNS server – serves mapping from name to IP address	Site machines
httpd	Web Server	Site machines
scheduler	Batch job scheduler	Login nodes only

Watch Security Updates

- Distributions usually do not contain known security holes
- Usually these are discovered over time, alerts and patches are issued to remove the hole
- Need to monitor security websites, mailing lists. distribution vendors security pages
  - www.securityfocus.org (fast response to new exploits)
  - www.cert.org (very complete index of security problem/patches)

Packets Entering Cluster

- Firewalls
  - Mechanism to allow inspection of packets
  - Decision rules on whether to admit packets based on source and destination
- Some options
  - Hardware on network
    - Easier to configure
    - Cost, slower response to security hole fixes
  - Software
    - Needs configuration

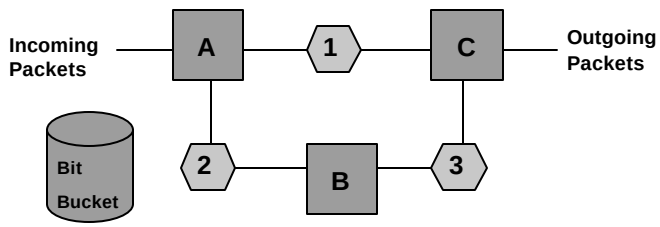
Where to Place Firewall

- Between uplink and entire site
  - Not so good as clusters have special needs
- Run firewall software on individual nodes
  - Unnecessary complexity
- Place in front of part or all of cluster
  - Entire cluster, computer nodes, management noes, server nodes etc
- For simplicity assume between cluster and non-cluster machines

Iptables / netfilter

- Assume the cluster is in 192.168.13.0/24 range
- Use a Linux machine with two NICs as router
  - One NIC on cluster network
  - One on the rest of the site network
  - Use *routed* or *gated* to route packets from one NIC to other
- As packets make their ways through the router they can be inspected and tested at various points
- Based on outcome packets may continue, be tested more, be altered, be discarded
- Iptables calls these checkpoints *chains*

Routing and netfilter points



Kernel space routing decision points	Netfilter points (chains)
A – incoming routing decision B – local machine process space C – postrouting decision	1. FORWARD netfilter table 2. INPUT netfilter table 3. OUTPUT netfilter table

Rules

- Ecah rule in chain is of form
  - If packet matches <x> the perform action <y>
- Packets start at first rule, and continues until last rule, unless it matches some <x>
  - If matches some <x> the corresponding action <y>is taken
    - Often action is to allow packet to continue past chain
  - If it makes it past all the rules some default action is taken
- Often procedure is
  - Block all network traffic by default – i.e. if no <x> matched
  - Allow packets we want through
- Can examine source, destination, protocol, service type (port)

Example

- Place firewall between cluster and Internet
- Drop all packets except sshd and httpd traffic for cluster nodes
- We can inspect current state of rules with
  iptables –L
- If no rules are defined it will be similar to

```
Chain INPUT (policy ACCEPT)
target     proto    opt      source   destination

Chain FORWARD (policy ACCEPT)
target     proto    opt      source   destination

Chain OUTPUT (policy ACCEPT)
target     proto    opt      source   destination
```

Changes to rules

- We not change the default rules as folows

```
iptables –P FORWARD DROP
iptables –A FORWARD –s 192.168.13.0/24 –d 0.0.0.0 –j ACCEPT
iptables –A FORWARD –s 0.0.0.0 - -protocol tcp - -dport 22 \
        -d 192.168.13.0/24 –j ACCEPT
iptables –A FORWARD –s 0.0.0.0 - -protocol tcp - -dport 80 \
        -d 192.168.13.0/24 –j ACCEPT
```

- The first sets the default as drop
- The 2nd allows all cluster traffic out
- The 3rd and 4th allow ssh and http traffic destined for cluster in

State after changes

```
>iptables –L
Chain INPUT (policy ACCEPT)
target     proto    opt      source  destination

Chain FORWARD (policy ACCEPT)
target     proto    opt      source           destination
ACCEPT     all      --       192.168.13.0/24  0.0.0.0
ACCEPT     all      --       0.0.0.0          192.168.13.0/24 tcp dpt:22
ACCEPT     all      --       0.0.0.0          192.168.13.0/24 tcp dpt:www

Chain OUTPUT (policy ACCEPT)
target     proto    opt      source  destination
```

Hardware Firewalling

- Use specialized hardware device
  - Adv: Relative ease of use, vendor support
  - Disadv: slower response to security holes, cost
- Firewall using router
  - Many routers can be configured as hardware firewalls
  - This is the approach used for fianna
  - A Foundry BigIron Gigabit switch/router is used to route traffic to the cluster
  - This implements firewalling of the cluster, and only permits access from selected domains, and using selected protocols