.Deb

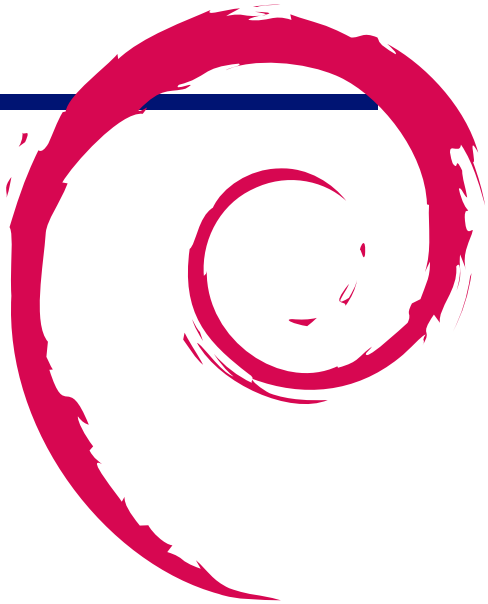
銀河系唯一のDebian専門誌

2019 年 12 月 22 日

nftables入門

デビアン勉強会

目次



1 最近の Debian 関連のミーティング報告

杉本典充

1.1 2019 年 10 月度 東京エリア Debian 勉強会

2019 年 10 月 19 日 (土) に東京エリア Debian 勉強会を開催しました。会場は荒川区立町屋文化センターの会議室を借りて行いました。参加者は 5 名でした。

セミナーは杉本さんの「Debian GNU/kFreeBSD セットアップガイド 2019 年版」を行いました。Debian GNU/kFreeBSD が Debian Ports に移行となりインストール方法や使えるパッケージが少ないながらどこまで動くのか試してみた内容でした。

「月間 Debian Policy」の時間は、最近更新があった Debian Policy はどのような課題があって変更に至ったのか、勉強会の参加している人で意見を出し合い共有しました。

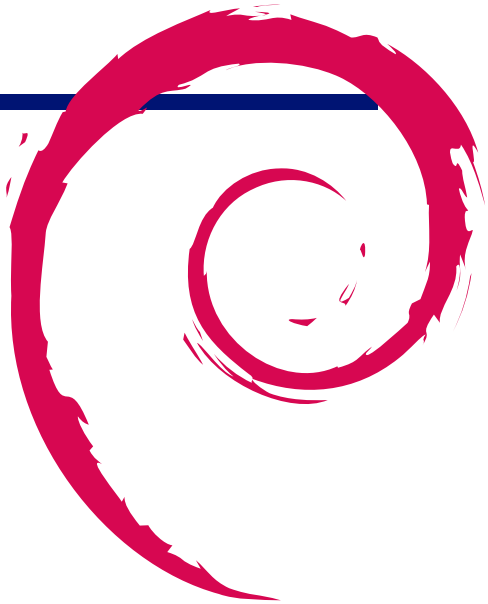
1.2 OSC 2019 Tokyo/Fall

2019 年 11 月 23 日 (土) に Debian JP Project ／東京エリア Debian 勉強会は、OSC 2019 Tokyo/Fall[*1]に参加しました。OSC の会場は明星大学様でした。

OSC のイベント全体では、11/23(土) に約 500 名、11/24(日) に約 330 名の方が来場しました。

セミナーでは「Debian Updates」という表題で Debian 10 buster のリリース情報の説明を杉本が発表し、参加者は 19 名でした。またブース展示を行い、約 43 名のイベント参加者と交流しました。

[*1] https://www.ospn.jp/osc2019-fall/



2 事前課題

杉本典充

今回の事前課題は以下です。

1. nftables を使ったことはありますか。OS は問いません。
2. Debian の話で聞きたいことがあれば教えてください。

2.1 koedoyoshida

1. 使ったことはありません
2. (回答なし)

2.2 NOKUBI Takatsugu (knok)

1. 使ったことはありません
2. Cosmo Communicator で Debian が動くそうですが誰か買いますか?

2.3 Kouhei Maeda (mkouhei)

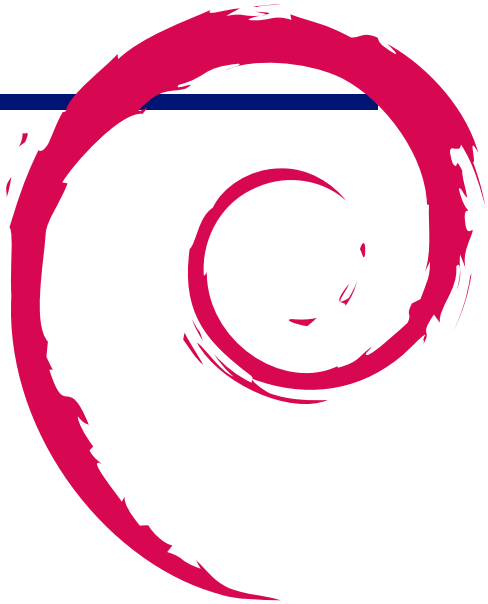
1. 使ったことはありません
2. (回答なし)

