Routing

Niezależnie od tego, czy wiesz, jak to działa, za każdym razem, gdy podłączasz komputer do sieci, masz do czynienia z problemem routingu. Routing to sposób przesyłania pakietu IP z jednego punktu do drugiego. Na przykład, kiedy wysyłasz wiadomość e-mail do znajomego w innym kraju lub nawet na drugą stronę ulicy, przesyłasz serię pakietów IP lub datagramów z komputera do komputera znajomego.

Tablica routingu

Tablice routingu możesz sprawdzić korzystając z polecenia poniżej

```
netstat -r
```

Omówienie tablicy routingu:

- Kolumna Destination identyfikuje sieć docelową.
- Kolumna Gateway identyfikuje zdefiniowaną bramę dla określonej sieci.
- Gwiazdka (*) pojawia się w tej kolumnie, jeśli w sieci nie jest potrzebna brama przekierowania.
- Kolumna Genmask pokazuje maskę sieci dla sieci; w tym przypadku jest to 255.255.255.0.
- Kolumna Iface przedstawia interfejs sieciowy. Gdy masz więcej niż jeden interfejs, zobaczyłbyś lo (dla pętli zwrotnej), eth0 (pierwsze urządzenie Ethernet) i eth1 (dla drugiego urządzenia Ethernet) i tak dalej dla liczby zainstalowanych interfejsów.
- W sekcji Flagi flaga U oznacza, że trasa jest w górę, a symbol G oznacza, że dla tej trasy należy użyć określonej bramy.
- Istnieją inne flagi, które możesz zobaczyć, które obejmują:
  - D dla instalacji dynamicznej
  - M dla zmodyfikowanej
  - R dla przywrócenia
- Te trzy flagi wskazują, że trasa została utworzona lub zmodyfikowana przez demona routingu po napotkaniu komunikatu ICMP Redirect. (Zwykle nie zobaczysz tych flag, chyba że używasz routingu dynamicznego).
- Możesz również zobaczyć ! flaga, która wskazuje odrzucającą trasę.
- Kolumna MSS wskazuje domyślny maksymalny rozmiar segmentu dla połączeń TCP na tej trasie.
- Kolumna Window wskazuje domyślny rozmiar okna dla połączeń TCP na tej trasie
- Kolumna irtt wskazuje początkowy czas podróży w obie strony dla tej trasy. Jądro używa tego do wybierania wartości dla pewnych parametrów TCP bez konieczności czekania na potencjalnie wolne odpowiedzi ze zdalnych hostów.
- Sieć pętli zwrotnej jest obsługiwana lokalnie, więc nie jest wymagana żadna brama przekazująca.
- Ostatnia linia zawiera Domyślne miejsce docelowe, czasami wyświetlane jako 0.0.0.0, co oznacza wszystko inne, które nie zostało jeszcze sklasyfikowane.

Routing na przykładzie Żeby zobaczyć tablicę trasowania należy użyć polecenia

```
ip route show
```

Routing Linux

```
### Najpierw sprawdźmy tablice routingu
route
```

Warto przypomnieć o możliwości uzyskania dużo dokładniejszych informacji o narzędziach i ich użyciu w systemie Linux. W tym celu skorzystamy z polecenia:

```
man
# np.
man route
route -n
# alternatywna wersja bardziej surowa ( mniej przyjazna do czytania)
```

Do tego ćwiczenia będą potrzebne dwa hosty połączone siecią, jeden nasz, drugi np. na maszynie wirtualnej. Do prześledzenia trasy jaką pokonuje pakiet z hostu A do hostu B należy sprawdzić adres IP hosta B

```
ipconfig # dla Windows
ip addr # dla Linux

### a następnie należy wywołać śledzenia trasy do hosta B

traceroute IP_HOST_B # np. 192.168.1.23

### W celu dodania tablicy routingowej skorzystaj z polecenia poniżej

sudo ip route add 192.168.2.0/24 dev ens33


### Możemy również przekierować ruch przez bramę

sudo ip route add 192.168.2.9/24 via 192.168.2.254 dev ens33
# gdzie 192.168.2.29/24 to adres podsieci, a 192.168.2.254 to adres bramy

### Do usunięcia adresu podsieci z tablicy routingu służy polecenie

sudo ip route del 192.168.2.0/24 dev ens33
```

