helion.pl
KSIĘGARNIA INTERNETOWA · WYDAWNICTWO

IDŹ DO
PRZYKŁADOWY ROZDZIAŁ
SPIS TREŚCI

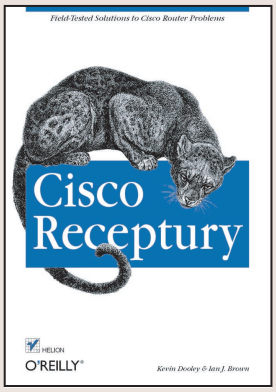
KATALOG KSIĄŻEK
KATALOG ONLINE
ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK
DODAJ DO KOSZYKA

CENNIK I INFORMACJE
ZAMÓW INFORMACJE O NOWOŚCIACH
ZAMÓW CENNIK

CZYTELNIA
FRAGMENTY KSIĄŻEK ONLINE

Cisco. Receptury



Autorzy: Kevin Dooley, Ian J. Brown
Tłumaczenie: Marek Pałczyński, Grzegorz Werner, Witold Zioło
ISBN: 83-7361-330-7
Tytuł oryginału: Cisco Cookbook
Format: B5, stron: 954

System operacyjny IOS firmy Cisco charakteryzuje się sporymi możliwościami działania i elastycznością, ale jednocześnie jest bardzo skomplikowany i trudno dostępny. Większość zadań można w nim zrealizować na kilka sposobów, a nikt nie chce przecież marnować cennego czasu na poszukiwanie najwłaściwszego rozwiązania.

Dlatego właśnie napisano tę książkę. Na szczęście większość zadań konfiguracyjnych może zostać podzielona na kilka niezależnych etapów – konfigurację interfejsu, mechanizmów obsługi protokołów routingu, łączy zapasowych, implementację algorytmów, filtrowania pakietów i włączanie innych systemów kontroli dostępu. To co faktycznie potrzebne jest administratorowi sieci, to zestaw dobrych receptur, które będą zawierać informacje niezbędne do realizacji najczęściej powtarzających się zadań. Gwarantują one właściwą konfigurację sieci i dają pewność, że zastosowane rozwiązania są właściwe i spełniają oczekiwania administratora.

Książka „Cisco. Receptury" zawiera przykłady rozwiązań większości występujących problemów konfiguracyjnych, w tym:

- konfigurowanie wielu rodzajów interfejsów, od interfejsów szeregowych, przez ATM po Frame Relay,
- konfigurowanie mechanizmów obsługi wszystkich powszechnie stosowanych protokołów routingu (RIP, EIGRP, OSPF o BGP),
- konfigurowanie systemu uwierzytelniania,
- konfigurowanie usług dodatkowych, takich jak DHCP i NAT,
- parametryzowanie łączy zapasowych i wykorzystanie protokołu HSRP do zarządzania routerami zapasowymi,
- zarządzenie routerami z uwzględnieniem usług SNMP i innych,
- wykorzystanie list dostępu do sterowania przepływem danych.

Książka ta z pewnością przyda się osobom, które na co dzień posługują się routerami firmy Cisco. Pomoże ona szybko i skutecznie rozwiązać wszystkie pojawiające się problemy. Nawet doświadczeni administratorzy sieci z pewnością skorzystają z zawartych w niej propozycji rozwiązań i szczegółowych opisów, pozwalających na nowe spojrzenie na określone zagadnienia. Osoby, które nie mają doświadczenia w pracy z routerami, a otrzymały zadanie zarządzania siecią wyposażoną w urządzenia Cisco, mogą dzięki tej książce zaoszczędzić sobie sporo pracy.

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

Wstęp ***15***

Rozdział 1. Konfigurowanie routera i zarządzanie plikami ***19***

1.0. Wprowadzenie 19
1.1. Konfigurowanie routera za pośrednictwem protokołu TFTP 22
1.2. Zapisywanie konfiguracji routera na serwerze 24
1.3. Uruchamianie routera z wykorzystaniem zdalnego pliku konfiguracyjnego 26
1.4. Pliki konfiguracyjne większe niż pojemność NVRAM 29
1.5. Usuwanie konfiguracji startowej 31
1.6. Pobieranie nowego obrazu IOS 34
1.7. Uruchamianie różnych obrazów IOS 37
1.8. Uruchamianie za pośrednictwem sieci 41
1.9. Kopiowanie obrazu IOS na serwer 43
1.10. Kopiowanie obrazu IOS za pomocą konsoli 44
1.11. Usuwanie plików z pamięci flash 47
1.12. Partycjonowanie pamięci flash 49
1.13. Wykorzystanie routera jako serwera TFTP 51
1.14. Wykorzystanie usługi FTP routera 53
1.15. Przygotowanie większej liczby plików konfiguracyjnych routera 55
1.16. Jednorazowa zmiana konfiguracji wielu routerów 57
1.17. Pobieranie informacji o wyposażeniu 61
1.18. Sporządzanie kopii zapasowej konfiguracji routera 63

Rozdział 2. Zarządzanie routerem 67
2.0. Wprowadzenie 67
2.1. Tworzenie aliasów poleceń 68
2.2. Zarządzaniem tablicą ARP routera 70
2.3. Dostosowywanie parametrów buforów routera 73
2.4. Protokół wyszukiwania Cisco Discovery Protocol 78
2.5. Wyłączanie obsługi protokołu CDP 82
2.6. Wykorzystanie „małych serwerów" 83
2.7. Dostęp do routera z wykorzystaniem protokołu HTTP 87
2.8. Korzystanie ze statycznych tablic nazw stacji 90
2.9. Korzystanie z systemu nazw domenowych 92
2.10. Wyłączanie odwzorowania nazw domenowych 95
2.11. Określanie czasu ponownego uruchomienia routera 97
2.12. Awaryjne zrzuty pamięci do pliku 100
2.13. Generowanie raportów zawierających dane o interfejsach 102
2.14. Generowanie raportu zawierającego informacje o tablicy routingu 105
2.15. Generowanie raportu zawierającego informacje z tablicy ARP 107
2.16. Generowanie pliku nazw stacji 109

Rozdział 3. Prawa dostępu i przywileje użytkowników 113
3.0. Wprowadzenie 113
3.1. Identyfikatory użytkowników 114
3.2. Szyfrowanie haseł 118
3.3. Doskonalsze techniki szyfrowania 119
3.4. Usuwanie haseł z pliku konfiguracyjnego routera 121
3.5. Deszyfracja haseł zaszyfrowanych standardowym algorytmem firmy Cisco 123
3.6. Wyświetlanie informacji o aktywnych użytkownikach 125
3.7. Wysyłanie komunikatów do innych użytkowników 128
3.8. Zmiana liczby portów VTY 130
3.9. Zmiana dopuszczalnego czasu korzystania z terminala VTY 132
3.10. Ograniczenie dostępu do terminali VTY przez wyznaczenie określonych protokołów 134
3.11. Ustawianie czasu komunikacji z wykorzystaniem linii VTY 136
3.12. Komunikaty 137
3.13. Wyłączanie publikowania komunikatów na poszczególnych portach 141
3.14. Wyłączanie linii routera 142
3.15. Zarezerwowanie jednego portu VTY dla administratora 144
3.16. Ograniczenie dostępu do usługi Telnet 146

3.17. Zapisywanie informacji o logowaniu z wykorzystaniem protokołu Telnet 147
3.18. Definiowanie adresu IP dla połączeń w protokole Telnet 148
3.19. Automatyzacja procedury logowania 149
3.20. Bezpieczny dostęp z wykorzystaniem usługi SSH 152
3.21. Zmiana poziomu uprawnień dla poleceń IOS 156
3.22. Definiowanie uprawnień użytkowników 159
3.23. Definiowanie uprawnień portu 162

Rozdział 4. TACACS+ 165

4.0. Wprowadzenie 165
4.1. Centralny system uwierzytelniania użytkowników 167
4.2. Ograniczanie dostępu do poleceń 170
4.3. Brak komunikacji z serwerem TACACS+ 172
4.4. Wyłączanie uwierzytelniania TACACS+ dla wybranych linii 174
4.5. Przechwytywanie informacji o wprowadzonych ciągach tekstowych 176
4.6. Zapisywanie zdarzeń systemowych 177
4.7. Ustalanie określonego źródłowego adresu IP dla komunikatów TACACS+ 179
4.8. Pobieranie darmowego oprogramowania serwera TACACS+ 180
4.9. Przykładowy plik konfiguracyjny serwera 181

Rozdział 5. Routing IP 187

5.0. Wprowadzenie 187
5.1. Wyszukiwanie trasy 190
5.2. Wyświetlanie tras określonego rodzaju 192
5.3. Zmiana formatu maski 194
5.4. Routing statyczny 198
5.5. Routing zamienny 202
5.6. Wyznaczanie tras na podstawie adresu źródłowego i założonej polityki routingu 204
5.7. Wyznaczanie tras na podstawie rodzaju aplikacji i określonej polityki routingu 208
5.8. Testowanie polityki routingu 211
5.9. Zmiana odległości administracyjnej 212
5.10. Przesyłanie pakietów różnymi trasami o jednakowym koszcie 216

Rozdział 6. Protokół RIP 219

6.0. Wprowadzenie 219
6.1. Konfiguracja protokołu RIP w wersji pierwszej 221
6.2. Filtrowanie tras protokołu RIP 224
6.3. Rozpowszechnianie informacji o trasach statycznych za pomocą protokołu RIP 227

6.4. Redystrybucja tras z wykorzystaniem odwzorowania tras 230
6.5. Trasa domyślna w protokole RIP 233
6.6. Wyłączanie obsługi protokołu RIP w interfejsie 234
6.7. Wysyłanie uaktualnień RIP do jednej stacji 237
6.8. Dodawanie stałej wartości do metryk tras 239
6.9. Zmiana zależności czasowych 241
6.10. Zmiana przerwy między pakietami 244
6.11. Wyzwalane uaktualnienia 246
6.12. Zwiększanie pojemności bufora wejściowego 248
6.13. Konfigurowanie protokołu RIP w wersji drugiej 249
6.14. Włączanie uwierzytelniania RIP 252
6.15. Uogólnianie tras RIP 254
6.16. Znaczniki tras 257

Rozdział 7. Protokół EIGRP 261

7.0. Wprowadzenie 261
7.1. Konfigurowanie protokołu EIGRP 263
7.2. Filtrowanie tras protokołu EIGRP 266
7.3. Redystrybucja tras w protokole EIGRP 270
7.4. Redystrybucja tras z wykorzystaniem odwzorowania tras 274
7.5. Trasa domyślna w protokole EIGRP 275
7.6. Wyłączenie obsługi protokołu EIGRP w określonym interfejsie 277
7.7. Uogólnianie tras w protokole EIGRP 279
7.8. Zmiana metryk EIGRP 282
7.9. Zależności czasowe 284
7.10. Uwierzytelnianie w protokole EIGRP 286
7.11. Rejestrowanie zmian w połączeniach z sąsiednimi routerami EIGRP 288
7.12. Ograniczanie wykorzystania pasma w protokole EIGRP 290
7.13. Routing EIGRP w sieciach wyniesionych 291
7.14. Oznaczanie tras 292
7.15. Status mechanizmu EIGRP 294

Rozdział 8. Protokół OSPF 299

8.0. Wprowadzenie 299
8.1. Konfigurowanie obsługi protokołu OSPF 305
8.2. Filtrowanie tras w protokole OSPF 307
8.3. Zmiana kosztu 309
8.4. Trasa domyślna w protokole OSPF 312
8.5. Redystrybucja tras statycznych w protokole OSPF 314

8.6. Redystrybucja tras zewnętrznych w protokole OSPF 316
8.7. Wybór routera DR 318
8.8. Ustawianie wartości RID protokołu OSPF 321
8.9. Uwierzytelnianie w protokole OSPF 323
8.10. Wybór odpowiedniego typu obszaru 327
8.11. Uogólnianie tras OSPF 335
8.12. Wyłączanie obsługi protokołu OSPF na wybranych interfejsach 338
8.13. Oznaczenie tras OSPF 340
8.14. Rejestrowanie zmian statusu sąsiednich routerów OSPF 341
8.15. Zależności czasowe protokołu OSPF 343
8.16. Przeglądanie informacji o działaniu protokołu OSPF z uwzględnieniem nazw domenowych 345
8.17. Debugowanie procesu OSPF 346

Rozdział 9. Protokół BGP 347

9.0. Wprowadzenie 347
9.1. Konfiguracja protokołu BGP 356
9.2. Opcja eBGP-multihop 362
9.3. Zmiana wartości atrybutu NEXT_HOP 364
9.4. Korzystanie z łączy dwóch dostawców ISP 365
9.5. Podłączenie do sieci dwóch dostawców ISP za pomocą redundantnych routerów 369
9.6. Ograniczanie rozpowszechniania informacji BGP 371
9.7. Zmiana wartości preferencji lokalnych 375
9.8. Rozkładanie ruchu 379
9.9. Usuwanie prywatnych identyfikatorów ASN z listy AS_PATH 381
9.10. Filtrowanie tras BGP na podstawie wartości AS_PATH 383
9.11. Zmniejszanie rozmiaru odbieranych tablic routingu 387
9.12. Uogólnianie wysyłanych informacji o trasach 390
9.13. Dodawanie identyfikatorów ASN do atrybutu AS_PATH 394
9.14. Redystrybucja tras w protokole BGP 396
9.15. Grupowanie sąsiednich routerów BGP 400
9.16. Uwierzytelnianie routerów 402
9.17. Łączenie różnych technik 404

Rozdział 10. Protokół Frame Relay 407

10.0. Wprowadzenie 407
10.1. Konfiguracja protokołu Frame Relay w podinterfejsach punkt-punkt 410
10.2. Opcje protokołu LMI 415

10.3. Wykorzystanie poleceń map podczas konfigurowania obsługi protokołu Frame Relay 417
10.4. Wykorzystanie podinterfejsów transmisji wielopunktowej 419
10.5. Konfigurowanie łączy SVC sieci Frame Relay 421
10.6. Symulacja sieci Frame Relay 424
10.7. Kompresja danych Frame Relay 426
10.8. Kompresja danych Frame Relay za pomocą polecenia map 428
10.9. Przeglądanie informacji o stanie łączy sieci Frame Relay 430