2.4 yy_y_ja_jp

1. 使ったことはありません
2. (回答なし)

2.5 dictoss

1. 使ったことがあります
2. ボード PC 系の情報（Raspberry Pi、PINE64)、コンテナのゲスト OS として debian はどの程度向いているのかの調査・見解



3 Debian 10 buster で nftables を使ってみる

杉本典充

3.1 はじめに

2019 年 7 月 6 日にリリースした安定版 Debian 10 buster ではネットワークフィルタリング機能に nftables がデフォルトで使われるように変わりました。

Debian 10 buster を使ってサーバを公開する場合には nftables を使うことがあると思いますので、調べてみました。

3.2 nftable とは

3.2.1 nftable の歴史

nftables[*2]とは、linux-3.13 （2014/1/19 リリース）から追加されたネットワークフィルタリング機能で iptables を置き換えることを目的として開発を進めている機能です。

iptables は linux-2.4 で新たに実装された netfilter[*3]という機能を使ってネットワークフィルタリング処理を行う仕組みで、歴史が長いです[*4]。

iptables は非常に長く多くの人たちに使われており、以下の課題があることがわかってきました[*5]。

- ソースコードに重複したコードが存在している
- 性能の頭打ち
- IPv4 と IPv6 のデュアルスタック時の管理の煩雑さ（nftables の inet ファミリーの追加につながる）
- ルールセットの更新処理がアトミックでない
- サードパーティアプリケーションが使える API がない（nftables では Netlink API がある）
- 構文の改善

上記を改善するため nftables の開発と linux kernel の netfilter 機能の改善が行われ、2014 年 1 月 20 日に nftables-0.099 がリリースされました[*6]。現在の最新版は 2019 年 12 月 2 日にリリースした nftables-0.9.3 が最新版です。

[*2] https://wiki.nftables.org/wiki-nftables/index.php/Main_Page
[*3] https://www.netfilter.org/
[*4] iptables-1.0.0.tar.bz2 に含まれる man ファイル”iptables.8”には 2000 年 3 月 20 日という日付が入っています。
[*5] https://wiki.nftables.org/wiki-nftables/index.php/Why_nftables%3F
[*6] https://www.netfilter.org/projects/nftables/downloads.html

3.2.2 debian と nftables

Debian においては、Debian wiki に nftables の情報があります[*7]。

Debian 9 stretch から nftables パッケージを提供しており、従来通り iptables をデフォルトとしつつ nftables も利用できる状況になっていました。このとき iptables が使う kernel module は以下になっています。

```
$ cat /etc/debian_version
9.11
$ lsmod | grep table
  ebtable_filter         16384  0
  ebtables               36864  1 ebtable_filter
  ip6table_filter        16384  0
  ip6_tables             28672  1 ip6table_filter
  iptable_filter         16384  1
  iptable_nat            16384  1
  nf_nat_ipv4            16384  1 iptable_nat
  iptable_mangle         16384  1
  ip_tables              24576  3 iptable_mangle,iptable_filter,iptable_nat
  x_tables               36864  12 ipt_REJECT,iptable_mangle,ip_tables,ebtables,iptable_filter,xt_tcpudp,
ipt_MASQUERADE,xt_CHECKSUM,ip6table_filter,xt_policy,xt_conntrack,ip6_tables
```

Debian 10 buster では iptables から nftables へデフォルトを切り替えており[*8]、nftables-0.9.0 を提供しています。nftables が使う kernel module は以下になっています。

```
$ cat /etc/debian_version
10.2
$ lsmod | grep table
nf_tables_set          32768  6
nf_tables             143360  60 nft_ct,nf_tables_set
nfnetlink              16384  1 nf_tables
```

また、iptables パッケージの提供もしており昔ながらの使い方もできるようになっています。iptables パッケージの中には "/usr/sbin/iptables-nft" と "/usr/sbin/iptables-legacy" の二種類を提供しており、*-nft の方は nf_tables の kernel module を使い、*-legacy の方は従来通り x_tables の kernel module を使っています。そのため、Debian 10 buster では、1. nf_tables の nft インタフェース、2. nf_tables の iptables 互換インタフェース、3. x_tables の旧来の iptables インタフェース の 3 つが動作するようになっています。

Debian 11 bullseye で nftables と iptables をどうしていくかの意見がメーリングリストやブログに投稿されています。意見としては iptables パッケージを optional 扱いにして nftables パッケージのみインストールされるようにする案に賛同する人が見られます（他の意見は firewalld をデフォルトにしてはどうかという案）。

