The nftables tutorial

Patrick McHardy
<kaber@trash.net>

Pablo Neira Ayuso
<pablo@netfilter.org>

Netdev 0.1
February 2015
Ottawa, Canada

What is nftables?

- New packet classification framework to replace {ip,ip6,arp,eb}tables based on lessons learnt.
- nftables was presented in Netfilter Workshop 2008 (Paris, France) and released in March 2009 by Patrick McHardy.
- Merged mainstream in October 2013, available since January 2014 in Linux kernel 3.13.
- It reuses the existing Netfilter building blocks: hooks, conntrack, NAT, logging and userspace queueing.
- We also reuse existing xtables extensions through nft compat.

Why nftables?

- Address iptables architectural design problems:
  - From kernelspace:
    - Avoid code duplication
      - Four families (arp, ip, ip6, bridge) derivated from the original iptables codebase.
      - Very similar extensions to match protocol fields and metadata.
    - Netlink API (including event notifications)
    - Better dynamic/incremental updates support
    - Linear ruleset evaluation: Generic set infrastructure allowing dictionaries.
  - From userspace:
    - New command line tool (with improved new syntax): nft
    - Proper userspace libraries for third party software

nftables source & documentation

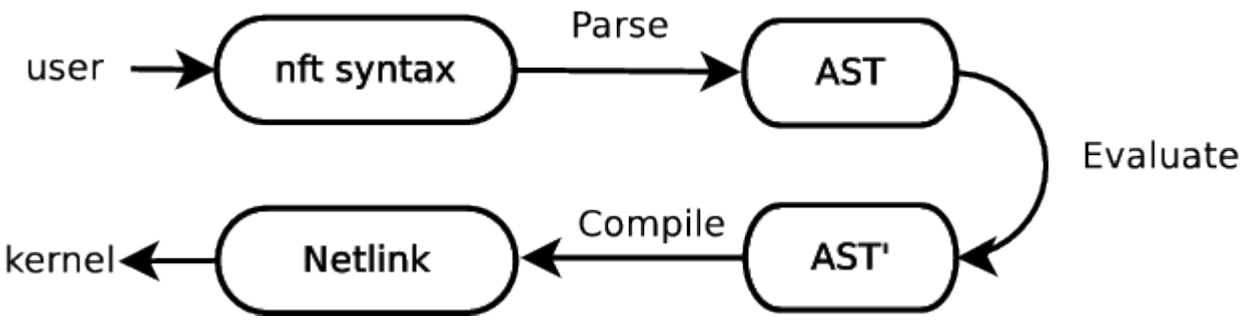
- Grab the code
  - Kernel:
    - http://www.kernel.org
    - http://git.kernel.org/cgit/linux/kernel/git/pablo/nf-next.git
  - Library: git://git.netfilter.org/libnftnl
  - User-space: git://git.netfilter.org/nftables
- Documentation
  - http://wiki.nftables.org (nftables HOWTO)
  - man nft
- If you find bugs:
  - https://bugzilla.netfilter.org
  - Send us an email to netfilter@vger.kernel.org

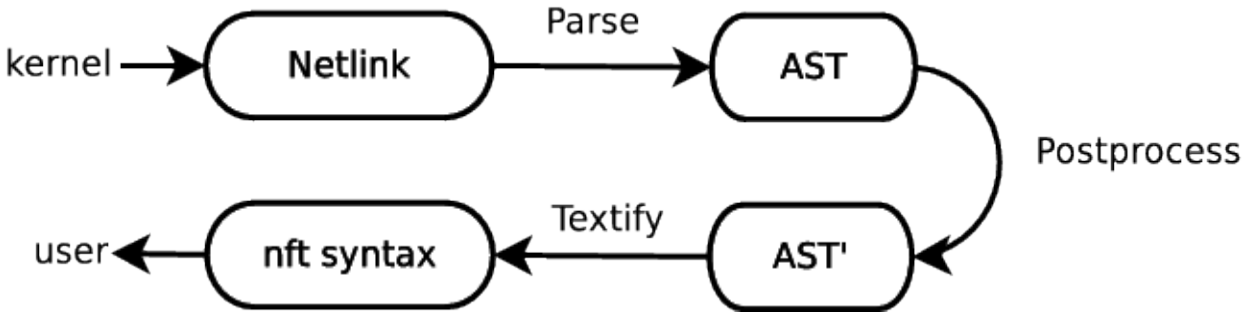
nftables: Tables and chains

- Tables are containers of chains (with no semantics)
  - families: ip, ip6, inet, bridge and arp
- Chains: list of rules
  - Base chains, registered as a hook in the stack
  - Non-base chains
- Live demo:
  - Adding, deleting and listing table
  - Adding, deleting and listing chains
    - Basechains
    - Non-base chains

nftables: Rules

- From userspace:





nftables: Rules

- Negation

```
# nft add rule ip filter input tcp dport != 80
```

- Ranges

```
# nft add rule ip filter input tcp dport 1-1024
# nft add rule ip filter input meta skuid 1000-1100
```