Rozdział 11. Kolejkowanie i przeciążenie sieci 433
11.0. Wprowadzenie 433
11.1. Szybkie przełączanie i mechanizm CEF 437
11.2. Ustawianie wartości pola DSCP i TOS 441
11.3. Priorytety kolejek 444
11.4. Kolejki użytkownika 447
11.5. Kolejki użytkownika a priorytety kolejek 451
11.6. Kolejkowanie WFQ 452
11.7. Kolejkowanie WFQ z uwzględnieniem klas 454
11.8. Unikanie przeciążeń — algorytm WRED 457
11.9. Protokół RSVP 460
11.10. Ogólne metody kształtowania ruchu 463
11.11. Kształtowanie ruchu w sieciach Frame Relay 465
11.12. Dopuszczalna szybkość transmisji — algorytm CAR 467
11.13. Implementacja sposobu działania zgodnego z zaleceniami RFC 472
11.14. Przeglądanie parametrów kolejek 476

Rozdział 12. Tunele oraz sieci VPN 479
12.0. Wstęp 479
12.1. Tworzenie tunelu 484
12.2. Tunelowanie obcych protokołów w IP 488
12.3. Tunelowanie, a protokoły routowania dynamicznego 490
12.4. Przeglądanie stanu tunelu 493
12.5. Tworzenie szyfrowanych sieci VPN łączących routery 495
12.6. Generowanie kluczy RSA 502
12.7. Tworzenie między routerami sieci VPN wykorzystującej klucze RSA 505
12.8. Tworzenie sieci VPN pomiędzy stacją roboczą a routerem 509
12.9. Kontrola stanu protokołu IPSec 512

Rozdział 13. Komutowane łącza zapasowe 517

13.0. Wstęp 517
13.1. Automatyczne nawiązywanie komutowanych połączeń zapasowych 521
13.2. Użycie interfejsów dialera 528
13.3. Użycie modemu asynchronicznego podłączonego do portu AUX 532
13.4. Użycie interfejsów zapasowych 534
13.5. Użycie funkcji dozoru dialera 537
13.6. Zagwarantowanie poprawnego rozłączenia 539
13.7. Poznanie stanu komutowanego połączenia zapasowego 540
13.8. Usuwanie problemów z zapasowymi połączeniami komutowanymi 544

Rozdział 14. Czas i protokół NTP 547

14.0. Wstęp 547
14.1. Oznaczanie czasem pozycji dzienników zdarzeń routera 549
14.2. Ustawianie zegara 552
14.3. Konfiguracja strefy czasowej 553
14.4. Konfiguracja czasu letniego 555
14.5. Synchronizacja czasu w routerach (protokół NTP) 556
14.6. Konfiguracja nadmiarowości w protokole NTP 560
14.7. Konfiguracja routera jako NTP Master 562
14.8. Zmiana okresu synchronizacji protokołu NTP 564
14.9. Użycie protokołu NTP do okresowego rozgłaszania uaktualnień czasu 564
14.10. Użycie protokołu NTP do okresowej multiemisji uaktualnień czasu 566
14.11. Włączanie i wyłączanie protokołu NTP w poszczególnych interfejsach 568
14.12. Uwierzytelnianie NTP 570
14.13. Ograniczanie liczby urządzeń równorzędnych 572
14.14. Ograniczanie urządzeń równorzędnych 573
14.15. Konfiguracja okresu zegara 574
14.16. Sprawdzanie stanu protokołu NTP 575
14.17. Rozwiązywanie problemów z protokołem NTP 577

Rozdział 15. DLSw 581

15.0. Wstęp 581
15.1. Konfiguracja DLSw 586
15.2. Użycie DLSw do mostkowania pomiędzy sieciami Ethernet i Token Ring 593
15.3. Konwersja adresów Ethernet na Token Ring 596
15.4. Konfiguracja SDLC 599
15.5. Konfiguracja SDLC w przypadku połączeń wielopunktowych 603

15.6. Użycie połączeń STUN 604
15.7. Użycie połączeń BSTUN 607
15.8. Kontrola fragmentacji pakietów DLSw 609
15.9. Znacznikowanie pakietów DLSw w celu zapewnienia wysokiej jakości usług (QoS) 610
15.10. Obsługa priorytetów SNA 612
15.11. Nadmiarowość i odporność na uszkodzenia w DLSw+ 614
15.12. Poznanie stanu DLSw 615
15.13. Poznanie stanu SDLC 616
15.14. Rozwiązywanie problemów z połączeniami DLSw 619

Rozdział 16. Interfejsy routera oraz media 625

16.0. Wstęp 625
16.1. Poznanie stanu interfejsu 626
16.2. Konfiguracja interfejsów szeregowych 634
16.3. Wykorzystanie wewnętrznej jednostki CSU/DSU linii T1 639
16.4. Wykorzystanie wewnętrznego modułu ISDN PRI 641
16.5. Wykorzystanie wewnętrznej jednostki CSU/DSU 56 Kbps 642
16.6. Konfiguracja asynchronicznego interfejsu szeregowego 645
16.7. Konfiguracja podinterfejsów ATM 646
16.8. Konfiguracja kodowania ładunku w obwodzie ATM 649
16.9. Konfiguracja parametrów interfejsu Ethernet 651
16.10. Konfiguracja parametrów interfejsu Token Ring 653
16.11. Konfiguracja trunków sieci VLAN wykorzystujących ISL 655
16.12. Konfiguracja trunków sieci VLAN wykorzystujących protokół 802.1Q 658

Rozdział 17. Simple Network Management Protocol 663

17.0. Wprowadzenie 663
17.1. Konfigurowanie SNMP 667
17.2. Pobieranie informacji z routera za pomocą narzędzi SNMP 670
17.3. Zapisywanie ważnych informacji o routerze do późniejszego pobrania przez SNMP 673
17.4. Pobieranie informacji inwentaryzacyjnych z listy routerów za pośrednictwem SNMP 675
17.5. Zabezpieczanie dostępu SNMP za pomocą list dostępu 677
17.6. Rejestrowanie prób nieautoryzowanego dostępu SNMP 679
17.7. Ograniczanie dostępu do bazy MIB 681
17.8. Modyfikowanie bieżącej konfiguracji routera za pośrednictwem SNMP 684
17.9. Kopiowanie nowego obrazu IOS za pośrednictwem SNMP 687
17.10. Hurtowa zmiana konfiguracji za pośrednictwem SNMP 689
17.11. Zapobieganie nieautoryzowanym zmianom konfiguracji 692

17.12. Utrwalanie numerów interfejsów 693
17.13. Włączanie pułapek i komunikatów inform SNMP 696
17.14. Wysyłanie komunikatów syslog w postaci pułapek i komunikatów inform SNMP 699
17.15. Ustawianie rozmiaru pakietu SNMP 701
17.16. Ustawianie rozmiaru kolejki SNMP 702
17.17. Ustawianie limitów czasu SNMP 704
17.18. Wyłączanie pułapek informujących o aktywacji i dezaktywacji łącza interfejsu 705
17.19. Ustawianie źródłowego adresu IP pułapek SNMP 706
17.20. Używanie mechanizmu RMON do wysyłania pułapek 707
17.21. Włączanie obsługi protokołu SNMPv3 712
17.22. Korzystanie z SAA 717

Rozdział 18. Rejestrowanie....723

18.0. Wprowadzenie 723
18.1. Włączanie lokalnego rejestrowania w routerze 725
18.2. Ustawianie rozmiaru dziennika 727
18.3. Usuwanie zawartości dziennika routera 728
18.4. Wysyłanie komunikatów dziennika na ekran 729
18.5. Korzystanie ze zdalnego serwera rejestrowania 731
18.6. Włączanie mechanizmu syslog w serwerze uniksowym 732
18.7. Zmiana domyślnej kategorii rejestrowania 734
18.8. Ograniczanie typów komunikatów dziennika wysyłanych do serwera 736
18.9. Ustawianie źródłowego adresu IP w komunikatach syslog 738
18.10. Rejestrowanie komunikatów syslog routera w różnych plikach 739
18.11. Porządkowanie plików syslog w serwerze 740
18.12. Testowanie konfiguracji serwera syslog 742
18.13. Zapobieganie rejestrowaniu najczęstszych komunikatów 744
18.14. Ograniczanie natężenia ruchu syslog 745

Rozdział 19. Listy dostępu....747

19.0. Wprowadzenie 747
19.1. Filtrowanie ruchu według adresu źródłowego lub docelowego 749
19.2. Dodawanie komentarza do listy ACL 753
19.3. Filtrowanie ruchu według aplikacji 754
19.4. Filtrowanie według znaczników w nagłówku TCP 760
19.5. Ograniczanie kierunku sesji TCP 761
19.6. Filtrowanie ruchu aplikacji korzystających z wielu portów 763
19.7. Filtrowanie według pól DSCP i TOS 765

19.8. Rejestrowanie przypadków użycia listy dostępu 766
19.9. Rejestrowanie sesji TCP 768
19.10. Analizowanie wpisów dziennika ACL 770
19.11. Korzystanie z nazwanych i zwrotnych list dostępu 773
19.12. Obsługa pasywnego trybu FTP 776
19.13. Używanie kontekstowych list dostępu 777

Rozdział 20. DHCP 783

20.0. Wprowadzenie 783
20.1. Korzystanie z adresu pomocnika IP 785
20.2. Ograniczanie wpływu adresów pomocnika IP 786
20.3. Dynamiczne konfigurowanie adresów IP routera za pomocą DHCP 788
20.4. Dynamiczne przydzielanie adresów IP klientom za pomocą DHCP 790
20.5. Definiowanie opcji konfiguracyjnych DHCP 792
20.6. Definiowanie okresu dzierżawy DHCP 795
20.7. Przydzielanie statycznych adresów IP za pomocą DHCP 796
20.8. Konfigurowanie klienta bazy danych DHCP 798
20.9. Konfigurowanie wielu serwerów DHCP do obsługi jednej podsieci 800
20.10. Wyświetlanie stanu DHCP 801
20.11. Debugowanie DHCP 803

Rozdział 21. NAT 805

21.0. Wprowadzenie 805
21.1. Konfigurowanie podstawowych funkcji NAT 807
21.2. Dynamiczne przydzielanie adresów zewnętrznych 809
21.3. Statyczne przydzielanie adresów zewnętrznych 810
21.4. Tłumaczenie niektórych adresów w sposób statyczny, a innych w sposób dynamiczny 811
21.5. Jednoczesne tłumaczenie adresów w obu kierunkach 813
21.6. Przepisywanie prefiksu sieci 815
21.7. Regulowanie zegarów NAT 816
21.8. Zmiana portów TCP używanych przez FTP 818
21.9. Sprawdzanie stanu NAT 819
21.10. Debugowanie NAT 821

Rozdział 22. Hot Standby Router Protocol 823

22.0. Wprowadzenie 823
22.1. Konfigurowanie podstawowych funkcji HSRP 828
22.2. Korzystanie z wywłaszczania HSRP 832

22.3. Reagowanie na problemy z innymi interfejsami 835
22.4. Równoważenie obciążenia z wykorzystaniem HSRP 837
22.5. Przekierowania ICMP w połączeniu z HSRP 840
22.6. Modyfikowanie zegarów HSRP 841
22.7. Używanie HSRP w sieci Token Ring 843
22.8. Obsługa SNMP w HSRP 846
22.9. Zwiększanie bezpieczeństwa HSRP 847
22.10. Wyświetlanie informacji o stanie HSRP 850
22.11. Debugowanie HSRP 851

Rozdział 23. Multicast IP 853

23.0. Wprowadzenie 853
23.1. Podstawowe przekazywanie ruchu multicast za pomocą protokołu PIM-DM 861
23.2. Routing multicast z wykorzystaniem PIM-SM i BSR 863
23.3. Routing multicast z wykorzystaniem PIM-SM i Auto-RP 867
23.4. Konfigurowanie routingu na użytek aplikacji multicast o niskiej częstotliwości transmisji 870
23.5. Konfigurowanie CGMP 873
23.6. Statyczne trasy multicast i członkostwa grupowe 874
23.7. Routing ruchu multicast z wykorzystaniem protokołu MOSPF 875
23.8. Routing ruchu multicast z wykorzystaniem protokołu DVMRP 877
23.9. Tunele DVMRP 880
23.10. Ograniczanie zasięgu multicast za pomocą TTL 881
23.11. Adresowanie z zasięgiem wyznaczonym administracyjnie 883
23.12. Wymiana informacji o routingu multicast za pomocą MBGP 886
23.13. Wykrywanie zewnętrznych źródeł za pomocą MSDP 888
23.14. Przekształcanie transmisji broadcast w multicast 890
23.15. Wyświetlanie informacji o stanie protokołów multicast 892
23.16. Debugowanie routingu multicast 902

Dodatek A Dodatkowe pakiety oprogramowania 905

Dodatek B Klasyfikacje IP Precedence, TOS i DSCP 909

Skorowidz 923

2

Zarządzanie routerem

2.0. Wprowadzenie

Niniejszy rozdział, podobnie jak poprzedni, zawiera informacje o sposobach zarządzania pracą routera. W poprzedniej części książki poruszano zagadnienia związane z ogólną administracją urządzeniami, w tym problematykę zarządzania systemem plików. W tym rozdziale większość tematów dotyczy spraw zarządzania oraz dostosowywania konfiguracji routerów, które mają na celu zwiększenie wydajności urządzeń. Zaprezentowane zostały również niektóre sposoby reagowania na sytuacje awaryjne, w tym, na przykład, wykonywanie awaryjnych zrzutów pamięci.

Systemy IOS Cisco obsługują wiele protokołów i usług przeznaczonych do wykonywania określonych zadań. Niektóre z nich znajdują szczególne zastosowanie w zarządzaniu i administrowaniu urządzeniami, natomiast inne są nieocenione przy testowaniu określonych rozwiązań. Jednym z najbardziej użytecznych elementów systemu jest protokół wyszukiwania CDP (ang. *Cisco Discovery Protocol*), który umożliwia gromadzenie wielu informacji na temat połączeń między urządzeniami Cisco realizowanymi w warstwie drugiej modelu OSI. Niniejszy rozdział opisuje wspomniany protokół, uwzględniając również niektóre znane problemy związane z bezpieczeństwem.

