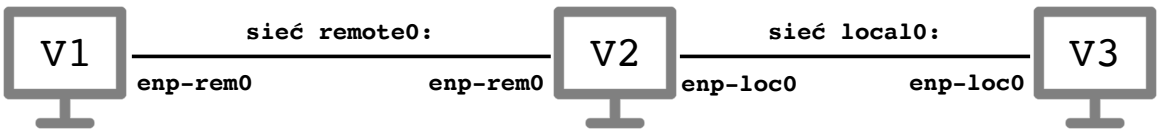
Warsztaty z Sieci komputerowych
Lista 8

Zadanie do zaprezentowania (2 pkt.)

Celem jest osiągnięcie topologii jak na rysunku poniżej. W tym i kolejnych zadaniach warto myśleć, że maszyna *Virbian3* jest lokalnym komputerem, maszyna *Virbian2* jest routerem, zaś maszyna *Virbian1* jest serwerem w Internecie.



- Uruchom trzy maszyny wirtualne *Virbian1* – *Virbian3* połączone za pomocą sieci wirtualnej `local0` i `remote0`. Odpowiednia para interfejsów powinna mieć nazwy `enp-loc0` zaś druga para interfejsów nazwy `enp-rem0` jak na rysunku powyżej. Aktywuj wszystkie interfejsy i uruchom Wiresharka na wszystkich maszynach wirtualnych.
- Wybierz dwie rozłączne adresowo sieci IP i przypisz interfejsom `enp-rem0` adresy IP z jednej z tych sieci, a interfejsom `enp-loc0` adresy IP z drugiej z tych sieci. Sprawdź, że z maszyny *Virbian2* możesz z powodzeniem pingnąć obie sąsiednie maszyny.
- Na maszynie *Virbian3* ustaw bramę domyślną na adres interfejsu `enp-loc0` maszyny *Virbian2*. Sprawdź, że z maszyny *Virbian3* możesz z powodzeniem pingnąć oba adresy IP maszyny *Virbian2*.
- Sprawdź, co dzieje się, jeśli z maszyny *Virbian3* pingasz maszynę *Virbian1*. Za pomocą Wiresharka sprawdź, że komunikaty *ICMP echo request* dochodzą do celu, ale odpowiedzi nie wracają do nadawcy. Dlaczego tak się dzieje?

Tutorial 1 (0 pkt.)

W tej części skonfigurujemy zaporę na maszynie *Virbian1*, wykorzystując moduł `nftables` jądra konfigurowany przez polecenie `nft`. Zaporę można konfigurować interaktywnie za pomocą tego programu, lecz wygodniej jest edytować plik konfiguracyjny `/etc/nftables.conf`.

- Upewnij się, że zawartość pliku `/etc/nftables.conf` na maszynie *Virbian1* jest (z dokładnością do białych znaków) taka, jak poniżej.

```
#!/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;
  }
}
```

▶ Wykonaj teraz polecenie

```
V1#> /etc/nftables.conf
```

Powyższe polecenie (na maszynie *Virbiani*) należy wykonywać *po każdej* edycji pliku `/etc/nftables.conf` na maszynie *Virbiani* (nie będzie to zaznaczone w poniższych zadaniach). Spowoduje to skonfigurowanie zapory zgodnie z instrukcjami z tego pliku. Obecnie instrukcje te usuwają wszystkie istniejące reguły a następnie implementują (pustą) konfigurację z sekcji `table inet filter {...}`.

▶ Zmodyfikuj plik `/etc/nftables.conf` na maszynie *Virbian1*, tak żeby pakiety przychodzące do maszyny *Virbian1* i przechodzące przez nią były wyrzucane, zaś pakiety wychodzące przepuszczane. Dodatkowo będziemy rejestrować wszystkie odrzucone w ten sposób pakiety do pliku dziennika. W tym celu sekcja `table inet filter {...}` powinna wyglądać następująco:

```
chain input {
  type filter hook input priority 0;
  log
  drop
}
chain forward {
  type filter hook forward priority 0;
  log
  drop
}
chain output {
  type filter hook forward priority 0;
  accept
}
```

▶ Wyświetl bieżące ustawienia zapory poleceniem

```
V1#> nft list ruleset
```

Wyświetlone reguły powinny odpowiadać zapisanym w pliku `/etc/nftables.conf`.