- default firewall utility changes for Debian 11 bullseye[*9]
- What to expect in Debian 11 Bullseye for nftables/iptables[*10]

3.3 Debian 10 buster で nftables を使ってみる

3.3.1 インストール方法

Debian 10 buster において提供している nftables パッケージをインストールすると nft コマンドを利用できるようになり、nftables が systemd に登録されます。

[*7] https://wiki.debian.org/nftables
[*8] https://www.debian.org/releases/stable/amd64/release-notes/ch-whats-new.ja.html
[*9] https://lists.debian.org/debian-devel/2019/07/msg00332.html
[*10] https://ral-arturo.org/2019/10/14/debian-netfilter.html

```
# apt-get install nftables

# which nft
/usr/sbin/nft

# systemctl status nftables
nftables.service - nftables
  Loaded: loaded (/lib/systemd/system/nftables.service; disabled; vendor preset: enabled)
  Active: inactive (dead)
  Docs: man:nft(8)
  http://wiki.nftables.org
```

nftables のパケットフィルタリングルールの設定ファイルは、/etc/nftables.conf に配置されます。初期状態では以下の設定が入っています。

```
$ cat /etc/nftables.conf
#!/usr/sbin/nft -f

flush ruleset

table inet filter {
  chain input {
    type filter hook input priority 0;
  }
  chain forward {
    type filter hook forward priority 0;
  }
  chain output {
    type filter hook output priority 0;
  }
}
```

3.3.2 nft コマンドの使い方

nftables によるネットワークフィルタリングの機能は nft コマンドで制御します。

”nft list ruleset” コマンドを実行すると現在のルールを表示します。

```
# nft list ruleset
table inet 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;
    }
}
```

nftables へのルール定義の追加、変更、削除は以下の流れで行います。

- table の作成
- chain の作成
- rule の作成

table の作成

まずは最初に table を作成します。table を作成するときの引数にはアドレスファミリーを指定します。アドレスファミリーは以下表を値を指定できます[*11]。

コマンドの例です。

```
# nft add table ip mytable
```

chain の作成

[*11] man nft より抜粋

表 1　nft で利用できるアドレスファミリー

アドレスファミリー	説明
ip	IPv4 address family.
ip6	IPv6 address family.
inet	Internet (IPv4/IPv6) address family.
arp	ARP address family, handling IPv4 ARP packets.
bridge	Bridge address family, handling packets which traverse a bridge device.
netdev	Netdev address family, handling packets from ingress.

次に chain を作成します。chain を作成するときの引数には フックタイプ を指定します[*12]。前述では ip (=IPv4) を指定していますので、以下表の IPv4/IPv6/Inet のフックタイプを指定できます[*13]。

表 2　nft の IPv4/IPv6/Inet で利用できるフックタイプ

フックタイプ	説明
prerouting	ルーティング処理の前に実行するフック
input	パケットの入力時に実行するフック
forward	パケットの転送時に実行するフック
output	パケットの出力時に実行するフック
postrouting	ルーティング処理の後に呼び出されるフック

コマンドの例です。

```
# nft add chain ip mytable mychain_in { type filter hook input priority 0 \; }
```

rule の作成

次に rule を作成をします。前述の例では IPv4 の入力パケットをフィルターする chain を作成しましたのでこれに準ずる rule を作成します。以下は、外部から入力パケット は ssh のみを許可し、他は拒否する設定にしてみた例です。

```
# nft add rule ip mytable mychain_in tcp dport ssh accept
# nft chain ip mytable mychain_in { policy drop \; }
```

上述の実行する nft コマンドを見ると「tcp」「dport」「ssh」「accept」と記述しています。「tcp」の部分は udp とも書け、「dport」の部分は sport とも書け、「ssh」の部分は プロトコル名またはポート番号でもよく、「accept」の部分は drop とも書けます。これらを記述を組み合わせて自分の実現したいルールを作ってください。

細かい rule を設定するコマンドの構文は「man nft」を参照してください。

設定の保存

ここまで設定すると以下のようなルールで動作しています[*14]。

[*12] linux kernel の Netfilter 機構ではネットワークの入出力するパケットに対して任意な処理をひっかける機能がありこれを「フック」と呼んでいます。nftables や iptables で行う設定はこのフック処理を設定するフロントエンドツールと言えると思います。

[*13] man nft より抜粋。

[*14] 例示のためこのようなルールになっていますが、IPv6 の ssh 通信が外からくると通信できてしまうため本番運用するにはおそらく設定として不完全ですので注意してください。

```
# nft list ruleset
table inet 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 ip mytable {
    chain mychain_in {
        type filter hook input priority 0; policy drop;
        tcp dport ssh accept
    }
}
```

nftables が参照する設定ファイルは /etc/nftables.conf であり、”nft list ruleset” コマンドの出力をそのまま保存すればよいようになっています。

```
# cp -p /etc/nftables.conf /etc/nftables.conf.backup1
# nft list ruleset > /etc/nftables.conf
```

nftables を再起動して同じ設定が読み込まれるか確認します。

```
# systemctl restart nftables
# nft list ruleset
```