W przypadku innych usług zazwyczaj najlepszym rozwiązaniem jest ich wyłączenie. Niektóre z interfejsów zarządzania (np. HTTP) lub protokołów testowych (określanych jako „małe serwery" TCP i UDP), nie odgrywają większej roli w zarządzaniu routerami i domyślnie są wyłączane. Inne protokoły (np. DNS) pełnią bardzo użyteczne funkcje i są z założenia włączone.

W rozdziale tym opisano także kilka istotnych opcji administracyjnych, takich jak definiowanie nazw dla innych urządzeń sieciowych i definiowanie aliasów dla poleceń, które ułatwiają zapamiętywanie i posługiwanie się skomplikowanymi instrukcjami. W końcowej części rozdziału przedstawiono cztery skrypty, których zadanie polega na gromadzeniu istotnych informacji na temat funkcjonujących w sieci urządzeń.

2.1. Tworzenie aliasów poleceń

Problem

Chcemy utworzyć aliasy dla często wykorzystywanych poleceń lub dla instrukcji, których składnia jest szczególnie skomplikowana.

Rozwiązanie

Do tworzenia aliasów poleceń routera służy instrukcja `alias`:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#alias exec rt show ip route
Router1(config)#alias exec on show ip ospf neighbor
Router1(config)#end
Router1#
```

Analiza

Zagadnienie tworzenia aliasów dla wyjątkowo złożonych poleceń systemowych jest znane administratorom systemów Unix od wielu lat. Dzięki nim upraszcza się procedurę wprowadzania instrukcji i oszczędza tym samym czas. Dzięki zastosowaniu aliasów długie polecenia zostają skrócone do kilku znaków. Technika ta ma szczególnie duże zastosowanie przy upraszczaniu często wykonywanych instrukcji lub tych, które są na tyle skomplikowane, że ich zapamiętanie sprawia duże trudności. Aliasy można tworzyć dla dowolnych poleceń, zawierając w nich również część lub wszystkie stosowane w danej instrukcji opcje i słowa kluczowe.

W prezentowanym przykładzie utworzono alias o nazwie `rt`, który zastępuję wykorzystywane niemal codziennie polecenie `show ip route`:

```
Router1(config)#alias exec rt show ip route
```

Dzięki zastosowaniu dwuliterowego aliasu można skrócić czas wprowadzania polecenia wyświetlającego tablicę routingu:

```
Router1#rt
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 172.25.1.1 to network 0.0.0.0

S    192.168.10.0/24 [1/0] via 172.22.1.4
     172.16.0.0/24 is subnetted, 1 subnets
```

```
C       172.16.2.0 is directly connected, FastEthernet0/0.2
     172.20.0.0/16 is variably subnetted, 3 subnets, 3 masks
O       172.20.10.0/24 [110/74] via 172.20.1.2, 00:52:55, Serial0/0.2
C       172.20.1.0/30 is directly connected, Serial0/0.2
O       172.20.100.1/32 [110/65] via 172.20.1.2, 00:52:55, Serial0/0.2
     172.22.0.0/16 is variably subnetted, 2 subnets, 2 masks
D       172.22.0.0/16 is a summary, 20:31:03, Null0
C       172.22.1.0/24 is directly connected, FastEthernet0/1
Router1#
```

Przy wyborze odpowiedniej nazwy dla aliasu polecenia należy pamiętać o tym, że powinna ona być krótka i łatwa do zapamiętania. Oczywiście konieczne jest dobranie takiej nazwy, które nie koliduje z jakimkolwiek z istniejących poleceń. We wcześniejszym przykładzie wybrano ciąg `rt`, gdyż nie pokrywa się on z nazwą żadnego z poleceń i jest jednocześnie akronimem od słów *routing table* (*tablica routingu*).

Utworzony alias może być wykorzystywany także jako element składowy dłuższego polecenia. Przykładowo, instrukcję `show ip route 172.16.2.0` można skrócić za pomocą aliasu `rt` w następujący sposób:

```
Router1#rt 172.16.2.0
Routing entry for 172.16.2.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
  * directly connected, via FastEthernet0/0.2
      Route metric is 0, traffic share count is 1
Router1#
```

Aliasy poleceń są szczególnie użyteczne, jeżeli stosuje się je konsekwentnie we wszystkich zarządzanych routerach. W przeciwnym przypadku konieczne jest zapamiętywanie różnych ciągów tekstowych wykorzystywanych w pracy z różnymi grupami urządzeń. Zaleca się, aby przy wdrażaniu tego typu rozwiązań, w definiowaniu aliasów uczestniczyli wszyscy członkowie zespołu zarządzania routerami, co pozwoli na opracowanie standardowego zestawu nazw, wykorzystywanego przez cały zespół. Zaleca się również definiowanie jak najłatwiejszych do zapamiętania ciągów tekstowych, ale przede wszystkim odradza się tworzenie aliasów dla wszystkich możliwych do wykorzystania poleceń. Rozwiązanie to powinno być stosowane jedynie w odniesieniu do instrukcji używanych najczęściej.

Stosowanie aliasów bywa również bardzo użyteczne podczas pisania skryptów. Można bowiem za ich pomocą tworzyć skrypty, które wykonują w każdym z routerów to samo zadanie, ale robią to w nieco inny sposób. Jednym z przykładów może być cotygodniowa operacja zerowania liczników poszczególnych list dostępowych. Problem tkwi w tym, że różne routery korzystają z różnych numerów list dostępowych. Można zatem w każdym z urządzeń utworzyć alias o takiej samej nazwie, ale przypisać poszczególnym z nich inny, stosowny zestaw poleceń. W końcu można utworzyć skrypt, który będzie wywoływał alias z linii poleceń, automatyzując w ten sposób proces, który w innym przypadku byłby ogromnie pracochłonny.

Aby wyświetlić listę wszystkich zdefiniowanych dla danego routera aliasów poleceń, należy użyć polecenia `show aliases`:

```
Router1#show aliases
Exec mode aliases:
  h                    help
  lo                   logout
  p                    ping
  r                    resume
  s                    show
  u                    undebug
  un                   undebug
  w                    where
  rt                   show ip route
  on                   show ip ospf neighbor

Router1#
```

Wywołanie tego polecenia w dowolnym routerze Cisco pozwala zauważyć, że niektóre z aliasów są tworzone domyślnie przez producenta.

2.2. Zarządzaniem tablicą ARP routera

Problem

Chcemy zmienić czas przechowywania danych w tablicy ARP routera.

Rozwiązanie

Aby zmienić wartość czasu przechowywania danych w tablicy ARP, należy zastosować polecenie `arp timeout`:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#interface Ethernet0
Router1(config-if)#arp timeout 600
Router1(config-if)#end
Router1#
```

Analiza

Każde urządzenie pracujące w sieci LAN zawiera tablicę danych protokołu odwzorowania adresów — tablicę danych ARP (ang. *Address Resolution Protocol*). Na podstawie wspomnianej tablicy dokonuje się odwzorowania adresów warstwy drugiej (adresów MAC) na adresy warstwy trzeciej (adresy IP). Gdyby tablica ARP nie była dostępna, urządzenie mogłoby budować pakiety protokołu IP, ale nie miałoby możliwości formowania ramek warstwy drugiej modelu OSI, które odpowiadają za przenoszenie pakietów.

Pozyskiwanie informacji dla tablicy ARP jest procesem dynamiczny. Gdy urządzenie przystępuje do przesłania pakietu do stacji o określonym docelowym adresie IP, ale nie posiada odpowiadającego temu adresowi IP adresu MAC, wysyła w sposób rozgłoszeniowy

pakiet protokołu ARP. Pakiet ARP dociera do wszystkich urządzeń przyłączonych do danego segmentu sieci LAN. Urządzenie, które posiada znany adres IP, odsyła odpowiedź ARP zawierającą poszukiwany adres MAC.

Dodatkowo, wiele urządzeń sieciowych wysyła pakiet *powiadamiania* ARP zaraz po przyłączeniu do sieci. Pakiet tego typu nie jest odpowiedzią na żadne zapytanie ARP, niemniej każda z przyłączonych do sieci stacji odbiera go i może zapisać informacje w nim przenoszone we własnej tablicy ARP, z której może skorzystać w przypadku ewentualnej późniejszej komunikacji z danym urządzeniem.

Procedura przesyłania zapytań i odpowiedzi ARP wprowadza opóźnienia w transmisji danych, gdyż zajmuje określoną ilość czasu. Ponadto, z uwagi na rozgłoszeniowy charakter takiej wymiany informacji, przesyłane pakiety są analizowane przez wszystkie stacje działające w segmencie sieci, co z kolei powoduje konieczność przerwania dotychczasowych zadań wykonywanych przez poszczególne urządzenia. Wysłanie zbyt dużej liczby pakietów ARP generuje nadmierny ruch sieciowy oraz powoduje zużywanie znacznej ilości zasobów przyłączonych do sieci urządzeń.

Aby ograniczyć wymianę danych ARP, wszystkie stacje posługujące się protokołem IP muszą posiadać tablice ARP, w których zapisują informacje o odwzorowaniu adresów. Co pewien czas najstarsze wpisy (wpisy dokonane przed określonym czasem) są z tablicy usuwane. W środowiskach, w których urządzenia często zmieniają swoje adresy (w sieciach, w których adresy IP są wydzierżawiane przez serwery DHCP na krótki okres) routery muszą dość często oczyszczać własne tablice ARP. Niekiedy zdarza się, że w pamięci ARP routera znajduje się tyle wpisów, że wyszukanie jakiegokolwiek trwa zbyt długo i pochłania zbyt wiele czasu procesora. Najważniejsze w takich przypadkach jest zachowanie równowagi między częstotliwością usuwania nieaktualnych wpisów, a odpowiednim ograniczeniem ruchu ARP.

Domyślnie routery Cisco usuwają niewykorzystywane dane ARP po czterech godzinach. Oznacza to, że jeżeli router przez cztery godziny nie odebrał jakichkolwiek pakietów od określonego urządzenia ani nie wysłał do niego żadnych informacji, usuwa dane ARP o tej stacji ze swojej pamięci podręcznej. Fabryczne ustawienie czasu usuwania wpisów ARP jest właściwe dla większości sieci Ethernet, niemniej w niektórych przypadkach zachodzi potrzeba zwiększenia wydajności sieci przez zmianę wartości tego parametru.

W przykładzie analizowanym w niniejszej recepturze czas usuwania danych ARP został ustawiony na 600 sekund (10 minut):

```
Router1(config-if)#arp timeout 600
```

Oczywiście nic nie stoi na przeszkodzie, żeby wykorzystać powyższe polecenie do zwiększenia czasu przechowywania danych. Generalnie nie zaleca się zmniejszania wartości tego parametru poniżej 5 minut, gdyż zazwyczaj skutkuje to zwiększeniem obciążenia procesora i sieci.

Wpisy przechowywane w danym czasie w pamięci routera można wyświetlić za pomocą polecenia `show ip arp`:

```
Router1#show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  172.25.1.5             8   0001.9670.b780  ARPA   Ethernet0
Internet  172.25.1.7             -   0000.0c92.bc6a  ARPA   Ethernet0
Internet  172.25.1.1             9   0010.4b09.5700  ARPA   Ethernet0
Internet  172.25.1.3             2   0010.4b09.5715  ARPA   Ethernet0
Router1#
```

Wśród wyświetlanych informacji znajdują się dane o adresach IP, czasie przechowywania wpisów, adresach MAC oraz rodzajach interfejsu. Wartość czasu przechowywania jest zerowana po każdorazowym odnotowaniu przez router faktu wymiany informacji z określonym urządzeniem. Zyskuje się w ten sposób gwarancję, że adresy stacji nie zostaną usunięte, niezależnie od tego, jak długo znajdują się w pamięci routera.

Polecenie `show ip arp` umożliwia również wyświetlenie informacji o jednym z wybranych adresów. Opcja ta ułatwia pozyskiwanie danych z dużej tablicy ARP. Jeżeli router pracuje w sieci rozległej, może w swojej pamięci przechowywać setki lub tysiące wpisów. To o wiele za dużo, żeby można je było ogarnąć wzrokiem:

```
Router1#show ip arp 172.25.1.5
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  172.25.1.5             2   0001.9670.b780  ARPA   Ethernet0
Router1#
```

To samo polecenie można wykorzystać do wyświetlenia informacji o określonym adresie MAC:

```
Router1#show ip arp 0010.4b09.5715
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  172.25.1.3             3   0010.4b09.5715  ARPA   Ethernet0
Router1#
```

Możliwe jest również pozyskiwanie danych o adresach ARP dla wybranego interfejsu routera:

```
Router1#show ip arp Ethernet0
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  172.25.1.5             4   0001.9670.b780  ARPA   Ethernet0
Internet  172.25.1.7             -   0000.0c92.bc6a  ARPA   Ethernet0
Internet  172.25.1.1             2   0010.4b09.5700  ARPA   Ethernet0
Internet  172.25.1.3             4   0010.4b09.5715  ARPA   Ethernet0
Router1#
```

W przypadku wystąpienia jakichkolwiek problemów z funkcjonowaniem tablicy ARP lub gdy zachodzi potrzeba natychmiastowego usunięcia wpisów można całkowicie oczyścić pamięć ARP, posługując się poleceniem `clear arp`:

```
Router1#clear arp
Router1#
```

Niestety, nie ma sposobu na usuwanie pojedynczych wpisów z listy. Jeżeli pojawia się konieczność ręcznego usunięcia danej pozycji, jedynym rozwiązaniem jest usunięcie całej zawartości tablicy. Wykonanie takiej operacji skutkuje chwilowym wzrostem natężenia ruchu ARP, gdyż router stara się odbudować pamięć ARP, pozyskując informacje o aktywnych w danej chwili urządzeniach. Należy zatem z umiarem wykorzystywać tę opcję.