- W osobnym terminalu uruchom polecenie

```
V1#> tail -f /var/log/syslog
```

wyświetlające bieżącą zawartość pliku dziennika maszyny *Virbian1* (czyli w szczególności odrzucone pakiety).

- Sprawdź, co zapisuje się do pliku dziennika i co wyświetlane jest w Wiresharku, gdy pingasz z maszyny *Virbian2* maszynę *Virbian1*, a co, gdy z maszyny *Virbian1* pingasz maszynę *Virbian2*. W którym przypadku zatrzymywane są komunikaty *ICMP echo request* a w którym komunikaty *ICMP echo reply*?

- Zaktualizuj `/etc/nftables.conf` na maszynie *Virbian1* dopisując do sekcji `chain input` {...} regułę wpuszczającą pakiety należące do już nawiązanych połączeń. Odpowiednia część pliku konfiguracyjnego powinna teraz wyglądać następująco:

```
chain input {
  type filter hook input priority 0;
  ct state established accept
  log
  drop
}
```

Zwróć uwagę na kolejność reguł: zaakceptowane pakiety nie będą rejestrowane. Zaobserwuj, że teraz pinganie maszyny *Virbian2* z maszyny *Virbian1* będzie już działać.

- Sprawdź, że nadal nie jest możliwe pingnięcie maszyny *Virbian1* z maszyny *Virbian2*. Co więcej, nie jest nawet możliwe pingnięcie maszyny *Virbian1* z niej samej:

```
V1$> ping 127.0.0.1
```

Aby to naprawić wpuść wszystkie połączenia lokalne oraz połączenia ICMP *echo request* z zewnątrz, zmieniając sekcję `chain input` {...} pliku konfiguracyjnego na następującą:

```
chain input {
  type filter hook input priority 0;
  iif lo accept
  ct state established accept
  ip protocol icmp icmp type echo-request accept
  log
  drop
}
```

W razie potrzeby składnię poleceń możesz sprawdzić w manualu, albo przeczytać jeden z dostępnych tutoriali dla `nftables`, np. https://wiki.archlinux.org/index.php/Nftables.

Sprawdź, że osiągalność maszyny *Virbian1* za pomocą ICMP uległa poprawie.

- Za pomocą polecenia

  `V2#> nmap -A -T4` *`adres_IP_Virbiana1`*

  sprawdź, jakie porty są dostępne do komunikacji na maszynie *Virbian1*. Przeczytaj uważnie wyświetlane informacje. Oglądając pakiety w Wiresharku i komunikaty o zablokowanych pakietach w pliku dziennika sprawdź, z jakimi portami usiłował połączyć się `nmap`.

Zadanie do zaprezentowania (3 pkt.)

- Uruchom usługę SSH na maszynie *Virbian1*. i spróbuj połączyć się z nią z maszyny *Virbian2*. Jakie polecenie należy wydać? Czy udaje się nawiązać połączenie? Jaki komunikat pojawia się w pliku dziennika?
- Aby naprawić problem z łącznością dodaj do pliku `/etc/nftables.conf` na maszynie *Virbian1* następujący wiersz:

  ```
  tcp dport 22 ct state new accept
  ```

