NAME

xtables-nft — iptables using nftables kernel api

DESCRIPTION

xtables-nft are versions of iptables that use the nftables API. This is a set of tools to help the system administrator migrate the ruleset from **iptables(8)**, **ip6tables(8)**, **arptables(8)**, and **ebtables(8)** to **nftables(8)**.

The **xtables-nft** set is composed of several commands:

- iptables−nft
- iptables−nft−save
- iptables−nft−restore
- ip6tables−nft
- ip6tables−nft−save
- ip6tables−nft−restore
- arptables−nft
- ebtables−nft

These tools use the libxtables framework extensions and hook to the nf_tables kernel subsystem using the **nft_compat** module.

USAGE

The xtables-nft tools allow you to manage the nf_tables backend using the native syntax of **iptables(8)**, **ip6tables(8)**, **arptables(8)**, and **ebtables(8)**.

You should use the xtables-nft tools exactly the same way as you would use the corresponding original tools.

Adding a rule will result in that rule being added to the nf_tables kernel subsystem instead. Listing the ruleset will use the nf_tables backend as well.

When these tools were designed, the main idea was to replace each legacy binary with a symlink to the xtables-nft program, for example:

```
/sbin/iptables -> /usr/sbin/iptables−nft−multi
/sbin/ip6tables -> /usr/sbin/ip6tables−nft−multi
/sbin/arptables -> /usr/sbin/arptables−nft−multi
/sbin/ebtables -> /usr/sbin/ebtables−nft−multi
```

The iptables version string will indicate whether the legacy API (get/setsockopt) or the new nf_tables api is used:

```
iptables −V
iptables v1.7 (nf_tables)
```

DIFFERENCES TO LEGACY IPTABLES

Because the xtables-nft tools use the nf_tables kernel API, rule additions and deletions are always atomic. Unlike iptables-legacy, iptables-nft −A .. will NOT need to retrieve the current ruleset from the kernel, change it, and re-load the altered ruleset. Instead, iptables-nft will tell the kernel to add one rule. For this reason, the iptables-legacy −−wait option is a no-op in iptables-nft.

Use of the xtables-nft tools allow monitoring ruleset changes using the **xtables−monitor(8)** command.

When using −j TRACE to debug packet traversal to the ruleset, note that you will need to use **xtables−monitor(8)** in −−trace mode to obtain monitoring trace events.

EXAMPLES

One basic example is creating the skeleton ruleset in nf_tables from the xtables-nft tools, in a fresh machine:

```
root@machine:~# iptables−nft −L
[...]
root@machine:~# ip6tables−nft −L
[...]
root@machine:~# arptables−nft −L
[...]
root@machine:~# ebtables−nft −L
[...]
root@machine:~# nft list ruleset
table ip filter {
        chain INPUT {
                type filter hook input priority 0; policy accept;
        }

        chain FORWARD {
                type filter hook forward priority 0; policy accept;
        }

        chain OUTPUT {
                type filter hook output priority 0; policy accept;
        }
}
table ip6 filter {
        chain INPUT {
                type filter hook input priority 0; policy accept;
        }

        chain FORWARD {
                type filter hook forward priority 0; policy accept;
        }

        chain OUTPUT {
                type filter hook output priority 0; policy accept;
        }
}
table bridge filter {
        chain INPUT {
                type filter hook input priority −200; policy accept;
        }

        chain FORWARD {
                type filter hook forward priority −200; policy accept;
        }

        chain OUTPUT {
                type filter hook output priority −200; policy accept;
        }
```

```
}
table arp filter {
        chain INPUT {
                type filter hook input priority 0; policy accept;
        }

        chain FORWARD {
                type filter hook forward priority 0; policy accept;
        }

        chain OUTPUT {
                type filter hook output priority 0; policy accept;
        }
}
```

(please note that in fresh machines, listing the ruleset for the first time results in all tables an chain being created).

To migrate your complete filter ruleset, in the case of **iptables(8)**, you would use:

```
root@machine:~# iptables-legacy-save > myruleset # reads from x_tables
root@machine:~# iptables-nft-restore myruleset   # writes to nf_tables
```

or

```
root@machine:~# iptables-legacy-save | iptables-translate-restore | less
```

to see how rules would look like in the nft **nft(8)** syntax.

LIMITATIONS

You should use **Linux kernel >= 4.17**.

The CLUSTERIP target is not supported.

To get up-to-date information about this, please head to **http://wiki.nftables.org/**.

SEE ALSO

nft(8), **xtables-translate(8)**, **xtables-monitor(8)**

AUTHORS

The nftables framework is written by the Netfilter project (https://www.netfilter.org).

This manual page was written by Arturo Borrero Gonzalez <arturo@debian.org> for the Debian project, but may be used by others.