

Network security

Exercise 9

How to build a wall of fire – Linux Netfilter

Tobias Limmer

Computer Networks and Communication Systems
Dept. of Computer Sciences, University of Erlangen-Nuremberg, Germany

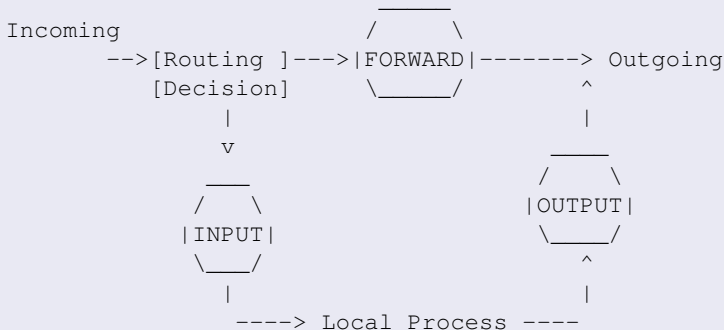
14. – 18.01.2008

Introduction to netfilter

- used for packet filtering in Linux (→ firewalls!)
- provides hooks in *Linux kernel* to intercept and manipulate network packets
- MANY extensions available
- some history:
 - started 1998 by Russel, chairman is now Harald Welte
 - several different command-line interfaces:
ipfwadm (2.0) → ipchains (2.2) → iptables (≥ 2.4)
 - got injunction against Sitecom Germany by Munich District Court in 2004

iptables

- command-line tool to insert and delete rules from packet filter table
- several chains available (simple version):



- attention: rules are volatile and will be lost upon reboot

iptables - rule management

- so, how do we insert rules into those tables?
- Example:

```
iptables -A INPUT -s 192.168.74.0/24 -j DROP
```

- general command structure:

```
iptables <cmd> [<chain>] [<filter rules>] [-j <action>]
```

- commands: append rule (“-A”), insert rule (“-I”), delete rule (“-D”), replace rule (“-R”), list rules (“-L”), flush rules (“-F”)
- chains: INPUT, OUTPUT, FORWARD
- filter rules: source/destination ip (“-s/d <ip>”), protocol (“-p <tcp,udp,icmp,...>”), interface (“-i/-o <iface>”), ...
- actions: ACCEPT, DROP, RETURN, jump to user-defined chain, ...

iptables - chain management

- default chains: INPUT, OUTPUT, FORWARD
- user-defined chains can be added/removed!
- Example:

```
iptables -N wwwserver
iptables -A wwwserver -p tcp --dport 80 -j ACCEPT
iptables -A wwwserver -j DROP

iptables -A FORWARD -d 192.168.74.23 -j wwwserver
```

- available commands: add chain (“-N <chain>”), remove chain (“-X <chain>”), set default policy for chain (“-P <chain> <DROP/ACCEPT/...>”, only valid for standard chains), flush rules in chain (“-F <chain>”)

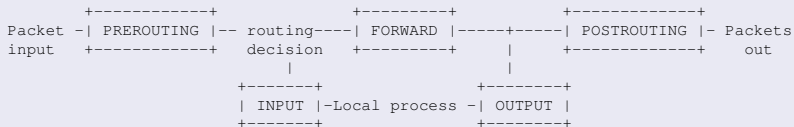
iptables - connection tracking

- stores information about state of connections in memory
- also application layer inspection for FTP, TFTP, IRC, PPTP, ...
- valid states: NEW, ESTABLISHED, RELATED, INVALID
- example:

```
iptables -P INPUT DROP
iptables -A INPUT -m state --state RELATED,ESTABLISHED
                                                    -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -p tcp --dport 21 -m state --state NEW
                                                    -j ACCEPT
iptables -A INPUT -j REJECT
```

- very fast implementation in kernel, is almost always used in firewall scripts

iptables - NAT 2



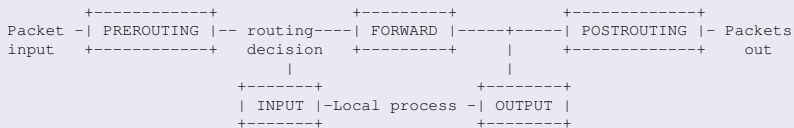
- POSTROUTING chain enables source NAT:

```
iptables -t nat -A POSTROUTING -s 172.17.0.0/16
-o eth0 -j SNAT --to-source 131.188.37.1
```

- PREROUTING chain enables destination NAT:

```
iptables -t nat -A PREROUTING -p tcp --dport 80
-j DNAT --to-destination 172.17.0.22:80
```

iptables - NAT 3



- what does the following line? 😊

```
iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80
                                           -j REDIRECT --to-port 80
```

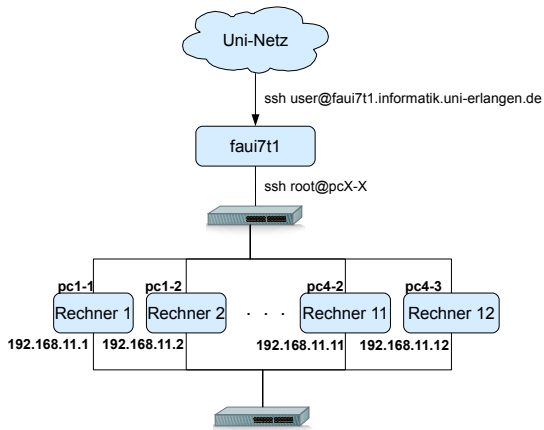
- simple masquerading (same as source NAT)

```
iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
                                           --to-ports 31000-60000
```

iptables - advanced routing

- not complicated enough? 😊
- iptables offers MUCH more features in form of extensions
- see the Kernel Packet Traveling Diagram:
<http://www.docum.org/docum.org/kptd/>
- matching extensions: owner, recent, limit, hashlimit, mark, ...
- action (usually called target/jump) extensions: classify, masquerade, queue, ulog, ...
- advanced rate limiting and shaping using the intermediate queue device (IMQ)

Netzstruktur



available computers: pcX-1, pcX-2, pcX-3 $X \in [1; 4]$



Some references (also used for graphs)

- **Linux 2.4 Packet Filtering HOWTO** from Rusty Russell
(<http://netfilter.org/documentation/HOWTO/packet-filtering-HOWTO.html>)
- **Linux Advanced Routing and Traffic Control HOWTO**
(<http://lartc.org/lartc.html>)
- **Iptables Tutorial 1.2.2** (<http://iptables-tutorial.frozentux.net/iptables-tutorial.html>)
- **Kernel Packet Traveling Diagram**
(<http://www.docum.org/docum.org/kptd/>)