  W którym miejscu należy go dodać? Jak wygląda cały plik `/etc/nftables.conf` po tej zmianie? Ponownie wypróbuj połączenie z serwerem SSH z maszyny *Virbian2*; tym razem próba powinna zakończyć się powodzeniem.
- Poleceniem `netstat` sprawdź, na jakich portach nasłuchują usługi na maszynie *Virbian2*.
- Skonfiguruj zaporę na maszynie *Virbian2*, tak żeby:
  - pakiety wychodzące z tej maszyny były akceptowane;
  - pakiety przechodzące przez tą maszynę były rejestrowane a następnie odrzucane;
  - pakiety przychodzące skierowane do usługi *echo* (port TCP 7) były przyjmowane;
  - pakiety przychodzące typu *ICMP echo request* były przyjmowane;
  - pakiety przychodzące do interfejsu lokalnego `lo` i interfejsu `enp-loc0` były przyjmowane;
  - pakiety przychodzące będące odpowiedziami na już wysłane pakiety były przyjmowane;
  - pozostałe pakiety przychodzące były najpierw rejestrowane a następnie odrzucane.
- Z maszyn *Virbian1* i *Virbian3* poleceniem `ping` sprawdź dostępność maszyny *Virbian2* (jej najbliższego interfejsu).
- Poleceniem `nmap` sprawdź, jakie porty są dostępne do komunikacji na maszynie *Virbian2*. Polecenie to uruchom na maszynie *Virbian1* i *Virbian3*. Czy wyniki różnią się? Dlaczego?

Tutorial 2 (0 pkt.)

W tym zadaniu skonfigurujemy mechanizm NAT na maszynie *Virbian2*, żeby umożliwić maszynie *Virbian3* komunikację z maszyną *Virbian1* (i innymi potencjalnymi maszynami osiągalnymi z *Virbian2* za pomocą interfejsu `enp-rem0`).

- Z maszyny *Virbian3* pingnij maszynę *Virbian1*. Co pojawia się w pliku dziennika maszyny *Virbian2*? Okazuje, się że winne jest blokowanie ruchu przechodzącego przez tę maszynę. Napraw to zmieniając sekcję `chain forward {...}` pliku konfiguracyjnego `/etc/nftables.conf` na następującą:

  ```
  chain forward {
    type filter hook input priority 0;
    ct state established accept
    iif enp-loc0 oif enp-rem0 ct state new accept
    log
    drop
  }
  ```

  Zmiany powodują przepuszczanie całego ruchu pochodzącego od maszyny *Virbian3* do maszyny *Virbian1* i przepuszczanie pakietów należących do już nawiązanych połączeń w drugą stronę. Sprawdź, że jeśli pingasz maszynę *Virbian1* z maszyny *Virbian3*, to pakiety *ICMP echo request* już docierają, ale wciąż nie wracają odpowiedzi *ICMP echo reply*.

- Moglibyśmy naprawić sytuację definiując odpowiednio routing na maszynie *Virbian1*. Ale w tym tutorialu będziemy zakładać, że nie mamy takiej możliwości i poradzimy sobie włączając funkcję źródłowego NAT na maszynie *Virbian2*. W tym celu na końcu pliku `/etc/nftables.conf` dopisz następujące wiersze

  ```
  table ip nat {
    chain postrouting {
      type nat hook postrouting priority 0;
      ip saddr adres_sieci_local0 oif enp-rem0 snat IP_enp-rem0_Virbiana2
    }
  }
  ```

  gdzie *adres_sieci_local0* powinien być podany w notacji CIDR. Spowodują one, że pakiety z oryginalnym adresem źródłowym pochodzącym z sieci `local0` i wychodzące przez interfejs `enp-rem0` będą otrzymywały adres IP tego interfejsu.

- Pingnij maszynę *Virbian1* z maszyny *Virbian2* i z maszyny *Virbian3*. Porównaj w Wiresharku na maszynie *Virbian1* komunikaty ICMP dochodzące do maszyny *Virbian1* w obu powyższych przypadkach. Zwróć w szczególności uwagę na różnice w polu TTL. Jakie są adresy źródłowe i docelowe tych komunikatów?

  Obejrzyj też te komunikaty w Wiresharku uruchomionym na maszynie *Virbian2*: wszystkie pakiety przechodzące będą rejestrowane dwukrotnie, tj. przed podmianą źródłowego adresu IP i po niej.

- Zdekonfiguruj interfejsy sieciowe i wyłącz maszyny wirtualne.

Lista i materiały znajdują się pod adresem http://www.ii.uni.wroc.pl/~mbi/dyd/.

Marcin Bieńkowski