Informacje o czasie usuwania danych ARP z tablicy dla wybranego interfejsu są prezentowane po wprowadzeniu polecenia `show interface`:

```
Router1#show interface Ethernet0
Ethernet0 is up, line protocol is up
  Hardware is Lance, address is 0000.0c92.bc6a (bia 0000.0c92.bc6a)
  Internet address is 172.25.1.7/24
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 00:10:00
  <dalsza część została usunięta>
```

2.3. Dostosowywanie parametrów buforów routera

Problem

Chcemy zmienić domyślne ustawienia buforów routera w celu zwiększenia wydajności urządzenia.

Rozwiązanie

W routerze wykorzystywane są dwa różne zestawy buforów — bufory publiczne i bufory interfejsów. Bufory te są wykorzystywane jako obszar pamięci przeznaczony do tymczasowego przechowywania pakietów podczas ich przetwarzania. Parametry buforów publicznych można zmieniać za pomocą poniższego kodu:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#buffers big initial 100
Router1(config)#buffers big max-free 200
Router1(config)#buffers big min-free 50
Router1(config)#buffers big permanent 50
Router1(config)#end
Router1#
```

Zmiana parametrów buforów interfejsu wymaga zastosowania podobnych poleceń:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#buffers Ethernet0 initial 200
Router1(config)#buffers Ethernet0 max-free 300
```

```
Router1(config)#buffers Ethernet0 min-free 50
Router1(config)#buffers Ethernet0 permanent 50
Router1(config)#end
Router1#
```

Analiza

Przed omówieniem samej zmiany rozmiaru buforów, trzeba wspomnieć o trzech rzeczach istotnych przy dokonywaniu wspomnianych zmian. Po pierwsze, dostosowywanie rozmiaru buforów zazwyczaj nie jest konieczne. Po drugie, niewłaściwe dobranie wartości rozmiaru bufora może być przyczyną znacznego obniżenia wydajności pracy routera i spowolnienia ruchu. Po trzecie, nawet w przypadku, gdy zmiana rozmiaru jest konieczna, wartości poszczególnych parametrów powinny być dobierane osobno dla każdej sieci lub dla każdego routera. Z tego względu niniejszą recepturę należy traktować jedynie jako ogólny przykład rozwiązania.

Router zawiera dwa różne zestawy buforów. Pierwszy z nich stanowi pulę publiczną, która może być wykorzystana przez urządzenie do dowolnych celów. Druga to pula interfejsu, która służy jedynie do przetwarzania pakietów danego interfejsu.

Pula buforów publicznych jest dalej dzielona na kilka pul, zależnie od ich rozmiaru. W tabeli 2.1 znajduje się zestawienie pul publicznych buforów.

Tabela 2.1. Pule publiczne buforów routera

Rozmiar bufora	Nazwa puli bufora
104 bajty	Small
600 bajtów	Middle
1536 bajtów	Big
4520 bajtów	VeryBig
5024 bajty	Large
18 024 bajty (domyślny)	Huge

Domyślny rozmiar buforów typu Huge wynosi 18 024 bajty. W przeciwieństwie do innych pul buforów publicznych rozmiar bufora tej puli może ulegać zmianie:

```
Router1(config)#buffers huge size 36048
```

Dopuszczalnymi wartościami rozmiaru bufora Huge są wartości z przedziału od 18 024 do 100 000 bajtów. Z uwagi na fakt, że router może korzystać z pamięci, operując obszarami o rozmiarze bufora, ustawienie bardzo dużych rozmiarów bufora umożliwia operowanie bardzo dużymi pakietami. Niemniej domyślna wartość 18 024 jest zazwyczaj wystarczająca do przetwarzania pakietów o największej wartości parametru MTU, jaki jest wykorzystywany w standardowych typach interfejsów. Zmiana wartości omawianego ustawienia jest bardzo rzadko spotykana. Rozmiary pozostałych buforów są stałe i nie mogą być zmieniane.

W przypadku buforów publicznych możliwe jest modyfikowanie czterech innych parametrów:

```
Router1(config)#buffers big initial 100
Router1(config)#buffers big max-free 200
Router1(config)#buffers big min-free 50
Router1(config)#buffers big permanent 50
```

Pierwsze z poleceń ustawia liczbę buforów danego typu, jakie router będzie wykorzystywał podczas uruchamiania. Jeżeli urządzenie pracuje w sieci o wyjątkowo dużym natężeniu ruchu, zaalokowanie dostatecznej liczby buforów, które sprostają obciążeniu routera, może zająć sporo czasu. Może się wówczas zdarzyć, że przy uruchomianiu urządzenia będą się pojawiały błędy wynikające z niedostatecznej liczby buforów. Rozwiązaniem problemu jest zwiększenie liczby buforów początkowych (`initial`).

Drugie polecenie wykorzystuje słowo kluczowe `max-free` do ustawienia maksymalnej liczby buforów danego typu (w tym przypadków buforów `big`), jakimi system może operować. W trakcie normalnej pracy urządzenia mogą się pojawić przypadki okresowego wzrostu natężenia ruchu, które wymuszą użycie większej liczby buforów. Ustawienie relatywnie niskiej wartości parametru powoduje, że po zakończeniu chwilowego wzrostu natężenia ruchu router zwalnia dodatkowo zaalokowaną pamięć, czyniąc ją dostępną dla innych celów. Z drugiej strony, ustawienie zbyt małej wartości, może sprawić, że w sieci o dużym natężeniu ruchu router nie będzie w stanie dostatecznie szybko alokować nowych buforów, żeby sprostać sytuacji.

Trzecie polecenie zawiera słowo kluczowe `min-free`, które odnosi się do przypadku przeciwnego w stosunku do opisanego wcześniej. Chcąc zapewnić poprawne funkcjonowanie routera w przypadku wzrostu natężenia ruchu, urządzenie rozpoczyna alokowanie pamięci systemowej w chwili, gdy liczba niewykorzystanych buforów spadnie poniżej wartości `min-free`. Jeżeli parametrowi `min-free` zostanie przypisana duża wartość, router będzie miał możliwość sprostania nawet największemu natężeniu ruchu. Jednocześnie ustawienie zbyt dużej wartości powoduje, że router musi wykonać znaczną ilość dodatkowej pracy podczas alokowania buforów, których nigdy nie wykorzysta.

Ostatnia z prezentowanych instrukcji definiuje minimalną liczbę buforów danego typu. Służy do tego słowo kluczowe `permanent`. Podczas uruchamiania router alokuje taką liczbę buforów, jaka jest określona tym parametrem. Pamięć przeznaczona na te bufory nigdy nie zostaje zwrócona do ogólnej puli pamięci systemowej. Właściwą wartością ustawienia jest taka wartość, która gwarantuje ograniczenie pracy routera wynikającej z konieczności alokowania buforów. Jednocześnie zbyt duża wartość parametru może niepotrzebnie zużywać cenne zasoby pamięci systemowej.

Jak nietrudno zauważyć, analizując drugi z przykładów, parametry puli buforów interfejsu są dokładnie takie samej jak ustawienia opisane powyżej:

```
Router1(config)#buffers Ethernet0 initial 200
Router1(config)#buffers Ethernet0 max-free 300
Router1(config)#buffers Ethernet0 min-free 50
Router1(config)#buffers Ethernet0 permanent 50
```

Najlepszym sposobem sprawdzenia, czy bufory wymagają jakichkolwiek korekt, jest zapoznanie się z wynikiem działania polecenia `show buffers`:

```
Router1#show buffers
Buffer elements:
     498 in free list (500 max allowed)
     760166 hits, 0 misses, 0 created

Public buffer pools:
Small buffers, 104 bytes (total 50, permanent 50):
     50 in free list (20 min, 150 max allowed)
     265016 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
Middle buffers, 600 bytes (total 25, permanent 25, peak 49 @ 1d09h):
     23 in free list (10 min, 150 max allowed)
     40749 hits, 10 misses, 30 trims, 30 created
     0 failures (0 no memory)
Big buffers, 1536 bytes (total 50, permanent 50):
     50 in free list (5 min, 150 max allowed)
     33780 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
VeryBig buffers, 4520 bytes (total 10, permanent 10):
     10 in free list (0 min, 100 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
Large buffers, 5024 bytes (total 0, permanent 0):
     0 in free list (0 min, 10 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
Huge buffers, 18024 bytes (total 0, permanent 0):
     0 in free list (0 min, 4 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)

Interface buffer pools:
Ethernet0 buffers, 1524 bytes (total 32, permanent 32):
     8 in free list (0 min, 32 max allowed)
     24 hits, 0 fallbacks
     8 max cache size, 8 in cache
     30963 hits in cache, 0 misses in cache
Serial0 buffers, 1524 bytes (total 32, permanent 32):
     4 in free list (0 min, 32 max allowed)
     54 hits, 3 fallbacks
     8 max cache size, 7 in cache
     172593 hits in cache, 32 misses in cache
Serial1 buffers, 1524 bytes (total 32, permanent 32):
     7 in free list (0 min, 32 max allowed)
     25 hits, 0 fallbacks
     8 max cache size, 8 in cache
     0 hits in cache, 0 misses in cache

Router1#
```

Znaczenie poszczególnych pól zostanie wyjaśnione na przykładzie jednej z pul publicznych buforów:

```
Small buffers, 104 bytes (total 50, permanent 50):
     50 in free list (20 min, 150 max allowed)
     265016 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
```

Przytoczony fragment listingu dotyczy buforów Small, które są 104-bajtowymi obszarami pamięci. W danej chwili router dysponował pięćdziesięcioma takimi buforami. Wszystkie one są buforami typu `permanent`, co oznacza, że pamięć przez nie zajmowana nie jest zwracana do puli ogólnej pamięci urządzenia.

Dane zawarte w drugiej linii informują, że wszystkie z pięćdziesięciu wspomnianych buforów znajdują się na liście wolnych (ang. *free*), tzn. niezajętych w danej chwili. Wartości 20 i 150 odpowiadają parametrom `min-free` i `max-free`, których znaczenie zostało omówione wcześniej.

Wartość pola *hits* umieszczonego w trzeciej linii informuje o tym, ile razy router z powodzeniem zaalokował bufory z danej puli. Wartość pola *misses* określa liczbę zakończonych sukcesem prób alokacji buforów z puli, podczas których router musiał alokować także bufory dodatkowe. Wartość *trims* informuje o liczbie dynamicznie alokowanych buforów, które zostały natychmiast zwrócone. Wartość *created* zawiera informacje o liczbie buforów, które zostały utworzone jako bufory dodatkowe w wyniku niedopasowania uwzględnionego w polu *misses*.

Informacje o poważnych problemach w funkcjonowaniu buforów są zamieszczane w ostatniej linii bloku. Ewentualne zmiany parametrów buforów powinny być wykonywane jedynie na podstawie danych zamieszczonych w tejże linii. Pole *failures* informuje o liczbie przypadków, w których próba alokowania bufora zakończyła się niepowodzeniem, co z kolei spowodowało odrzucenie pakietu. Ostatnie z pól oznaczone jest jako *no memory* (brak pamięci). Wartość liczbowa w nim zawarta odpowiada liczbie przypadków wystąpienia błędu, który wynikał z braku pamięci niezbędnej do zaalokowania bufora. Problem ten zwykle ma bardzo poważne konsekwencje i może zostać usunięty przez zwiększenie pamięci routera.

Należy pamiętać, że w przypadku gdy próba zaalokowania bufora z jednej puli zakończy się niepowodzeniem, router zażąda utworzenia bufora z pamięci następnej większej puli. Zatem w sytuacji, kiedy nie będzie możliwe utworzenie bufora Big do obsługi 1500-bajtowego pakietu, zostanie wykorzystany bufor z puli VeryBig. Z tego względu można niekiedy zauważyć, że wartość *hits* puli VeryBig jest większa niż zero, mimo iż każdy interfejs routera ma ustawiony parametr MTU na poziomie 1500 bajtów. Dlatego też zaleca się tworzenie kilku stałych (`permanent`) buforów z puli większej niż największa wartość MTU.

Przyjrzyjmy się także statusowi buforów interfejsów:

```
Ethernet0 buffers, 1524 bytes (total 32, permanent 32):
    8 in free list (0 min, 32 max allowed)
    24 hits, 0 fallbacks
    8 max cache size, 8 in cache
    30963 hits in cache, 0 misses in cache
```

Powyższy listing zawiera wartości podobne do tych, które zostały omówione powyżej. Istnieją jednak także pewne różnice. Pierwsza z nich wiąże się z występowaniem pola *fallbacks*. W polu tym zapisywane są informacje o liczbie przypadków, kiedy w danym

interfejsie router musiał utworzyć dodatkowe bufory i musiał je pobrać z puli publicznej o odpowiednim rozmiarze. W prezentowanym przykładzie rozmiar bufora wynosi 1524 bajty, co oznacza, że dodatkowe bufory musiałyby być pobierane z puli Big.

Router utrzymuje pewną grupę buforów podręcznych niezależnie od tego, czy są w nich przechowywane jakiekolwiek dane czy też nie. Ich liczba jest różna w zależności od rodzaju urządzenia. Podobnie jak we wcześniej opisywanych przypadkach należy zwracać szczególną uwagę na wartość pola *misses*. Jeśli wartości pól *misses* i *fallbacks* są niskie, nie ma powodu do zmieniania ustawień buforów interfejsu.

Zmieniając parametry buforów, trzeba pamiętać o wcześniejszym sprawdzeniu ilości wolnej pamięci routera, do czego służy polecenie `show memory`:

```
Router1#show memory
                Head    Total(b)     Used(b)     Free(b)   Lowest(b)
Largest(b)
Processor     17DA4C    13112756     2308632    10804124    10577100
10663072
      I/O     E00000     2097152      336980     1760172     1740988
1759812
```

Zmieniając ustawienia buforów routera, trzeba obserwować zmiany zachodzące w obszarze dostępnej pamięci, gdyż ewentualne modyfikacje mogą wpływać zarówno na pamięć procesora, jak i układów wejścia-wyjścia. Przeznaczenie zbyt dużego obszaru pamięci na bufory może spowodować, że router nie będzie dysponował dostatecznie dużym jej obszarem, by mógł poprawnie funkcjonować w chwili wzrostu obciążenia.

