

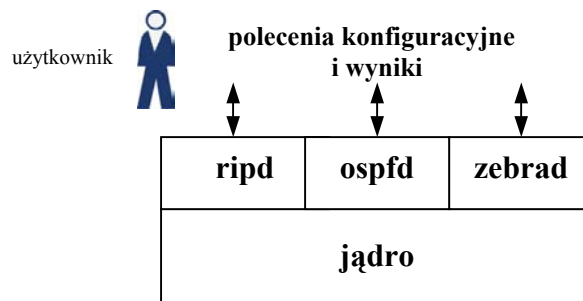
Wprowadzenie do pakietu Zebra

1. Wstęp

Pakiet Zebra jest narzędziem przeznaczonym do realizacji dynamicznego wyboru trasy IP w systemie Linux. Stanowi warstwę (nakładkę) pośredniczącą w komunikacji między użytkownikiem a jądrem systemu.

Na pakiet składa się kilka programów realizujących różne internetowe protokoły dynamicznego wyboru trasy, m.in. RIPv1, RIPv2, OSPFv2, OSPFv3 i BGP4; zapewniono również wsparcie dla protokołu IPv6. Nazwy programów (demonów) są bezpośrednio związane z nazwami protokołów (np. nazwa `ripd` od protokołu RIP, `ospfd` od OSPF itd.).

Tablica tras oraz informacje o interfejsach pamiętane są w jądrze systemu operacyjnego. Za komunikację z jądrem odpowiada specjalny proces-zarządca, **zebrad**. Przekształca on zlecenia użytkownika i komunikaty protokołów na odpowiednie wywołania systemowe, które odczytują lub zapamiętują nowe informacje w jądrze. Budowę pakietu ilustruje rysunek 1.



Rysunek 1. Budowa pakietu zebra

1.1 Język konfiguracyjny

Język konfiguracji stosowany w pakiecie Zebra wzorowany jest na języku systemu Cisco IOS. Dzięki temu praca w programie Zebra przypomina pracę z routerem Cisco.

1.2 Pliki konfiguracyjne i terminale

Konfiguracja routera dokonywana jest na dwa sposoby. Pierwszy polega na wczytaniu poleceń konfiguracyjnych z pliku. Każdemu protokołowi odpowiada inny plik (pliki konfiguracyjne umieszczone są w katalogu `/usr/local/etc/`). Korzystając z drugiego sposobu, można dodatkowo podczas pracy procesu-routera konfigurować go na bieżąco za pośrednictwem terminala. Terminal jest wątkiem procesu przeznaczonym do komunikacji z użytkownikiem; komunikacja ta odbywa się za pośrednictwem protokołu TCP. Poniższa tabela zawiera informacje o domyślnych numerach portów używanych przez terminale poszczególnych procesów.

Proces	Numer portu
zebrad	2601
ripd	2602
ripngd	2603
ospfd	2604
ospf6d	2606
bgpd	2605

Tabela 1. Numery portów terminali

Terminal wywoływany jest usługą telnet. Dla przykładu, wydanie polecenia `telnet localhost 2601` spowoduje uruchomienie terminala procesu-zarządcy zebra. Istnieje również możliwość uruchomienia procesu na wybranym przez siebie numerze portu; należy wówczas skorzystać z opcji `-P`.

1.3 Kompilacja i instalacja pakietu

Po skopiowaniu pliku z kodem źródłowym (rozszerzenie `.tar.gz`) należy najpierw dokonać jego dekompresji; można w tym celu posłużyć się poleceniem systemowym:

```
tar -xzf <nazwa pliku>.
```

Aby zainstalować pakiet, trzeba w katalogu głównym projektu wykonać kolejno trzy następujące polecenia:

```
./configure  
make  
make install
```

2. Przykłady

Poniższe przykłady ilustrują pracę z programem **zebrad**. Pierwszy pokazuje, jak wywołać terminal procesu. Hasło, którego program oczekuje od użytkownika, zapisane jest w pliku konfiguracyjnym `/usr/local/etc/zebra.conf`:

```
hostname Router  
password zebra
```

Przykład 1. Zawartość pliku `/usr/local/etc/zebra.conf`

```
# telnet localhost 2601  
Trying 127.0.0.1...  
Connected to localhost.  
Escape character is '^]'.  
  
Hello, this is zebra (version 0.94).  
Copyright 1996-2002 Kunihiro Ishiguro.  
  
User Access Verification  
  
Password:  
Router>
```

Przykład 1. Wywołanie terminala procesu zebra

Drugi przykład prezentuje dwa równoważne sposoby konfigurowania interfejsu rutera (poprzez terminal i w pliku konfiguracyjnym /usr/local/etc/zebra.conf) oraz diagnozowanie interfejsu. Jednej karcie sieciowej przypisano tu dwa adresy IP: 192.168.1.1 oraz 192.168.2.1.

```
Router> enable
Router# configure terminal
Router(config)# interface eth0
Router(config-if)# ip address 192.168.1.1/24
Router(config-if)# ip address 192.168.2.1/24
Router(config-if)# no shutdown
Router(config-if)# end
Router#
```

a)

```
interface eth0
ip address 192.168.1.1/24
ip address 192.168.2.1/24
no shutdown
```

b)

```
Router# show interface eth0
Interface eth0
  index 2 metric 1 mtu 1500 <UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST>
  HWaddr: 00:03:47:b2:22:f2
  inet 192.168.1.1/24 broadcast 192.168.1.255
  inet 192.168.2.1/24 broadcast 192.168.2.255
  inet6 fe80::203:47ff:feb2:22f2/64
    input packets 3514, bytes 1587445, dropped 0, multicast packets 0
    input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 0, missed 0
    output packets 1045, bytes 77235, dropped 0
    output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
    collisions 14
```

c)

Przykład 2. Dwa sposoby konfigurowania interfejsu: a) przez terminal, b) w pliku /usr/local/etc/zebra.conf oraz c) diagnozowanie interfejsu.

Ostatni przykład pokazuje, jak dodać statyczną trasę oraz jak wyświetlić zawartość tablicy tras.

```
Router# configure terminal
Router(config)# ip route 192.168.3.0/24 192.168.2.2
Router(config)# end

Router# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       B - BGP, > - selected route, * - FIB route

K>* 0.0.0.0/0 via 150.254.17.97, eth0
C>* 127.0.0.0/8 is directly connected, lo
C>* 150.254.17.96/27 is directly connected, eth0
C>* 192.168.1.0/24 is directly connected, eth0
C>* 192.168.2.0/24 is directly connected, eth0
S>* 192.168.3.0/24 [1/0] via 192.168.2.2, eth0
```

Przykład 3. Konfigurowanie statycznej trasy

3. Literatura

1. Strona domowa projektu zebra: www.zebra.org
2. Dokument „Linux Advanced Routing & Traffic Control HOWTO”, dostępny pod adresem <http://lartc.org/howto/index.html>