Firewalls, IPsec and Linux

by

Harald Welte <laforge@netfilter.org>

Contents

Introduction

What this is:

- A broad overview about the advanced Linux networking features
- Intended for a network savyy audience that has little Linux background

What this presentation is not:

- A tutorial on how to use iptables, tc, iproute2, brctl
- An introduction into the cool code we write every day ;)

It will try to show you what you can do with Linux networking, not how.

Firewalls, IPsec and Linux

Introduction

Linux and Networking

- Linux is a true child of the Internet
- Early adopters: ISP’s, Universities
- Lots of work went into a highly scalable network stack
- Not only for client/server, but also for routers
- Features unheared of in other OS’s

Introduction

Did you know, that a stock 2.6.5 linux kernel can provide

- a stateful packet filter ?
- fully symmetric NA(P)T ?
- policy routing ?
- QoS / traffic shaping ?
- IPv6 firewalling ?
- packet filtering, NA(P)T on a bridge ?
- layer 2 (mac) address translation ?

If not, chances are high that this presentation will tell you something new.

Netfilter Hooks

- What is netfilter?
  - System of callback functions within network stack
  - Callback function to be called for every packet traversing certain point (hook) within network stack
  - Protocol independent framework
  - Hooks in layer 3 stacks (IPv4, IPv6, DECnet, ARP)
  - Multiple kernel modules can register with each of the hooks

Traditional packet filtering, NAT, ... is implemented on top of this framework

Can be used for other stuff interfacing with the core network stack, like DECnet routing daemon.

IP tables

- Packet selection using IP tables
  - The kernel provides generic IP tables support
  - Each kernel module may create it’s own IP table
  - The three major parts of 2.4 firewalling subsystem are implemented using IP tables
    - Packet filtering table ’filter’
    - NAT table ’nat’
    - Packet mangling table ’mangle’
  - Could potentially be used for other stuff, e.g. IPsec SPDB

IP Tables

- Managing chains and tables
  - An IP table consists out of multiple chains
  - A chain consists out of a list of rules
  - Every single rule in a chain consists out of
    - match[es] (rule executed if all matches true)
    - target (what to do if the rule is matched)

matches and targets can either be builtin or implemented as kernel modules

  - The userspace tool iptables is used to control IP tables
    - handles all different kinds of IP tables
    - supports a plugin/shlib interface for target/match specific options

Connection Tracking Subsystem

- Connection tracking...
  - implemented seperately from NAT
  - enables stateful filtering
  - protocol modules (currently TCP/UDP/ICMP/GRE/SCTP)
  - application helpers (currently FTP,IRC,H.323,talk,SNMP,RTSP)
  - does _NOT_ filter packets itself
  - can be utilized by iptables using the 'state' match
  - is used by NAT Subsystem

Network Address Translation

- Network Address Translation
  - Previous Linux Kernels only implemented one special case of NAT: Masquerading
  - Linux 2.4.x / 2.6.x can do any kind of NAT.
  - NAT subsystem implemented on top of netfilter, iptables and conntrack
  - Following targets available within 'nat' Table
    - SNAT changes the packet's source whille passing NF_IP_POST_ROUTING
    - DNAT changes the packet's destination while passing NF_IP_PRE_ROUTING
    - MASQUERADE is a special case of SNAT
    - REDIRECT is a special case of DNAT

Packet Mangling

- Purpose of mangle table
  - packet manipulation except address manipulation
- Targets specific to the ’mangle’ table:
  - DSCP - manipulate DSCP field
  - IPV4OPTSSTRIP - strip IPv4 options
  - MARK - change the nfmark field of the skb
  - TCPMSS - set TCP MSS option
  - TOS - manipulate the TOS bits
  - TTL - set / increase / decrease TTL field

Linux Bridging

- Bridging (brctl)
  - Includes support for Spanning Tree
  - Fully supports packet filtering and NAT (!) on a bridge
  - Can also filter and translate layer 2 MAC addresses
  - Can implement a ’brouter’ (bridge certain traffic, route other)

Linux Policy Routing

- Policy Routing (iproute2)
  - Allows routing decisions on arbitrary information
  - Provides up to 255 different routing tables within one system
  - By combining via nfmark with iptables, any matches of the packet filter can be used for the routing decision
  - Very useful in complex setups with mutiple links (e.g. multiple DSL uplinks with dynamic addresses, asymmetric routing, ...)

Linux Traffic Shaping

- Traffic Control (tc)
  - Framework for lots of algorithms like RED,SFQ,TBF,CBQ,CSZ,GRED,HTB
  - Very granular control, especially for very low bandwidth links
  - Present since Linux 2.2.x but still not used widely
  - Lack of documentation, but situation is improving (www.lartc.org)

Free S/WAN

- Free S/WAN
  - Was a politically motivated effort to provide IPsec for Linux 2.0+
  - Goal was to encrypt as much Internet Traffic as possible
  - Software architecture didn't fit very well with Linux 2.4/2.6 network stack
  - Project has been shut down, however Open S/WAN continues support
  - Is in widespread production use and has received a lot of testing
  - Political motivation prevented any U.S. citizen to contribute code

Linux 2.6.x IPsec

- Linux 2.6.x IPsec
  - Linux networking gods disaproved Free S/WAN political restrictions and software design
  - Thus, they decided to write their own IPsec stack
  - Result is in the stock 2.6.x kernel series
  - Offers complete support for transport and tunnel mode
  - Can be used with FreeSWAN (pluto) or KAME (isakmpd) userspace
  - Remaining problems
    - No integration with hardware crypto accelerators yet
    - No implementation of NAT traversal yet
    - Interaction with iptable_nat still has to be sorted out

cipe, vtun, openswan and others

- Other VPN protocols/programs
  - Evolved as linux specific VPN implementations since the Linux Kernel was lacking stock IPsec support for a long time
  - Are totally incompatible to IPsec and only compatible to themselves
  - Are of questionable security (at least in case of cipe, vtun)
  - Are mostly userspace implementations
  - Are way easier to configure
  - Can provide layer 2 tunnels to route (or bridge!) all kinds of protocols
  - openvpn with X.509 certificates is a very clean and easy solution for building strong VPN tunnels between two linux gateways

Thanks

- Thanks to
  - the BBS scene, Z-Netz, FIDO, ...
    - for heavily increasing my computer usage in 1992
  - KNF (http://www.franken.de/)
    - for bringing me in touch with the internet as early as 1994
    - for providing a playground for technical people
    - for telling me about the existance of Linux!
  - Alan Cox, Alexey Kuznetsov, David Miller, Andi Kleen
    - for implementing (one of?) the world's best TCP/IP stacks
  - Paul 'Rusty' Russell
    - for starting the netfilter/iptables project
    - for trusting me to maintain it today
  - Astaro AG
    - for sponsoring parts of my netfilter work
- The slides and the an according paper of this presentation are available at http://www.gnumonks.org/