2.4. Protokół wyszukiwania Cisco Discovery Protocol

Problem

Chcemy pozyskać informacje o tym, jakie urządzenia są przyłączone do poszczególnych interfejsów routera.

Rozwiązanie

Protokół wyszukiwania CDP (ang. *Cisco Discovery Protocol*) można włączać i wyłączać w obrębie poszczególnych interfejsów lub w odniesieniu do całego routera:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#cdp run
Router1(config)#interface Serial0/0
Router1(config-if)#cdp enable
Router1(config-if)#exit
Router1(config)#interface FastEthernet0/0
Router1(config-if)#no cdp enable
```

```
Router1(config-if)#exit
Router1(config)#interface FastEthernet1/0
Router1(config-if)#cdp enable
Router1(config-if)#end
Router1#
```

Analiza

Domyślnie protokół CDP jest włączony zarówno w routerze, jak i we wszystkich jego interfejsach. Jeżeli został wcześniej wyłączony (co zostało omówione w recepturze 2.5) i zachodzi potrzeba ponownego jego uruchomienia, należy wydać polecenie konfiguracyjne `cdp run`:

```
Router1(config)#cdp run
```

Domyślnie przetwarzanie danych protokołu CDP jest uruchomione we wszystkich interfejsach routera. W przypadku, gdy zachodzi konieczność wyłączenia go w ramach jednego interfejsu, trzeba posłużyć się poleceniem `no cdp enable`:

```
Router1(config)#interface Serial0/0
Router1(config-if)#no cdp enable
```

CDP jest protokołem wewnętrznym firmy Cisco i umożliwia identyfikowanie urządzeń Cisco w sieci oraz wymienianie danych identyfikacyjnych między poszczególnymi stacjami. Aby uzyskać informacje o najbliższych urządzeniach, które obsługują protokół CDP, wystarczy wprowadzić polecenie `show cdp neighbours`:

```
Router1#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
Router2          Ser 0/0            179          R        2621      Ser 0/1
Switch1          Fas 1/0            152         T S       WS-C2924  2/2
Router1#
```

Jak nietrudno zauważyć, wynikiem wykonania wspomnianego polecenia jest lista zawierająca informacje o nazwie, rodzaju oraz modelu wszystkich sąsiednich urządzeń. W jej skład wchodzą również dane o interfejsach routera (za pomocą których urządzenie komunikuje się z określonymi urządzeniami sąsiednimi) oraz informacje o urządzeniach sąsiednich przyłączonych do poszczególnych interfejsów.

W analizowanym przykładzie ostatni wpis oznacza przełącznik ethernetowy Cisco Catalyst. Największą zaletą omawianego rozwiązania jest właśnie to, że udostępnia ono także informacje o urządzeniach drugiej warstwy. Inne mechanizmy rozpoznawania urządzeń, takie jak protokół ARP, protokoły routingu czy nawet znane polecenie `ping`, odnoszą się jedynie do elementów trzeciej warstwy. W przypadku protokołu CDP można pozyskiwać informacje nawet o tych urządzeniach, które nie mają skonfigurowanych adresów IP.

Szczegółowe dane na temat innych jednostek sieciowych można uzyskać, dodając do polecenia opcję `detail`:

```
Router1#show cdp neighbors detail
-------------------------
Device ID: Router2
Entry address(es):
  IP address: 10.1.1.2
Platform: cisco 2621,  Capabilities: Router
Interface: Serial0/0,  Port ID (outgoing port): Serial0/1
Holdtime : 136 sec

Version :
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-IK9O3S-M), Version 12.2(13), RELEASE SOFTWARE
(fc1)
Copyright (c) 1986-2002 by cisco Systems, Inc.
Compiled Tue 19-Nov-02 22:27 by pwade

advertisement version: 2

Device ID: Switch1
Entry address(es):
  IP address: 172.25.1.4
Platform: WS-C2924,  Capabilities: Trans-Bridge Switch
Interface: FastEthernet1/0,  Port ID (outgoing port): FastEthernet0/12
Holdtime : 116 sec

Version :
Cisco Internetwork Operating System Software
IOS (tm) C2900XL Software (C2900XL-C3H2S-M), Version 12.0(5)WC3b,
RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2002 by cisco Systems, Inc.
Compiled Fri 15-Feb-02 10:14 by antonino

advertisement version: 2
Duplex: full

Router1#
```

Na podstawie powyższego zestawienia można określić adresy IP urządzeń sąsiednich oraz ustalić wersje systemów Cisco IOS lub CatOS.

Obydwie jednostki obsługują protokół CDP w wersji drugiej, który został wprowadzony przez firmę Cisco w systemie IOS 12.0(3)T. Zawarto w nim trzy nowe pola, które okazały się bardzo użyteczne w zastosowaniach w sieciach LAN. Są nimi: nazwa domenowa VTP (ang. *VTP Domain Name*), podstawowa sieć VLAN 802.1Q (*802.1Q native VLAN*) oraz konfiguracja pracy dupleksowej. Jak nietrudno zauważyć, analizując przedstawiony wydruk, router i przełącznik uzgodniły, że dane będą wymieniane w trybie pełnego dupleksu. Konfiguracja pracy dupleksowej oraz standard 802.1Q zostały szczegółowo omówione w rozdziale 16.

Nowo wprowadzona opcja informacji o pracy dupleksowej okazała się niezwykle użyteczna, gdyż na jej podstawie router i przełącznik mogą automatycznie rozpoznać możliwe tryby komunikowania się. W kolejnym przykładzie zaprezentowano sposób postępowania routera w przypadku, gdy wystąpi różnica między jego trybem pracy, a trybem pracy przełącznika. W przełączniku ustawiono komunikację półdupleksową. Dzięki zastosowaniu protokołu CDP router wykrył różnicę i umieścił w dzienniku pracy następujący komunikat:

```
Feb  6 11:36:11: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on
FastEthernet1/0 (not half duplex), with 003541987 (switch) FastEthernet0/12
(half duplex).
```

Protokół CDP w wersji drugiej jest domyślnie obsługiwany we wszystkich systemach IOS od wersji 12.0(3)T włącznie. Aby wyłączyć w routerze obsługę drugiej wersji protokołu CDP i uruchomić obsługę wersji pierwszej, należy zastosować następujące polecenie konfiguracyjne:

```
Router1(config)#no cdp advertise-v2
```

Trudno powiedzieć, dlaczego producent uwzględnił możliwość zmiany wersji protokołu, ponieważ nie odnotowano żadnych problemów we współdziałaniu urządzeń obsługujących protokół CDP w wersji pierwszej i drugiej. Istnieją co prawda pewne problemy związane z zabezpieczeniami (które zostaną omówione w recepturze 2.5), ale najlepszym sposobem ich rozwiązania jest całkowite wyłącznie obsługi CDP.

Ogólne ustawienia protokołu CDP w routerze można przeanalizować po wydaniu polecenia `show cdp`:

```
Router1#show cdp
Global CDP information:
        Sending CDP packets every 60 seconds
        Sending a holdtime value of 180 seconds
        Sending CDPv2 advertisements is  enabled
Router1#
```

Na podstawie przedstawionego wydruku można powiedzieć, że router wysyła pakiety informujące o jego obecności w sieci co 60 sekund, co jest wartością domyślną. Parametr *holdtime* definiuje czas, przez jaki router będzie oczekiwał na kolejne ogłoszenie CDP od jednego z sąsiednich urządzeń. Jeżeli w określonym czasie ogłoszenie nie zostanie odebrane, router usuwa informację o danym urządzeniu ze swojej tablicy jednostek sąsiednich.

Aby nadać nowe wartości tym ustawieniom (zmiana ma charakter globalny i odnosi się do całego routera), należy zastosować poniższe polecenia:

```
Router(config)#cdp timer 30
Router(config)#cdp holdtime 240
```

Przy definiowaniu wartości obydwu parametrów wykorzystywaną jednostką jest sekunda. Ustawienie czasu rozsyłania ogłoszeń (`timer`) może przyjmować wartości z przedziału od 5 do 254 sekund. Wartość parametru `holdtimer` musi zawierać się w przedziale od 10 do 255 sekund.

Zobacz również

Receptura 2.5, rozdział 16.

2.5. Wyłączanie obsługi protokołu CDP

Problem

Nie chcemy pozwolić na to, żeby sąsiednie urządzenia pozyskiwały informacje o routerze. Powodem takich działań są względy bezpieczeństwa.

Rozwiązanie

Aby wyłączyć obsługę protokołu CDP w jednym z interfejsów, należy zastosować polecenie `no cdp enable`:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#cdp run
Router1(config)#interface FastEthernet0/0
Router1(config-if)#no cdp enable
Router1(config-if)#end
Router1#
```

Z kolei całkowite wyłączenie obsługi protokołu CDP w routerze wymaga podania instrukcji `no cdp run`:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#no cdp run
Router1(config)#end
Router1#
```

Analiza

Protokół CDP może być niezwykle użyteczny, gdyż pozwala pozyskiwać wiele informacji na temat sąsiednich urządzeń sieciowych. Jednocześnie cecha ta może być potencjalną wadą rozwiązania ze względu na problemy z zachowaniem odpowiedniego poziomu zabezpieczeń. Pakiety CDP nie podlegają szyfrowaniu, dlatego mogą być przechwytywane przez inne urządzenia funkcjonujące w sieci. Odtworzenie struktury sieci na podstawie pozyskanych w ten sposób danych nie jest szczególnym problemem. Gdyby komuś udało się uzyskać dostęp do routera za pomocą usług Telnet lub SNMP, mógłby on wykorzystać tablice CDP do określenia topologii całej sieci, rozwiązań warstwy drugiej i trzeciej, a także do ustalenia wersji IOS, modeli i typów routerów oraz przełączników, a także schematu adresowania IP. Posiadanie tego typu informacji daje możliwość przeprowadzenia bardzo efektywnego ataku na daną sieć.

Z tego powodu wielu administratorów decyduje się na wyłączenie obsługi protokołu CDP. Jeżeli więc wyłącza się mechanizm CDP ze względów bezpieczeństwa, prawdopodobnie najlepszym rozwiązaniem będzie wyłączenie go w całym routerze, a nie na poszczególnych interfejsach. Wyłączenie obsługi protokołu na pojedynczym interfejsie zabezpiecza system

jedynie przed podsłuchem informacji rozgłaszanych za pomocą protokołu CDP. Jednak nadal możliwy jest dostęp do tablicy CDP przy wykorzystaniu usług takich jak Telnet i SNMP. Informacje o sieci są więc nadal zagrożone.

Trzeba wyjaśnić, że zagrożenie bezpieczeństwa systemu wynika z możliwości przeprowadzenia celowego i dokładnie zaplanowanego ataku na daną sieć zarówno z jej wnętrza, jak i z sieci zewnętrznej przyłączonej do sieci atakowanej. Z tego względu stanowczo zaleca się wyłączanie obsługi protokołu CDP we wszystkich routerach, które mają połączenia z sieciami zewnętrznymi, a szczególnie z internetem. Z kolei wyłączanie omawianej opcji w sieciach wewnętrznych zabezpiecza system przed działaniami osób bezpośrednio przyłączonych do sieci. Należy więc rozważyć korzyści wynikające z zastosowania protokołu CDP i ryzyko ewentualnego ataku ze strony osób, które mają uprawnienia do korzystania z sieci. To, czy opcja CDP zostanie wyłączona czy nie, zależy jedynie od zaufania do użytkowników sieci.

Zobacz również

Receptura 2.4.

2.6. Wykorzystanie „małych serwerów"

Problem

Chcemy włączać i wyłączać takie usługi routera jak *finger*, *echo* i *chargen*.

Rozwiązanie

Aplikacja *finger* umożliwia zdalne sprawdzenie, kto jest zalogowany w routerze. Aby ją włączyć, należy użyć polecenia `ip finger`:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#ip finger
Router1#
```

Każdy z routerów Cisco posiada pewien zestaw nieskomplikowanych aplikacji serwerowych protokołów TCP i UDP, które często przydają się podczas prowadzenia różnego rodzaju testów:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#service tcp-small-servers
Router1(config)#service udp-small-servers
Router1(config)#end
Router1#
```

Analiza

Program *finger* jest narzędziem, które stanowi odpowiednik polecenia `show users`, ale które można stosować w odniesieniu do routerów zdalnych. Uruchomienie programu *finger* w systemach Unix sprowadza się zazwyczaj do wykorzystania następującego polecenia:

```
Freebsd% finger @Router1
[Router1]

    Line       User       Host(s)              Idle       Location
    66 vty 0   kdooley    idle                 00:22:47   freebsd
    67 vty 1   ijbrown    idle                 1d07h      freebsd
*   68 vty 2              idle                 00:00:00   freebsd

    Interface  User        Mode                Idle       Peer Address

Freebsd%
```

Dostęp do serwera *finger* jest również możliwy za pomocą programu klienta Telnetu przy zestawieniu połączenia z portem 79. Wykonanie tego typu operacji z innego routera wymaga wprowadzenia następującego polecenia:

```
Router2#telnet 10.1.1.2 finger
Trying 10.1.1.2, 79 ... Open

    Line       User       Host(s)              Idle       Location
    66 vty 0   kdooley    idle                 00:24:14   freebsd
    67 vty 1   ijbrown    idle                 1d07h      freebsd
*   67 vty 1              idle                 00:00:00   10.2.2.2

  Interface    User       Mode                     Idle   Peer Address

[Connection to 10.1.1.2 closed by foreign host]
Router2#
```

Warto zauważyć, że w obydwu przypadkach na liście znajdują się nie tylko użytkownicy routera, ale również sam proces programu *finger*, który oznaczono symbolem gwiazdki.

Protokół *finger* został zdefiniowany w standardzie RFC 1288. Domyślnie jego obsługa jest w routerach wyłączana. Mimo iż omawiane rozwiązanie jest bardzo wygodnym w użyciu sposobem sprawdzenia, kto korzysta ze zdalnego routera (bez potrzeby osobistego logowania się w routerze), stanowi znaczne zagrożenie dla systemu zabezpieczeń urządzenia. Nie dość, że udostępnia informacje o identyfikatorach użytkowników, zajmuje jedną z linii VTY, która przy stałym wykorzystaniu uniemożliwia korzystanie z urządzenia osobom, które mają do tego prawo. Protokół *finger* ma też swoją niechlubną przeszłość, gdyż jeden z pierwszych ataków z zastosowaniem wirusów (słynnego Morris Worm), który spowodował wyłączenie znacznej części urządzeń w internecie był atak wykorzystujący błąd w pierwotnej implementacji usługi *finger*.

Biorąc ten fakt pod uwagę, zaleca się wyłączenie protokołu *finger* we wszystkich konfigurowanych routerach. Jeżeli z jakichkolwiek przyczyn jest on w danej chwili włączony, można go w następujący sposób wyłączyć:

```
Router1(config)#no ip finger
```

Polecenie `ip finger` zastępuje polecenie `service finger`, które można jeszcze znaleźć w wielu materiałach źródłowych:

```
Router1(config)#service finger
```

Jeżeli korzysta się ze starszej wersji polecenia, router automatycznie zastąpi ją nowszą wersją instrukcji.

Routery Cisco zawierają również pewien zbiór aplikacji TCP i UDP, które są często spotykane w urządzeniach wykorzystujących protokół IP. W systemach IOS 12.0 i wersjach późniejszych „małe serwery” TCP i UDP są domyślnie wyłączone. We wcześniejszych wersjach IOS są one włączone.

Zastosowanie wspomnianych serwerów jest marginalne i zaleca się ich wyłączanie, o ile nie są one wykorzystywane w procedurach testowych. Ich zadanie polega na nasłuchiwaniu pakietów przychodzących z dowolnych źródeł, co czyni te rozwiązania podatnymi na ataki typu DoS. W atakach DoS zakłada się zazwyczaj, że serwer TCP zaakceptuje połączenie z dowolnej stacji, która takiego połączenia zażąda. Wysłanie przez któregokolwiek z użytkowników strumienia pakietów TCP SYN na jeden z portów sprawia, że router musi na nie odpowiedzieć, przeznaczając na ten cel wewnętrzne zasoby systemowe. Taka sytuacja może doprowadzić do wyczerpania zasobów systemowych routera.

Serwery UDP stanowią potencjalne zagrożenie z tego względu, że użytkownicy sieci mogą podmieniać własne adresy (ang. *spoofing*), zmuszając router do odsyłania pakietów do komputerów osób trzecich. Podobny atak można przeprowadzić z wykorzystaniem serwerów TCP. Router odpowiada na każdy pakiet TCP SYN pakietem SYN ACK. Inne urządzenie sieciowe może nie poradzić sobie z obsługą niespodziewanych pakietów SYN ACK.

Z tego względu zaleca się wyłączanie wspomnianych usług, chyba że ich działanie jest konieczne.

```
Router1(config)#no service tcp-small-servers
Router1(config)#no service udp-small-servers
```

Jednak w takim przypadku serwery powinny uwierzytelniać użytkowników.

W tabeli 2.2 zestawiono dostępne serwery TCP i UDP. Każdy z serwerów został zaimplementowany w routerze na tych samych portach zarówno w wersji TCP, jak i UDP. Numery portów są powszechnie znane i implementowane w wielu aplikacjach sieciowych. Zastosowanie wymienionych serwerów ogranicza się głównie do procedur testowych.

Najłatwiejszym sposobem sprawdzenia, do czego służą poszczególne usługi, jest ich przetestowanie. Działanie serwerów łatwiej jest zademonstrować przy wykorzystaniu protokołu TCP, gdyż można do tego celu wykorzystać standardową aplikację Telnet. Cała procedura sprowadza się wówczas do poprawnego określenia portu TCP.

Tabela 2.2. Małe serwery TCP i UDP

Numer portu	Nazwa	Dokument RFC	Opis
7	`echo`	RFC 862	Serwer odsyła do klienta pakiet, który od niego otrzymał.
9	`discard`	RFC 863	Serwer odrzuca wszystkie dane pochodzące do danego klienta.
13	`daytime`	RFC 867	Serwer odsyła wartości aktualnej daty i czasu, po czym kończy sesję.
19	`chargen`	RFC 864	Serwer przesyła do klienta stały strumień znaków ASCII.

Funkcja `echo` powoduje, że serwer odsyła do klienta te same dane, które od niego otrzymał:

```
Freebsd% telnet Router1 echo
Trying 172.25.25.1...
Connected to Router1.
Escape character is '^]'.
Przykladowe zdanie testujace usluge echo.
Przykladowe zdanie testujace usluge echo.
^]
telnet> quit
Connection closed.
Freebsd%
```

W wersji wykorzystującej protokół UDP funkcja *echo* kopiuje otrzymany segment danych i odsyła go do nadawcy.

Funkcja *discard* jest znacznie mniej użyteczna. Umożliwia bowiem klientowi utworzenie sesji TCP z serwerem, po czym ignoruje wszystkie otrzymane dane:

```
Freebsd% telnet Router1 discard
Trying 172.25.25.1...
Connected to Router1.
Escape character is '^]'.
Przykladowe zdanie testujace usluge discard.
^]
telnet> quit
Connection closed.
Freebsd%
```

Wersja UDP aplikacji oczekuje na pakiety UDP na porcie 9, ale nie odpowiada na nie w żaden sposób.

Serwer *daytime* w wersji TCP akceptuje żądania utworzenia połączenia, wysyła pakiet zawierający informację o dacie i czasie (w formacie ASCII), po czym kończy sesję:

```
Freebsd% telnet Router1 daytime
Trying 172.25.25.1...
Connected to Router1.
Escape character is '^]'.
Sunday, January 5, 2003 17:41:21-EST
Connection closed by foreign host.
Freebsd%
```

Wersja UDP serwera *daytime* nasłuchuje na porcie 13. i odpowiada na żądania pojedynczym pakietem zawierającym te same dane, które są generowane przez wersję TCP usługi. Wykorzystanie serwera *daytime* do pozyskiwania informacji o czasie jest znikome. Znacznie efektywniejsze w tym względzie są inne aplikacje, np. NTP, które dostarczają bardziej aktualnych danych. Usługi NTP zostały omówione w rozdziale 14.

Funkcja generowania znaków (*chargen*) wydaje się być najbardziej użyteczną usługą małych serwerów TCP. Po ustanowieniu połączenia z określonym portem, router rozpoczyna przesyłanie do klienta strumienia danych. Często takie rozwiązanie jest wykorzystywane w charakterze generatora ruchu dla ubogich, umożliwiając badanie obciążenia sieci:

```
Freebsd% telnet Router1 chargen
Trying 172.25.25.1...
Connected to Router1.
Escape character is '^]'.
!"#$%&'(  )*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg
!"#$%&'(  )*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefgh
"#$%&'(  )*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghi
#$%&'(  )*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghij
$%&'(  )*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijk
%&'(  )*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijkl
<kolejne linie zostały usunięte>
^]
telnet> quit
Connection closed.
Freebsd%
```

Wersja UDP serwera *chargen* nasłuchuje pakietów UDP na porcie 19. i generuje pojedynczy pakiet odpowiedzi, który może się składać z przypadkowej liczby znaków dobieranej z zakresu od 0 do 512.

Zobacz również

Rozdział 14., dokumenty RFC 1288, RFC 862, RFC 863, RFC 864 oraz RFC 867, „Wieczór trzech króli" Williama Szekspira.

2.7. Dostęp do routera z wykorzystaniem protokołu HTTP

Problem

Chcemy konfigurować router oraz monitorować jego pracę za pomocą przeglądarki internetowej.

Rozwiązanie

System IOS został wyposażony przez firmę Cisco w serwer HTTP. Włączenie usługi HTTP routera umożliwia komunikowanie się z nim za pomocą standardowej przeglądarki internetowej lub klienta Telnetu:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#access-list 75 permit 172.25.1.1
Router1(config)#access-list 75 deny any
Router1(config)#ip http server
Router1(config)#ip http access-class 75
Router1(config)#end
Router1#
```

Analiza

Po skonfigurowaniu wspomnianej opcji w routerze można się połączyć z urządzeniem za pomocą standardowej przeglądarki internetowej. W poniższym przykładzie wykorzystano do tego celu tekstową przeglądarkę Lynx, która prezentuje stronę routera w następujący sposób:

```
                                                        Router1 Home Page

                                    Cisco Systems

Accessing Cisco 2621 "Router1"

        Telnet - to the router.

        Show interfaces - display the status of the interfaces.
        Show diagnostic log - display the diagnostic log.
        Monitor the router - HTML access to the command line interface at
                level 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15

        Connectivity test - ping the nameserver.

        Show tech-support - display information commonly needed by tech
                support.

        QoS Device Manager - Configure and monitor QoS through the web
                interface.
    ------------------------------------------------------------------------

  Help resources

    1. CCO at www.cisco.com - Cisco Connection Online, including the
       Technical Assistance Center (TAC).
    2. tac@cisco.com - e-mail the TAC.
    3. 1-800-553-2447 or +1-408-526-7209 - phone the TAC.
    4. cs-html@cisco.com - e-mail the HTML interface development group.
```

Słowa wyróżnione pogrubionym drukiem są odsyłaczami, które umożliwiają uruchomianie poleceń EXEC systemu IOS. Przykładowo, odsyłacz *Show interfaces* uruchamia polecenie `show interfaces` i powoduje wyświetlenie wyniku w postaci strony WWW.

Możliwe jest również konfigurowanie routera za pomocą przeglądarki. Po wybraniu jednej z opcji określającej poziom dostępu, użytkownik otrzymuje możliwość korzystania z poleceń EXEC właściwych dla danego poziomu autoryzacji. Więcej informacji o poziomach uwierzytelniania znajduje się w rozdziale 3.

Dostęp do routera za pomocą protokołu HTTP został wprowadzony w IOS w wersji 11.2. Trzeba jednak zaznaczyć, że w pierwszych wersjach rozwiązania występował poważny błąd, który został usunięty dopiero w IOS 12.1(5). Błąd ten powodował zatrzymywanie pracy routera w przypadku, gdy użytkownik popełnił relatywnie niewielki błąd literowy. Wprowadzenia znaku zapytania jako części polecenia wykonywanego za pomocą usługi Telnet powoduje, że router udostępnia informacje na temat wszystkich opcji danego polecenia. Jednak umieszczenie znaku zapytania w adresie URL skutkowało zatrzymaniem pracy routera. W ten sposób nawet osoby, które miały prawo korzystania z usługi, mogły popełnić błąd, który skutkował poważnymi konsekwencjami. Dlatego nie zaleca się korzystania z omawianego rozwiązania w systemach IOS wcześniejszych niż 12.1(5).

W najnowszych wersjach systemu IOS interfejs WWW nie jest ani bardziej, ani mniej bezpieczny niż dostęp do poleceń EXEC z wiersza poleceń aplikacji Telnet. W obydwu przypadkach trzeba podawać odpowiednie informacje uwierzytelniające. Poszczególne sposoby uwierzytelniania, takie jak wykorzystywana w usługach Telnet metoda AAA, omówiono w rozdziałach 3. i 4. Wszystkie wymieniane tam sposoby obowiązują również w rozwiązaniach bazujących na protokole HTTP, a do ich konfiguracji służy słowo kluczowe `authentication`. Aby skonfigurować serwer HTTP tak, żeby korzystał z uwierzytelniania typu AAA, należy wprowadzić następujące polecenie:

```
Router1(config)#ip http authentication aaa
```

Możliwe jest również wyznaczenie urządzeń, które mają prawo korzystania z interfejsu WWW routera. Służy do tego słowo kluczowe `access-list`. W poniższym przykładzie przekazano do routera informację o tym, że dostęp do serwera HTTP definiuje lista dostępowa nr 75, która z kolei umożliwia komunikowanie się z usługą tylko jednej stacji o określonym adresie IP:

```
Router1(config)#access-list 75 permit 172.25.1.1
Router1(config)#access-list 75 deny any
Router1(config)#ip http access-class 75
```

Korzystanie z wiersza poleceń interfejsu Telnet wydaje się jednak łatwiejsze od posługiwania się interfejsem WWW. Jedynym argumentem przemawiającym za używaniem protokołu HTTP jest możliwość udostępnienia podstawowych poleceń (np. `show interfaces`) technikom pierwszego poziomu.

Zobacz również

Rozdział 3.

2.8. Korzystanie ze statycznych tablic nazw stacji

Problem

Chcemy utworzyć w routerze statyczną tablicę nazw stacji.