- Prefixes

```
# nft add rule ip filter input ip daddr 192.168.10.0/24
# nft add rule ip filter input meta mark 0xffffffff/24
```

- Flags

```
# nft add rule ip filter input ct new,established
```

- Bitwise

```
# nft add rule ip filter input ct mark and 0x0000ffff == 0x00001234
```

- Assignment

```
# nft add rule ip filter input ct mark set 10
# nft add rule ip filter input ct mark set meta mark
```

nftables: Rules

- Live demo:
  - Adding/Inserting rules
    - Expressions: payload, meta, ct
    - Ranges, prefix, bitwise, flags
  - Flushing table / chains
  - Deleting rules
  - Flushing rules
  - Listing and flushing the ruleset

nftables: Rules

- New features:
  - Optional rule counters:

```
# nft add rule ip filter input counter
# nft list table filter
...
```

  - Several actions in one single rule:

```
# nft add rule ip filter input \
    counter log prefix \"packet drop: \" drop
```

  - Interactive mode (still missing autocompletion):

```
# nft -i
nft>
```

  - Debugging mode:

```
# nft –debug=all ….
```

  - Live demo.

nftables: Sets

- Generic set infrastructure: You can create sets of any supported datatypes.
  - Anonymous sets:

```
# nft add rule ip filter input tcp dport { 22, 80, 443 } counter
```

  - Named sets:

```
# nft add set filter blackhole { type ipv4_addr \; }
# nft add element filter blackhole { 192.168.0.1, 192.168.0.10 }
# nft add rule ip filter input ip daddr @blackhole counter accept
```

  - Maps:

```
# nft add rule filter input snat ip saddr map { \
  1.1.1.0/24 : 192.168.3.11 , \
  2.2.2.0/24 : 192.168.3.12}
```

Nftables: Sets

- Existing set types:
  - rhashtable
  - rb-tree (for range matching)
- The kernel selects the best set for you:
  - Memory

    ```
    > add set filter set1 { type ipv4_addr ; policy memory ; }
    ```
  - Performance

    ```
    > add set filter set1 { type ipv4_addr ; policy performance ; }
    ```

nftables: Sets

- Dictionaries:

```
# nft -i
nft> add chain ip filter tcp-chain
nft> add chain ip filter udp-chain
nft> add chain ip filter icmp-chain
nft> add rule ip filter input ip protocol vmap { tcp : jump tcp-chain, \
                                                 udp : jump udp-chain, \
                                                 icmp: jump icmp-chain }
nft> add rule ip filter tcp-chain counter
nft> add rule ip filter udp-chain counter
nft> add rule ip filter icmp-chain counter
```

nftables: Sets

- Dictionaries:

```
nft> insert rule ip filter tcp-chain tcp dport vmap { 22 : accept, 80 : accept, 443 :
accept }
nft> add rule ip filter tcp-chain drop
nft> list table filter
table ip filter {
        chain input {
                type filter hook input priority 0;
                ip protocol vmap { icmp : jump icmp-chain, tcp : jump tcp-chain, udp :
jump udp-chain}
       }
       chain tcp-chain {
                tcp dport vmap { http : accept, ssh : accept, https : accept}
                counter packets 1 bytes 40
                drop
        }
       chain udp-chain {
               counter packets 29 bytes 3774
       }

       chain icmp-chain {
              counter packets 1 bytes 84
       }
}
```

nftables: ruleset

- Save ruleset

```
# echo “flush ruleset” > ruleset.file
# nft list ruleset >> ruleset.file
```

- Reload ruleset

```
# nft -f ruleset.file
```

- Flush ruleset

```
# nft flush ruleset
```

nftables: compat tools

- {ip,ip6,arp,eb}tables-compat

```
# iptables-compat -I INPUT -p tcp -j DROP
# iptables-compat-save > ruleset
# iptables-compat-restore < ruleset
```

- Still missing glue code to allow usage of xtables extensions from nft:

```
# nft add rule filter input ipt [ -j TCPMSS … ]
```

nftables status

- Currently under active development:
  - ~60% iptables supported extensions in native nft.
  - Still completing core features: Generic set infrastructure enhancements.
  - Bug hunting / fixing.
  - We'll release iptables 1.6.0 soon including {ip,ip6,arp,eb}tables-compat tools.
- Userspace:
  - nft:
    - version 0.4, released in December 16th, 2014.
  - libnftnl (requires libmnl):
    - Version 1.0.2, released in December 16th, 2014.

The nftables tutorial

Patrick McHardy
<kaber@trash.net>

Pablo Neira Ayuso
<pablo@netfilter.org>

Netdev 0.1
February 2015
Ottawa, Canada