**Základy práce s utilitou nftables**

**Sady pravidel (ruleset):**

Obsahují všechny tabulky a řetězce aktuálně načtené v jádru systému.

**Příkazy:**

* iist – vypíše aktuální sady pravidel
* flush – vymaže všechno nastavení (tabulky, řetězce)

**Typy adres IP protokolu:**

* ip – adresy IPv4
* ip6 – adresy IPv6
* inet – adresy IPv4 i IPv6

**Tabulky (tables):**

Tabulky obsahují řetězce, ty obsahují hooky a pravidla. Tabulku určuje typ adres a její název.

**Příkazy:**

* add – přidá novou tabulku (*nft add table typ\_adres název\_tabulky*)
* delete – vymaže existující tabulku
* list – vypíše řetězce a pravidla z tabulky
* flush – vymaže všechny řetězce a pravidla z tabulky

**Řetězce:**

Řetězce obsahují hooky a pravidla.

**Příkazy:**

* add – přidá nový řetězec do určené tabulky, při nastavení priority a hooku bude zpracovávat procházející packety
* create – stejné jako příkaz add, ale vrací chybu, pokud už daný řetězec existuje
* delete – odstraní vybraný řetězec
* rename – přejmenuje konkrétní řetězec
* list – vypíše všechna pravidla v daném řetězci
* flush – odstraní všechna pravidla konkrétního řetězce

**Typy řetězců:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Typ** | **Typ adres** | **Hooky** | **Popis** |
| filter | Vše | Vše | Standardní výchozí typ |
| nat | Ip, ip6, inet | prerouting, input, output, postrouting | Provádí operaci NAT |
| route | Ip, ip6 | output | Provede nové vyhledání trasy packetu, pokud došlo ke změně části IP záhlaví, jestliže má být packet přijat. |

**Hooky:**

Hooky slouží k zachycení požadované síťové komunikace a jejímu předání ke zpracování pomocí zadaných pravidel v příslušném řetězci.

**Hooky pro IPv4/IPv6/Inet**

|  |  |
| --- | --- |
| **Název hooku** | **Komunikace** |
| prerouting | všechny packety vstupující do systému, zpracování před procesem směrování packetu |
| input | packety, které končí na lokálním systému |
| forward | packety, které systémem pouze procházejí, jejich cílem je jiný systém |
| output | packety, které vzniknou na lokálním systému a odtud odcházejí pryč |
| postrouting | packety opouštějící systém jsou zpracovány tímto hookem |
| ingress | packety vstupující do systému (předtím, než vstoupí ke zpracování L3 protokoly, akce se odehrává před vstupem do hooku prerouting) |

**Priorita zpracování:**

Určuje pořadí zpracování při procházení řetězců se stejným hookem, může být určena jako integer nebo názvem.

**Výběr standardních priorit:**

|  |  |  |  |
| --- | --- | --- | --- |
| Název priority | Hodnota | Typy adres | Hooky |
| dstnat | -100 | Ip, ip6, inet | prerouting |
| filter | 0 | Ip, ip6, inet, arp, netdev | vše |
| srcnat | 100 | Ip, ip6, inet | postrouting |

Ve skutečnosti je priorit definovaných více.

**Výchozí politika:**

Výchozí akce, která se má provést s packetem, pokud nevyhoví žádnému pravidlu v řetězci:

* accept – packet je přijat (propuštěn k dalšímu zpracování)
* drop – packet je zablokován

**Syntaxe vytvoření řetězce s hookem a výchozí politikou:**

*nft add chain typ\_adres název\_tabulky název\_řetězce { type typ\_řetězce hook název\_hooku; pripority priorita; policy výchozí\_politika }*

**Pravidla:**

Jednotlivá pravidla se přidávají do řetězců a zpracovávají komunikaci. Pokud u pravidel není specifikovaný typ adres, použije se standardně typ ip. Pořadí zpracování pravidel lze nastavit pomocí hodnot handle nebo index.

**Příkazy:**

* add – přidá nové pravidlo na konec řetězce, pokud není specifikováno pořadí
* insert – vloží pravidlo na určitou pozici mezi existující pravidla
* replace – podobné jako add, ale nahradí existující pravidlo
* delete – odstraní zadané pravidlo

**Direktivy pravidel:**

* iif – index vstupního rozhraní
* oif – index výstupního rozhraní
* iifname – jméno vstupního rozhraní
* oifname – jméno výstupního rozhraní
* saddr – zdrojová adresa (původce komunikace)
* daddr – cílová adresa (příjemce komunikace)
* sport – zdrojový port
* dport – cílový port
* ct state – stav komunikace (new | related,established | invalid)
* counter – počítadlo packetů
* protokoly: tcp, udp, icmp, all + další dle definovaných názvů nebo portů

**Akce (statement):**

Akce určuje, co má systém provést s danými packety, které vyhověly konkrétnímu pravidlu.

**Výběr z typů akcí:**

* accept – packety jsou přijaty (propuštěny dále)
* drop – packety jsou zahozeny (komunikace je zablokována)
* continue – pokračuje se zpracováním dalšího pravidla (výchozí akce, pokud nedošlo ke shodě s aktuálním pravidlem)
* jump – skok do jiného řetězce, zpracování pokračuje prvním pravidlem v novém řetězci dokud nedojde k finálnímu rozhodnutí nebo akci return (projetí všech pravidel), uchovává původní pozici, při akci return pokračuje na původní pozici původního řetězce
* return – návrat zpět do původního řetězce po zpracování pravidel nového řetězce při akci jump
* goto – stejné jako jump, neuchovává pozici
* log – zapíše událost (shoda s pravidlem) do systémového logu
* reject – podobné jako drop, vrací zpět chybovou zprávu
* snat to adresa – odchozí NAT, v packetu opouštějícím LAN nahradí interní zdrojovou IP adresu za veřejnou IP adresu
* dnat to adresa[:port] – příchozí NAT, určuje, na jakou vnitřní adresu v LAN a port má být packet směrován
* masquerade – totéž jako snat, používá se, pokud je adresa odchozího rozhraní nastavené z DHPC
* redirect to port – přesměrování vnitřní komunikace na jiný port

S**yntaxe pravidla:**

*nft add rule typ\_adres název\_tabulky název\_řetězce udp dport 53 couter accept*

**Sety (sady nastavení):**

Sety umožňují zjednodušit konfiguraci řetězců tím, že jedno pravidlo může obsahovat více parametrů vyhodnocované komunikace (více adres, portů atd.)

**Typy setů:**

* pojmenované – je možné je měnit za běhu, musí být definované předem
* anonymní – není možné je měnit za běhu, definují se přímo v pravidlech   
  např. *ip saddr { 10.0.0.1, 10.0.0.2 } tcp dport { 80, 443 }*

**Maps (mapy):**

Fungují podobně jako sety, ale slouží k vytváření slovníků (dvojice odpovídajících parametrů).

**Typy map:**

* pojmenované - je možné je měnit za běhu, musí být definované předem
* anonymní - není možné je měnit za běhu, definují se přímo v pravidlech   
  např. *ip nat prerouting dnat to tcp dport map { 25 : 10.0.0.1, 22: 10.0.0.2 } tcp dport { 80, 443 }*

Přednastavené rulesety se dají napsat ve formě skriptu, zápis je jednodušší, není potřeba stále opakovat příslušné příkazy. Celý set se pak dá načíst do jádra pomocí příkazu nft -f název\_souboru.