Rozwiązanie

Umieszczanie wpisów w statycznej tablicy nazw stacji jest możliwe dzięki poleceniu `ip host`:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#ip host freebsd 172.25.1.1
Router1(config)#ip host router2 10.1.1.1 172.22.1.4
Router1(config)#end
Router1#
```

Analiza

W wielu poleceniach routera można zastąpić adresy IP nazwami stacji. Plik konfiguracyjny staje się wówczas bardziej przejrzysty, gdyż zamiast enigmatycznych adresów IP zawiera łatwiejsze do przyswojenia nazwy urządzeń. Router musi mieć jednak możliwość odwzorowania nazw na adresy IP. W urządzeniach Cisco zaimplementowano dwa mechanizmy zmiany nazw na adresy. Pierwszy z nich wykorzystuje omawiane w niniejszej recepturze statyczne tablice nazw, natomiast drugi bazuje na serwerach DNS i jest omówiony w recepturze 2.9.

Statyczne wpisy są dostępne jedynie lokalnie. Router nie dzieli się informacjami zawartym w tablicy z innymi routerami czy innymi urządzeniami pracującymi w sieci. W przeciwieństwie do odwzorowania wykonywanego w systemie DNS procedura pozyskiwania informacji o stacji nie zależy od usług zewnętrznych, takich jak serwery nazw. Jeżeli w danym urządzeniu wykorzystuje się zarówno statyczne tablice nazw, jak i system DNS, router będzie przede wszystkim posługiwał się tablicą statyczną, co daje możliwość przesłonięcia danych udostępnianych normalnie przez serwery DNS.

Największa niedogodność wynikająca ze stosowania statycznych tablic nazwa jest związana z tym, że odwzorowanie ma charakter statyczny. Oznacza to, że jakakolwiek zmiana adresów IP wymaga ingerencji w treść tablic. Z kolei do największych zalet rozwiązania należy zaliczyć jego niezależność od zewnętrznych serwerów. Jeżeli jakakolwiek z istotnych funkcji routera zostanie zdefiniowana z wykorzystaniem nazwy stacji zamiast jej adresu IP, w przypadku chwilowej niedostępności serwera DNS komunikacja może się okazać niemożliwa.

Z tego powodu zaleca się stosowanie statycznych tablic nazw zamiast systemu DNS, o ile przy konfiguracji routera zastępuje się adresy IP nazwami urządzeń.

W prezentowanym przykładzie zawarto informację o tym, że stacja o nazwie *router2* posiada dwa adresy IP:

```
Router1(config)#ip host router2 10.1.1.1 172.22.1.4
```

Jeżeli dla jednego urządzenia zostanie określona większa liczba adresów IP, router będzie wykorzystywał każdy z nich w kolejności, w jakiej występują w poleceniach konfiguracyjnych. Tworząc wpis dla sąsiedniego routera, warto rozpocząć definiowanie adresów IP od adresu interfejsu przyłączonego do tego samego segmentu sieci, a po nim umieścić pozostałe osiągalne adresy IP.

Polecenie `ip host` pozwala również na definiowanie numerów portów, a konkretnie na definiowanie portów TCP, które będą wykorzystywane do zestawiania połączenia telnetowego ze stacją o określonej nazwie. Domyślnie polecenie `telnet` tworzy połączenie w protokole TCP z portem o numerze 23. W poniższym przykładzie nazwą stacji jest `mail`, a router otrzymał informację, żeby łączyć się z portem `25` (SMTP) urządzenia:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#ip host mail 25 172.25.1.1
Router1(config)#end
Router1#
```

Uruchomienie narzędzia *Telnet* z podaniem nazwy `mail` powoduje zestawienie połączenia z usługą SMTP:

```
Router1#telnet mail
Trying mail (172.25.1.1, 25)... Open
220 freebsd.oreilly.com ESMTP Postfix
quit
221 Bye

[Connection to mail closed by foreign host]
Router1#
```

Router łączy się bezpośrednio z portem serwera dostarczania poczty — portem 25. Aby przesłonić zdefiniowany w tablicy numer portu, można w wierszu poleceń narzędzia *Telnet* dołączyć drugi parametr odpowiadający wybranemu portowi:

```
Router1#telnet mail 25
```

Wyświetlenie pełnej listy zdefiniowanych statycznych wpisów stacji wymaga wprowadzania polecenia `show hosts`:

```
Router1#show hosts
Default domain is not set
Name/address lookup uses static mappings

Host                      Port  Flags      Age Type   Address(es)
freebsd                   None  (perm, OK)  0   IP    172.25.1.1
router2                   None  (perm, OK)  0   IP    10.1.1.1
                                                      172.22.1.4
mail                      25    (perm, OK)  0   IP    172.25.1.1
Router1#
```

Zobacz również

Receptura 2.9.

2.9. Korzystanie z systemu nazw domenowych

Problem

Chcemy tak skonfigurować router, żeby do odwzorowywania nazw stacji wykorzystywał system DNS.

Rozwiązanie

Aby włączyć w routerze opcję korzystania z serwerów DNS przy poszukiwaniu adresów IP dla nazw stacji, trzeba określić jego nazwę domenową oraz co najmniej jeden serwer nazw:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#ip domain-lookup
Router1(config)#ip domain-name helion.pl
Router1(config)#ip name-server 172.25.1.1
Router1(config)#ip name-server 10.1.20.5
Router1(config)#end
Router1#
```

Analiza

Jak już wspomniano w recepturze 2.8, do odwzorowywania nazw na adresy IP można wykorzystywać system DNS. W praktyce routery Cisco mają tę opcję domyślnie włączoną. Jednak ze względu na fakt, że w swoich plikach konfiguracyjnych nie zawierają informacji o domyślnym serwerze nazw, starają się do tego celu wykorzystywać adres rozgłoszeniowy (255.255.255.255). Oznacza to, że wymienione polecenie `ip domain-lookup` jest niezbędne tylko w przypadku, gdy z jakichkolwiek przyczyn opcja korzystania z systemu DNS została w routerze wyłączona.

Po zapisaniu w ustawieniach routera poprawnej nazwy serwera DNS można korzystać z nazw domenowych dowolnych stacji, których dane znajdują się w określonym serwerze DNS. Serwer wykorzystywany w prezentowanym poniżej przykładzie ma możliwość wymiany informacji z serwerami w internecie, dlatego umożliwia wykorzystanie polecenia `ping` w odniesieniu do serwera WWW firmy Cisco:

```
Router1#ping www.cisco.com
Translating "www.cisco.com"...domain server (172.25.1.1) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 198.133.219.25, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 80/91/104 ms
Router1#
```

Z analizy listingu wynika, że router przesłał zapytanie DNS do serwera nazw o adresie `172.25.1.1`, żądając w nim zamiany nazwy *www.cisco.com* na odpowiadający jej adres IP. Odpowiedź serwera zawierała adres `198.133.219.25`. W kolejnym kroku router wykorzystał polecenie `ping` w taki sposób, jakby zamiast nazwy domenowej parametrem był adres IP.

W prezentowanym wcześniej przykładzie zdefiniowano dwa serwery nazw:

```
Router1(config)#ip name-server 172.25.1.1
Router1(config)#ip name-server 10.1.20.5
```

Router przesyła zapytania do poszczególnych serwerów DNS w kolejności, w jakiej dokonano wpisów. Załóżmy, że wykorzystujemy polecenie `ping` w odniesieniu do nieistniejącej stacji *receptury.helion.pl*:

```
Router1#ping receptury.helion.pl
Translating "receptury.helion.pl"...domain server (172.25.1.1)(10.1.20.5)
% Unrecognized host or address, or protocol not running.

Router1#
```

Z analizy odpowiedzi wynika, że w pierwszej kolejności router przesłał zapytanie do serwera nazw o adresie `172.25.1.1`. Gdy serwer ten uznał, że nie może dokonać odwzorowania nazwy na adres IP, router skierował zapytanie do drugiego serwera nazw (o adresie `10.1.20.5`). W rezultacie procedura zakończyła się niepowodzeniem, gdyż stacja o podanej nazwie nie istnieje.

Chcąc się zapoznać z ustawieniami systemu DNS, należy wydać polecenie `show hosts`:

```
Router1#show hosts
Default domain is helion.pl
Name/address lookup uses domain service
Name servers are 172.25.1.1, 10.1.20.5

Host                      Port  Flags      Age Type   Address(es)
www.cisco.com             None  (temp, OK) 0   IP     198.133.219.25
Router1#
```

Wykonanie instrukcji powoduje wyświetlenie nazwy domeny, nazw serwerów DNS (w określonej kolejności) oraz informacji o ostatnio dokonywanych odwzorowaniach. Router przechowuje wyniki ostatnich zapytań w pamięci podręcznej, gdyż dzięki temu eliminuje się konieczność kierowania zapytań do serwerów DNS przy kolejnych próbach komunikacji z tym samym urządzeniem. Różnica między omawianym sposobem budowania tablicy nazw, a metodą statycznego jej tworzenia (która została opisana w recepturze 2.8) polega na tym, że wpisy dynamiczne są po upływie określonego czasu automatycznie usuwane. Czas ten jest określany przez serwer DNS oddzielnie dla każdej nazwy. Nie można go w routerze zmienić.

Polecenie `ip domain-name` umożliwia określenie domeny, do której należy dana sieć:

```
Router1(config)#ip domain-name helion.pl
```

Dzięki zdefiniowaniu nazwy domeny, zamiast wprowadzania pełnej nazwy domenowej stacji (FQDN — ang. *Fully Qualified Domain Name*), można się posługiwać samymi nazwami właściwymi stacji. Można zatem użyć nazwy *poczta* zamiast *poczta.helion.pl*, a mimo to odwzorowanie zostanie wykonane poprawnie.

W niektórych organizacjach wykorzystuje się więcej niż jedną domenę. W takim przypadku podczas konfigurowania routera należy użyć kilku poleceń `ip domain-list`, które pozwolą na zdefiniowanie kilku nazw domen. Dzięki poniższym poleceniom można umieścić w routerze informację o przynależności do dwóch domen:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#ip domain-list helion.pl
Router1(config)#ip domain-list helion2.pl
Router1(config)#end
Router1#
```

Jeżeli nie została określona lista domen, ale zdefiniowano nazwę domeny, router będzie korzystał z nazwy domeny. Jednak w przypadku wprowadzenia zarówno nazwy domeny, jak i listy domen urządzenie zignoruje parametr nazwy domeny. Dlatego podczas deklarowania listy domen konieczne było powtórzenie wcześniej wprowadzonej nazwy domeny.

Kolejność umieszczania wpisów na liście domenowej nie jest obojętna, gdyż przy budowaniu zapytań z wykorzystaniem pełnej nazwy FQDN router będzie korzystał ze zdefiniowanej listy. Jeżeli zatem zostanie wysłane zapytanie o stację o nazwie *poczta*, router ustali prawidłową domenę niezależnie od tego, w której z nich znajduje się stacja *poczta*. Jednak w przypadku, gdy stacja znajduje się w obydwu domenach, połączenie zostanie zestawione z jednostką *poczta.helion.pl*, a nie z *poczta.helion2.pl* — zgodnie z kolejnością występowania domen na liście. Oczywiście nic nie stoi na przeszkodzie, żeby połączyć się ze stacją o adresie *poczta.helion2.pl*, ale konieczne będzie wprowadzanie pełnej nazwy domenowej.

Polecenie `show hosts` wyświetla również informacje o listach domen:

```
Router1#show hosts
Default domain is helion.pl
Domain list: helion.pl, helion2.pl
Name/address lookup uses domain service
Name servers are 172.25.1.1, 172.25.1.3, 10.1.20.5

Host                      Port  Flags      Age Type   Address(es)
www.cisco.com             None  (temp, OK) 0   IP    198.133.219.25
freebsd                   None  (perm, OK) 0   IP    172.25.1.1
Router1#
```

Zobacz również

Receptura 2.8.

2.10. Wyłączanie odwzorowania nazw domenowych

Problem

Chcemy nie dopuścić do tego, żeby router próbował utworzyć połączenie ze stacją o nazwie, która w rzeczywistości nie jest adresem, ale błędem literowym powstałym podczas wpisywania polecenia.

Rozwiązanie

Aby uniemożliwić routerowi próby odwzorowania błędnych wartości tekstowych, należy zastosować polecenie `ip domain-lookup`:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#no ip domain-lookup
Router1(config)#end
Router1#
```

Możliwe jest również uniemożliwienie odwzorowywania błędnych wartości w routerach, które korzystają z systemu DNS. Trzeba w tym celu zmienić domyślny sposób działania polecenia EXEC, wykorzystywanego w odniesieniu do nieznanych poleceń:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#line vty 0 4
Router1(config-line)#transport preferred none
Router1(config-line)#end
Router1#
```

Analiza

Zgodnie z informacjami przedstawionymi w recepturze 2.9, routery domyślnie starają się odwzorowywać wszystkie nazwy stacji za pomocą serwerów DNS. Jeżeli adres serwera DNS nie zostanie umieszczony w plikach konfiguracyjnych urządzenia, router będzie korzystał z adresu IP rozgłoszenia lokalnego, czyli 255.255.255.255. Kierowanie zapytań do nieistniejącego serwera jest nie tylko bezcelowe, ale również czasochłonne. W przypadku wystąpienia takiego problemu w czasie sesji interaktywnej, router nie udostępni zgłoszenia EXEC, dopóki nie upłynie czas przeznaczony na realizację zapytania. Domyślne ustawienia pracy routera powodują, że każde nieznane polecenie interpretuje on jako nazwę stacji, z którą użytkownik chce zestawić połączenie. W analogiczny sposób traktowane są wszystkie ewentualne błędy literowe powstałe podczas pisania poleceń:

```
Router1#pnig
Translating "pnig"...domain server (255.255.255.255)

Translating "pnig"...domain server (255.255.255.255)
 (255.255.255.255)
Translating "pnig"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer address
Router1#
```

W prezentowanym przykładzie błędnie wpisano polecenie `ping`. Router oczywiście nie zna takiego polecenia, zakłada więc, że podano nazwę stacji i stara się ją odwzorować na adres IP. Każdy, kto pracował z routerami Cisco dłużej niż kilka minut, doskonale zna ten problem — irytacja powodowana błędnym wpisaniem polecenia jest potęgowana przez konieczność czekania kilku sekund na upłynięcie dopuszczalnego czasu realizacji żądania.

Jednym z łatwiejszych sposób wyeliminowania omawianego problemu jest wyłączenie korzystania z systemu DNS, co zostało uczynione w pierwszym z przykładów:

```
Router1(config)#no ip domain-lookup
```

Proponowane rozwiązanie jest właściwe tylko w przypadku, gdy nie korzysta się z usług DNS. Przy wyłączonej opcji odwzorowywania adresów błędne polecenia nadal są interpretowane jako nazwy stacji, ale router stara się wyszukać je w statycznej tablicy nazw. Ponieważ taka procedura nie wymaga odczekiwania określonego czasu, kończy się natychmiast, umożliwiając wprowadzenie poprawnego polecenia:

```
Router1#pnig
Translating "pnig"
% Unknown command or computer name, or unable to find computer address
Router1#
```

Routery, które zostały poprawnie skonfigurowane do pracy w wykorzystaniem serwerów DNS (co opisano w recepturze 2.9), domyślnie będą próbowały odwzorować błędnie wprowadzone polecenia na adresy IP. Jednak ze względu na fakt, że na zapytania będzie odpowiadał serwer nazw, czas oczekiwania nieco się skróci. Router przesyła wówczas zapytania do wszystkich serwerów nazw w określonej kolejności, aż do uzyskania odpowiedzi lub wyczerpania liczby serwerów:

```
Router1#pnig
Translating "pnig"...domain server (172.25.1.1) (10.1.20.5)
% Unrecognized host or address, or protocol not running.

Router1#
```

Taki sposób postępowania w przypadkach wystąpienia błędów jest wyjątkowo nieefektywny i niepraktyczny przy korzystaniu z serwerów DNS. Warto spróbować rozwiązać problem inaczej.