設定に問題がなければサーバの起動時にルールを自動反映するように設定します。

```
# systemctl enable nftables
```

3.4 利用シーンにおける設定例

3.4.1 インターネットに公開する web サーバ向け

インターネットに公開する web サーバでは、基本的に http、https、ssh のポートを公開すればよいと思います。設定は以下の条件で設定してみました。

- 入力パケットのデフォルト処理は破棄とし、許可したルールの入力パケットのみを受信
- ループバックインタフェースの通信はすべて許可
- 自分のホストから通信を開始した場合の相手のサーバからの戻りパケットは受信許可
- TCP のコネクションの状態としておかしいパケットは破棄
- ping のエコー要求を許可
- ssh を LAN 内から通信を許可
- ssh をインターネットからポート番号を tcp の 10022 ポートに変更した状態で通信を許可
- http/https はインターネットからも LAN 内からも通信を許可
- IPv6 では ssh、http/https のサービスはせず、ping6 の応答もしない

nftables の設定は以下になります。

```
# nft list ruleset
table ip filter {
    chain input {
        type filter hook input priority 0; policy drop;
        ct state { established, related } accept
        ct state { invalid } drop
        iifname "lo" accept
        icmp type { echo-reply, echo-request } accept
        tcp dport ssh ip saddr { 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16 } accept
        tcp dport 10022 ip saddr { 0.0.0.0-255.255.255.255 } accept
        tcp dport { http, https } ip saddr { 0.0.0.0-255.255.255.255 } 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 drop;
        iifname "lo" accept
    }

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

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

3.4.2 インターネットに公開する openvpn サーバ向け

openvpn[*15] とは SSL-VPN の機能をもった VPN サーバ及び VPN クライアントアプリケーションです。openvpn を使って VPN 接続したクライアントは openvpn サーバを中継して別のネットワークのサーバと通信することができます。

ここでは openvpn に接続したサーバ及びクライアントに割り当てられるネットワークアドレス帯”192.168.200.0/24”からパケットが入力された場合に、他のネットワークへパケットをルーティングする設定を行います。

```
# nft add table ip nat
# nft add chain ip nat postrouting { type nat hook postrouting priority 100 \; }
# nft add rule ip nat postrouting oifname "eth0" ip saddr 192.168.200.0/24 masquerade
```

openvpn を起動すると追加される”tun”インタフェースについても受信許可し、openvpn が利用する UDP と TCP の 119 番ポートの通信を受信許可に設定します。

nftables の設定は以下になります。

[*15] https://openvpn.net/community/

```
# nft list ruleset
table ip filter {
    chain input {
        type filter hook input priority 0; policy drop;
        ct state { established, related } accept
        ct state { invalid } drop
        iifname "lo" accept
        iifname "tun*" accept
        icmp type { echo-reply, echo-request } accept
        tcp dport ssh ip saddr { 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16 } accept
        tcp dport 10022 ip saddr { 0.0.0.0-255.255.255.255 } accept
        udp dport openvpn ip saddr { 0.0.0.0-255.255.255.255 } accept
        tcp dport openvpn ip saddr { 0.0.0.0-255.255.255.255 } 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 drop;
        iifname "lo" accept
    }

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

    chain output {
        type filter hook output priority 0; policy accept;
    }
}
table ip nat {
    chain postrouting {
        type nat hook postrouting priority 100; policy accept;
        oifname "eth0" ip saddr 192.168.200.0/24 masquerade
    }
}
```

3.5 おわりに

Debian 10 でデフォルトになった nftables を調べてみました。

今後 nftables が主流になっていくと思われるため Debian 10 buster を使い始める方は nftables も合わせて使い始めてみてはいかがでしょうか。

3.6 参考文献

- 「nftables - Debian Wiki」 https://wiki.debian.org/nftables
- 「netfilte.org」 https://www.netfilter.org/
- 「netfilter wiki」 https://wiki.nftables.org/wiki-nftables/index.php/Main_Page
- 「Linux における新たなパケットフィルタリングツール「nftables」入門 - さくらのナレッジ」 https://knowledge.sakura.ad.jp/22636/
- 「nftables で基本的なフィルタリングを設定してみた - SIOS TECH.LAB」 https://tech-lab.sios.jp/archives/16930

Debian 勉強会資料

2019 年 12 月 22 日　　初版第 1 刷発行

東京エリア Debian 勉強会 （編集・印刷・発行）