Port forwarding

Jedną z często przydatnych technik jest tunelowanie SSH. Załóżmy, że mamy dwa hosty w sieci. Jeden z nich udostępnia usługę Apache ze stroną internetową. Strona internetowa jest dostępna tylko z poziomu hosta, a my chcemy skorzystać z niej zdalnie. W tym celu wykorzystamy tunelowanie SSH. Po prostu przekierujemy porty z jednego hosta na drugi i uzyskamy dostęp jako "localhost". Warto rozróżnić dwie opcje tunelowanie -L -R. Ta pierwsza oznacza przekierowanie portów ze zdalnego hosta do lokalnego, a ta druga z lokalnego do zdalnego hosta. Jest jeszcze trzeci sposób tunelowania, który będzie można znaleźć na końcu.

Zacznijmy od tej pierwszej

```
ssh -L -f -N 8888:localhost:80 karmaz@192.168.0.5
### Opcja L oznacza przekierowanie portów ze zdalnego hosta do lokalnego, F przejście
w tryb tła, a N wyłączenia wyjścia na hoście zdalnym. Po wykonaniu tego polecenia
```

```
będziemy mogli skorzystać z usług hosta zdalnego na porcie 80 (dla przykładu serwer
Apache) korzystając z naszego adresu IP localhost na porcie 8888.

### W celu potwierdzenia poprawnego tunelowania, sprawdźmy aktywne połączenia sieciowe

netstat -antp

### Teraz w lewej kolumnie powinniśmy znaleźć nasz adres IP z portem 8888, a po prawej
adres IP hosta zdalnego z portem 80

### W celu potwierdzenia połączenia możemy włączyć nasłuchiwanie na zdalnym hoście i
połączyć się z nim na naszym hoście

### Zdalny host
nc -nlvp 80

### Nasz host
nc -nv localhost 8888

### Powinniśmy na komputerze naszego hosta otrzymać informację o udanym połączeniu
### Warto zwrócić uwagę, że łączymy się z naszym własnym hostem na porcie 8888 dzieje
się tak dlatego, że przekierowaliśmy cały ruch z hosta zdalnego na porcie 80 do
naszego hosta na porcie 8888

### Wykonajmy podobny przykład, ale uruchamiając na hoście zdalnym usługę Apache

### Zdalny host
sudo service apache2 start

### Jeżeli nie mamy usługi Apache warto ją zainstalować poleceniem poniżej
sudo apt-get install apache2

### Nasz host
### W celu sprawdzenia czy możemy skorzystać z usługi Apache zdalnego hosta, w pasku
przeglądarki wpiszmy http://localhost:8888/. Powinna wyświetlić się nam strona
startowa Apache

### Można też skorzystać z narzędzia curl, które połączy się z stroną z linii komend
curl http://localhost:8888

### W przypadku poprawnego tunelowania dostaniemy w odpowiedzi treść strony na zdalnym
hoście

### W celu zakończenia tunelowania szukamy odpowiedniego procesu
ps aux | grep 8888

# a następnie zabijamy proces
kill PID # gdzie PID to identyfikator procesu
```

Teraz druga metoda tunelowania

```
ssh -R -f -N 8888:192.168.0.5 karmaz@192.168.0.5

### Nasz host
### Uruchamiamy serwer Apache
sudo service apache2 start

### Zdalny host

### Teraz na zdalnym hoście wchodzimy pod adres http://localhost:8888

### Możemy też użyć do tego curla
curl http://localhost:8888
### Odpowiedź z treścią strony świadczy o poprawnym połączeniu


### W celu zakończenia tunelowania szukamy odpowiedniego procesu
ps aux | grep 8888

# a następnie zabijamy proces
kill PID # gdzie PID to identyfikator procesu
```

Tunelowanie dynamiczne

Czasami zachodzi konieczność użycia wielu aplikacji na różnych portach. Ustawianie połączenia po SSH do każdej z nich byłoby uciążliwe. Jest na to rada. Tunelowanie dynamiczne, które wykorzystuje protokół SOCKS, stąd też wymóg żeby aplikacja z którą się kontaktujemy też miała wsparcie dla protokołu SOCKS. Tutaj warto zaznaczyć, że dzięki tej metodzie uzyskamy dostęp do usług hosta zdalnego na hoście lokalnym.

```
ssh -f -N -D 8888 karmaz@192.168.0.5
```