OpenBSD/x-ray
OpenBSD on medical x-ray machines

Henning Brauer
<info@henningbrauer.com>
Medical Environment

• Certification, certification, certification
• Major changes require re-certification
  - Fixing problems is hard
• Long equipment lifetime
Medical Environment

• No remote access to live systems
  – Limited access in maintainance mode
  – Cannot legally acquire any data
    • Not even a typical request rate

• No remote updates

• Incremental development does not work
Medical environment

• Field engineers are x-ray engineers
• Not IT people, not sysadmins
• Can replace components
  – But not debug network problems or the like
Medical Environment

• Patient data
  – A stolen credit card number can be voided
  – Medical fact (broken leg, breast cancer) is hard to change
X-ray machine architecture

- Compact, mobile systems to fully equipped x-ray rooms
- X-ray (picture) and fluorescopy (movie)
- Several networked systems
- Connected to hospital network
X-ray machine architecture

- Generator, x-ray tube and control circuits
- Image sensor
  - digital and wireless on new ones
- Table, wall mount
  - Usually motorized
- Foot switches
- Workstation for the operator
X-ray machine architecture

- Radiology Information System (RIS)
- Patient data with x-ray request is sent to the machine
- Patient data with x-ray images is sent back
  - Review, diagnosis, archiving
X-ray machine architecture

• New x-ray machines have digital sensors
  – Connected via Ethernet

• Several ways to retrofit older systems
  – Common: film replaced by a cassette, which goes into a reader, connected via Ethernet
  – Sometimes the reader is connected to the outside network
X-ray machine architecture

● 3rd party components

● Often little to no competition
  – Small market
  – high development and certification costs

● Old protocols don‘t die easily
  – FTP is alive
X-ray machine architecture

- Internal network and external network are the same Layer 2 network
- Interrupted data transfer from sensor: bodily injury!
  - x-ray process has to be repeated
X-ray machine architecture

- Want to shield the internal network from the external
- Philips has an OpenBSD firewall for that
  - For 10 years already
- embedded i386, 4-5 LAN ports, bridge, pf
OpenBSD/x-ray

- Custom ramdisk
  - System can be powered off any time
  - Everything needed for bridge and pf
  - ssh and some basic tools
OpenBSD/x-ray

• no persistent config on the ramdisk

• „magic“ IP address

• Management system configures
  – Including pf and bridge rules
OpenBSD/x-ray

- pf provides everything one can wish for to filter IP traffic
- but ARP...
ARP

- IP to MAC address
- Outside systems must not claim internal IPs
- Static IP-MAC mappings on all internal systems not feasible
## ARP

<table>
<thead>
<tr>
<th>Off</th>
<th>Len</th>
<th>Field</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>HTYPE</td>
<td>Hardware Address Type</td>
<td>Ethernet: 1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>PTYPE</td>
<td>Protocol Address Type</td>
<td>IPv4: 0x0800</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>HLEN</td>
<td>Hardware Address Length</td>
<td>Ethernet: 6</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>PLEN</td>
<td>Protocol Address Length</td>
<td>IPv4: 4</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>OPER</td>
<td>Operation</td>
<td>request: 1, reply: 2</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>SHA</td>
<td>Sender Hardware Address</td>
<td>MAC address</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>SPA</td>
<td>Sender Protocol Address</td>
<td>IP address</td>
</tr>
<tr>
<td>18</td>
<td>6</td>
<td>THA</td>
<td>Target Hardware Address</td>
<td>MAC address</td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>TPA</td>
<td>Target Protocol Address</td>
<td>IP address</td>
</tr>
</tbody>
</table>
ARP request

- 10.0.0.1 (MAC: 11:22:33:44:55:66) wants to talk to 10.0.0.2
- Request sent to ff:ff:ff:ff:ff:ff (broadcast)

<table>
<thead>
<tr>
<th>OPER</th>
<th>1 (request)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPA</td>
<td>10.0.0.1</td>
</tr>
<tr>
<td>THA</td>
<td>ignored</td>
</tr>
<tr>
<td>TPA</td>
<td>10.0.0.2</td>
</tr>
</tbody>
</table>

- 10.0.0.2 learns: 10.0.0.1 is 11:22:33:44:55:66
ARP reply


<table>
<thead>
<tr>
<th>OPER</th>
<th>2 (reply)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHA</td>
<td>77:88:99:aa:bb:cc</td>
</tr>
<tr>
<td>SPA</td>
<td>10.0.0.2</td>
</tr>
<tr>
<td>TPA</td>
<td>10.0.0.1</td>
</tr>
</tbody>
</table>

- 10.0.0.1 learns: 10.0.0.2 is 77:88:99:aa:bb:cc
ARP filter

• Need to filter ARP to control MAC learning
  – block arp from outside with any inside MAC/IP in SHA/SPA

• pf doesn‘t even see ARP traffic

• the bridge filter is the obvious place
bridge arpfilter

- new bridge_arpfilter() is just ~40 LOC
- ioctl, headers etc add little
- the rule parser in ifconfig is a rather strange beast
- entire diff with manpage just over 400 lines
example

ifconfig bridge0 rule block in on em0 arp spa 10.0.0.1
ifconfig bridge0 rule block in on em0 src 11:22:33:44:55:66
ifconfig bridge0 rule block in on em0 \n  arp request sha 11:22:33:44:55:66
ifconfig bridge0 rule block in on em0 \n  arp reply sha 11:22:33:44:55:66
bridge arpfilter

- Reverse ARP can be matched likewise
- „rarp“ keyword instead of „arp“
bridge(4) vs switch(4)

• bridge needs to die, switch is the future
• Implementing filters in switch was out of scope for this project
• Layer 2 filters make sense without bridge or switch
Generic Layer 2 Filters

• Should have layer 2 filtering capabilities on any Ethernet interface

• Want logging
  – bridge filters don’t have that really
  – Adding proper logging is pretty involved
Generic Layer 2 Filters

- pflog already fits the bill
- a lot more infrastructure is already there in pf
- pf doesn’t even see non-IP packets
  - Entry points in `ip_input() / ip_output()` and `ip6_input / ip6_output()`
pf and Layer 2

- Could add new entry points lower in the stack
- Rules could even combine layer 2 and higher layers
- But it gets nasty quickly
pf and Layer 2

- rule with MAC address matching and non-Ethernet packets?
- Rule with IP matching and non-IP packets?
- moving entry points is tricky
pf and Layer 2

- Re-use pf code, but separate L2 ruleset?
- New section in pf.conf or entirely separate?
- L2 filters independent from bridge(4) make the transition to switch(4) easier
Lessons learned

no obvious signs of NSA / BND / ... tampering in my phone
Thanks!

• Philips
  - for being very open and allowing me to present this extraordinary OpenBSD use case

• Holger Mikolon
  - who helped a lot with the paper and was a fantastic host at Philips
Questions?