Layer 2 filtering
and transparent firewalling

Cédric Blancher <sid@mismag.com>

July 11, 2003
Why focus on layer 2

br-nf and Netfilter: bridged firewall
  - Bringing frames content to Netfilter’s framework
  - Applications

ebtables: layer 2 filtering
  - Architecture and framework
  - Functionalities

Applications
  - brouter
  - ARP traffic check
Why focus on layer 2

- br-nf and Netfilter: bridged firewall
  - Bringing frames content to Netfilter’s framework
  - Applications

- ebtables: layer 2 filtering
  - Architecture and framework
  - Functionalities

- Applications
  - brouter
  - ARP traffic check
Why filtering at layer 2?

- Enforce low level network access control
- Enforce security while expanding network
- Enforce coherent layer 2 and 3 informations
- Mitigating layer 2 attacks
- Preventing layer 2 cover channels
Practical examples

- Adding WiFi to an existing LAN
- Protecting CE stuff with silly network connectivity
- Adding new security zones to an existing LAN
- Dealing with connected roaming equipments (laptop, handled)
- etc.
Why focus on layer 2

- br-nf and Netfilter: bridged firewall
  - Bringing frames content to Netfilter’s framework
  - Applications

- ebtables: layer 2 filtering
  - Architecture and framework
  - Functionalities

- Applications
  - brouter
  - ARP traffic check
Goals

- Adding access control
- Avoid existing network disturbance
- Bridged firewall
What is a bridged firewall?

▶ It’s a bridge
▶ It’s a firewall
▶ Applause!
How does it work

- Bridge frames payload is extracted
- Payload is sent through firewalling framework
- If accepted, frame is sent through the bridge
- No routing, no IP: Transparent firewall
Set it up for Linux

- Use br-nf patch
- Use Netfilter
- You’re done
Why is it good for us

- We have firewalling stuff
- It does not break local addressing, DHCP and LAN stuff
- We can set firewall up with minimum administrative pain

^as long as we properly filter those protocols
Drawbacks

- Limited to firewalling framework capabilities
- What if other layer 3 protocols?
- See latest NetScreen advisory
Why focus on layer 2

br-nf and Netfilter: bridged firewall
  ▶ Bringing frames content to Netfilter’s framework
  ▶ Applications

ebtables: layer 2 filtering
  ▶ Architecture and framework
  ▶ Functionalities

Applications
  ▶ brouter
  ▶ ARP traffic check
Layer 2 filtering

Bridged firewalls are good but limited

- We need to deal with unsupported layer 3 protocols
- We may need to filter those protocols
- We may need to enforce layer 2 ACLs
- We need to layer 2 firewalling framework
ebtables provides L2 filtering for Linux based bridges

- L2 filtering: ethernet, 802.1q, 802.3 in progress
- Limited L3 filtering: ARP, IP
- Full IP filtering through Netfilter
- L2 NAT capabilities
- Both logical and physical in/out filtering
- logging, accounting, marking
- brouter capabilities
- iptables like configuration :/
- Valuable framework for low level filtering and manipulation
Layer 2 filtering

ebtables framework

- Very similar to Netfilter's one
- broute, nat and filter tables
- chains (BROUTING, PREROUTING, INPUT, FORWARD, OUTPUT)
- user chains, rules, insert/append/delete and cool
Not clear huh!

- Frames dropped in BROUTING are sent to routing process
- Frames sent to FORWARD are bridged frames
- Frames sent to INPUT are (MAC) dsted to local
- Frames sent to OUTPUT are (MAC) srced from local
- Covers near all situations
- Beware: ebtables only works on bridged interfaces!
Netfilter IP framework interaction
Well, you think it is complicated enough?

- br-nf patch is part of ebtables framework
- Netfilter sees bridged frames in order to provide full IP filtering
- Can lead to *very* complicated configuration
Layer 2 filtering

Frames through the bridge
Frames enforced to physical interfaces
So?

- We have our layer 2 filtering
- We have our transparent filtering
- We can still route or bridge as we like
- Cool stuff, but tricky to figure out
Why focus on layer 2

br-nf and Netfilter: bridged firewall
  - Bringing frames content to Netfilter’s framework
  - Applications

ebtables: layer 2 filtering
  - Architecture and framework
  - Functionalities

Applications
  - brouter
  - ARP traffic check
brouter stuff

- LAN destined frames are bridged
- Others are routed
Easy stuff

- `iptables -t nat -A POSTROUTING -s 172.16.1.0/24 -d 172.16.1.0/24 -j ACCEPT`

- `iptables -t nat -A POSTROUTING -s 172.16.1.0/24 -j MASQUERADE`

Now we can add filtering stuff, within Netfilter or `ebtables`
ARP check

- ARP is weak
- We have to enforce some security
What we can do

- `iptables -A FORWARD -p arp --arp-htype ! 1 -j DROP`
- `iptables -A FORWARD -p arp --arp-ptype ! 0x800 -j DROP`

- Implement an ARP purity matches
  - MAC src is the same within ethernet header and ARP payload
  - MAC dst is the same within ethernet header and ARP payload for answers
  - ARP message length is OK
Other things we can do

- "Port security" like setup using physical iface filtering
- MAC level ACLs or PVLAN like setup
- 802.1q trunks filtering
- Some cleanup for unused L2 and L3 protocols (802.3, Appletalk, NetBEUI, etc.)
- Advanced routing using ebtables NAT stuff
- QoS based on L2 stuff using marks
- etc.
In the end

- ebtables is a powerful tool
- ebtables may be difficult to set up
- some features must be developed (802.3)
Thanks

<Spam>

► Bart De Schuymer for ebtables and help
► Renaud "Nessus Méyawalpa" Deraison
► Nico "In Cisco we trust" Fischbach
► Laurent "T’es GNU sous tes vêtements ?" Oudot
► MISC magazin people and guys at Bashl0rdz.com
► Niels Provos and Air Moutane for this great track
► Audience for its patience

</Spam>

^Are you GNaked ?