Router próbuje dokonywać odwzorowania błędów typograficznych dlatego, że każda linia VTY domyślnie korzysta z usługi Telnet jako metody transportowej. Oznacza to, że można rozpocząć sesję Telnet, wprowadzając w wierszu poleceń jedynie nazwę stacji docelowej. Nie ma potrzeby wprowadzania samego polecenia `telnet`, dlatego ciąg tekstowy

pnig jest interpretowany przez router jako polecenie telnet pnig. Jeżeli jednak w metodzie transportowej zostanie przypisana wartość none, router nie będzie podejmował prób ustanowienia połączenia, dopóki w wierszu poleceń nie zostanie wprost podana instrukcja telnet:

```
Router1(config)#line vty 0 4
Router1(config-line)#transport preferred none
```

Proponowane rozwiązanie eliminuje problem niewłaściwej interpretacji błędów typograficznych i uznawania ich za nazwy stacji:

```
Router1#pnig
        ^
% Invalid input detected at '^' marker.

Router1#
```

Po wprowadzeniu omówionych zmian urządzenie uznaje nieznane ciągi tekstowe za błędne polecenia, a nie za nazwy stacji. W praktyce metoda ta wydaje się najbardziej użyteczna, gdyż pozwala na korzystanie z systemu DNS.

Zobacz również

Receptura 2.8, receptura 2.9.

2.11. Określanie czasu ponownego uruchomienia routera

Problem

Chcemy, żeby o określonej godzinie router automatycznie przeładował własny system.

Rozwiązanie

Zastosowanie polecenia reload in umożliwia określenie czasu, po jakim router automatycznie przeładuje system:

```
Router1#reload in 20
Reload scheduled for 11:33:53 EST Sat Feb 1 2003 (in 20 minutes)
Proceed with reload? [confirm] <enter>
Router1#
```

Korzystając z polecenia reload at, można określić dokładną datę i godzinę, o której procedura ponownego uruchomienia zostanie przeprowadzona:

```
Router1#reload at 14:00 Feb 2
Reload scheduled for 14:00:00 EST Sun Feb 2 2003 (in 26 hours and 44 minutes)
Proceed with reload? [confirm] <enter>
Router1#
```

Wyznaczając datę i czas ponownego uruchomienia, zaleca się wykorzystanie dokładnego źródła czasu, które zagwarantuje, że restart nastąpi w odpowiednim momencie. Więcej informacji na temat czasu i jego źródeł zamieszczono w rozdziale 14.

Analiza

Zazwyczaj administrator zatrzymuje i ponownie uruchamia router osobiście. Niekiedy jednak może wystąpić konieczność wykonania tej czynności o określonej godzinie. Przykładowo, restart jest jedynym sposobem usprawnienia błędnie podzielonej pamięci routera. Jednak z pewnością nikt nie chciałby przeprowadzać takiej operacji w czasie godzin pracy. Dzięki zastosowaniu prezentowanej metody można zlecić wykonanie zadania o północy lub w innym czasie, kiedy natężenie ruchu jest relatywnie małe.

Innym przykładem wykorzystania mechanizmu opóźnionego restartu jest chęć niedopuszczenia do zablokowania komunikacji z routerem podczas wykonywania potencjalnie niebezpiecznych zmian w plikach konfiguracyjnych. Ewentualność uniemożliwienia sobie korzystania z routera podczas wprowadzania zmian konfiguracyjnych występuje dość często — na przykład podczas modyfikowania list dostępowych do komunikowania się z urządzeniem lub w chwili konfigurowania mechanizmów routingu. Rozwiązaniem problemu może być poprzedzenie jakichkolwiek działań przekazaniem do routera polecenia wykonania automatycznego restartu po 15 minutach. Wówczas w przypadku zablokowania komunikacji z urządzeniem administrator nie będzie mógł zapisać ustawień konfiguracji pracy w pamięci NVRAM. Zatem w chwili, gdy router uruchomi się ponownie, uaktywniona zostanie wcześniejsza konfiguracja. Niewłaściwe ustawienia zostaną automatycznie anulowane.

Jeżeli po zakończeniu procedury konfiguracyjnej wszystkie zmiany okażą się właściwe, można zapisać ustawienia w pamięci NVRAM i odwołać automatyczny restart (odwoływanie restartu zostanie opisane w dalszej części receptury).

Polecenie `reload in` umożliwia również określenie przyczyn restartu:

```
Router1#reload in 1:20 Uaktualnienie IOS
Reload scheduled for 12:37:45 EST Sat Feb 1 2003 (in 1 hour and 20 minutes)
Reload reason: Uaktualnienie IOS
Proceed with reload? [confirm] <enter>
Router1#
```

Tekst wprowadzany po wartości czasu jest uznawany przez polecenie `reload` za opis powodu, dla którego wykonuje się restart. Wprowadzone informacje są rejestrowane w dzienniku zdarzeń podczas przystępowania do procedury ponownego uruchomienia urządzenia. Opcja ta została wprowadzona w systemie IOS 12.2. W zapisanym komunikacie oprócz samego opisu przyczyn restartu znajdują się informacje o czasie wprowadzenia polecenia, czasie wykonania procedury oraz nazwie użytkownika, który zażądał ponownego uruchomienia routera:

```
Feb  1 11:17:47: %SYS-5-SCHEDULED_RELOAD: Reload requested for 12:37:45 EST
Sat Feb 1 2003 at 11:17:45 EST Sat Feb 1 2003 by marek on vty0 (172.25.1.1).
Reload Reason: Uaktualnienie IOS.
```

Przyczynę restartu można również określić podczas wprowadzania polecenia `reload at`:

```
Router1#reload at 23:20 Feb 15 Uaktualnienie IOS
Reload scheduled for 23:20:00 EST Sat Feb 15 2003 (in 124 hours and 48 minutes)
Reload reason: Uaktualnienie IOS
Proceed with reload? [confirm] <enter>
Router1#
```

Polecenie `show reload` wyświetla informacje o wszystkich oczekujących procedurach ponownego uruchomienia:

```
Router1#show reload
Reload scheduled for 12:37:45 EST Sat Feb 1 2003 (in 1 hour and 19 minutes) by
marek on vty0 (172.25.1.1)
Reload reason: Uaktualnienie IOS
Router1#
```

Aby anulować zaprogramowany restart, należy zastosować polecenie `reload cancel`:

```
Router1#reload cancel
Router1#

***
*** --- SHUTDOWN ABORTED ---
***
```

Po anulowaniu procedury router rozsyła komunikat systemowy, który informuje wszystkich użytkowników, że restart został anulowany. W systemach IOS od wersji 12.2 odwołanie zaplanowanego przeładowania systemu jest również zapisywane w dzienniku zdarzeń:

```
Feb  1 11:19:10: %SYS-5-SCHEDULED_RELOAD_CANCELLED:  Scheduled reload
cancelled at 11:19:10 EST Sat Feb 1 2003
```

Po zaplanowaniu ponownego uruchomienia systemu router okresowo przesyła do użytkowników informację przypominającą o wyłączeniu. Domyślne ustawienia gwarantują wysyłanie powiadamiania na godzinę, 30 minut, 5 minut i minutę przed rozpoczęciem procedury. Odwołanie restartu jest możliwe w dowolnym momencie przed jego wykonaniem.

Komunikaty o wyłączeniu urządzenia mają następującą postać:

```
Router1#

***
*** --- SHUTDOWN in 1:00:00 ---
***

***
*** --- SHUTDOWN in 0:30:00 ---
***

***
*** --- SHUTDOWN in 0:05:00 ---
***

***
*** --- SHUTDOWN in 0:01:00 ---
***
Connection closed by foreign host.
```

Zobacz również

Rozdział 14.

2.12. Awaryjne zrzuty pamięci do pliku

Problem

W pracy routera wystąpiły poważne błędy i trzeba wykonać zrzut pamięci, by mógł on być przesłany do centrum pomocy technicznej Cisco (TAC — ang. *Technical Assistance Center*).

Rozwiązanie

Aby dokonać zrzutu pamięci routera po wystąpienia poważniejszego błędu, trzeba posłużyć się poleceniem `exception dump` i poinformować urządzenie, w jaki sposób przesłać dane do serwera:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#ip ftp source-interface Loopback0
Router1(config)#ip ftp username ijbrown
Router1(config)#ip ftp password ijpassword
Router1(config)#exception protocol ftp
Router1(config)#exception region-size 65536
Router1(config)#exception dump 172.25.1.3
Router1(config)#end
Router1#
```

Analiza

Prezentowana receptura jest jedną z tych, które powinny być wykorzystywane jak najrzadziej. Główną przyczyną wykonywania zrzutów pamięci routera jest konieczność dostarczenia danych do centrum pomocy technicznej firmy Cisco, które umożliwią zdiagnozowanie problemu. W przypadku poważniejszych problemów centrum TAC zawsze prosi o dosłanie danych na temat stanu pamięci routera w chwili awarii. Niniejsza receptura powinna zatem przygotować Czytelnika do właściwego reagowania w przypadku wystąpienia problemów.

Zrzut jest zapisem zawartości pamięci routera na chwilę przed wykonaniem wymuszonej procedury restartu. Pozyskane dane muszą zostać przesłane do serwera, gdyż jest ich zbyt dużo, by mogły być zapisane w pamięci trwałej.

Podczas zrzutu tak naprawdę tworzone są dwa pliki. Jeden z nich odpowiada głównej pamięci systemu, a drugi pamięci układów wejścia-wyjścia. Na podstawie zapisanych danych inżynierowie Cisco mogą określić przyczynę niewłaściwego zachowania oprogramowania i przygotować poprawkę uwzględnianą w kolejnych wersjach IOS.

Domyślnie transfer danych zrzutu odbywa się za pośrednictwem protokołu TFTP. Jednak stanowczo zaleca się zastąpienia go protokołem FTP. Większość aplikacji TFTP odmawia posłuszeństwa, gdy ilość przesyłanych informacji przekracza 16 MB. Wówczas jedynym skutecznym sposobem dostarczenia danych jest wykorzystanie protokołu FTP. Ponadto protokół FTP gwarantuje większą skuteczność transferu plików niż protokół TFTP. W prezentowanym przykładzie zademonstrowano sposób zastosowania do przesyłania plików zrzutów protokołu FTP. Więcej informacji na temat konfigurowania routera do pracy z protokołem FTP znajduje się w recepturze 1.14.

Zrzuty pamięci są potencjalnie zagrożone występowaniem błędów, gdyż router nie wykonuje ich, aż do chwili zaistnienia poważniejszego problemu programowego. Błąd oprogramowania może naruszyć zawartość pamięci routera i uczynić dalsze przetwarzanie danych niemożliwym (włącznie ze sporządzeniem zrzutu pamięci). Prawdopodobieństwo poprawnego wykonania operacji można znacznie zwiększyć, przeznaczając mały obszar pamięci na kopię zapasową wykorzystywaną w chwili naruszenia pamięci podstawowej:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#exception region-size 65536
Router1(config)#end
Router1#
```

Obszar pamięci przeznaczony na tworzenie zrzutu można definiować. Wartość domyślna wynosi 16 384 bajty, zaleca się jednak zwiększenie jej do 65 536 bajtów. Dzięki temu szansa na wykonanie prawidłowego zrzutu pamięci znacznie rośnie.

Zgodnie z ustawieniem domyślnym router tworzy dwa pliki zrzutów o nazwach *nazwaStacji-core* i *nazwaStacji-coreiomem*. W analizowanym przykładzie nazwą routera jest *Router1*, zatem pliki miałyby nazwy *Router1-core* i *Router1-coreiomem*. Aby zmienić konwencję nazewniczą, należy zastosować polecenie `exception core-file`:

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#exception core-file router5 compress
Router1(config)#end
Router1#
```

Warto zwrócić uwagę na fakt, że polecenie zostało uzupełnione o dodatkową opcję — `compress`. Dzięki niej router kompresuje plik zrzutu przed przesłaniem go do serwera. Przywrócenie pliku do pierwotnej postaci sprowadza się do wykonania na serwerze uniksowego polecenia `uncompress`. Nie zaleca się jednak korzystania z opcji kompresji, gdyż wiąże się ona z dodatkowym obciążeniem procesora i pamięci, które i tak są w przypadku awarii przeciążone. Ponadto wykorzystanie wspomnianej opcji wcale nie zmniejsza w istotny sposób rozmiaru pliku. W niektórych przypadkach powoduje wręcz zwiększenie jego objętości.

Przygotowując serwer, należy się upewnić, że jest na nim zarezerwowany odpowiedni obszar dysku, który pozwoli na zapisanie dwóch plików zrzutów. Rozmiary plików różnią się w zależności od rodzaju routera, a dokładniej — w zależności od ilości pamięci w nim zainstalowanej. Należy więc przyjąć, że pliki zrzutu mają taki sam rozmiar jak cała zainstalowana w routerze pamięć.

Aby wymusić na urządzeniu dokonanie zrzutu pamięci w trakcie normalnej pracy, można wykorzystać polecenie `core`. Zastosowanie instrukcji `write core` pozwala na sprawdzenie, czy wszystko jest przygotowane na ewentualną awarię systemu:

```
Router1#write core
Remote host [172.25.1.3]? <enter>
Base name of core files to write [Router1-core]? <enter>
Writing Router1-coreiomem
!!!
```

sieci są w nim pozyskiwane bezpośrednio z routerów. Wyniki działania skryptu są zapisywane w pliku CSV, który można w łatwy sposób importować do aplikacji arkuszy kalkulacyjnych.

Przykład 2.1. netstat.pl

```
#!/usr/local/bin/perl
#
#      netstat.pl -- skrypt pozyskuje od innych routerów szczegółowe
       informacje
#                    na temat adresowania w podsieciach poszczególnych
                     interfejsów.
#
# Ustawienie parametrów środowiska
$workingdir="/home/cisco/net";
$snmpro="ORARO";
#
$rtrlist="$workingdir/RTR_LIST";
$snmpwalk="/usr/local/bin/snmpwalk -v 1 -c $snmpro";
$snmpget="/usr/local/bin/snmpget -v 1 -c $snmpro";
open (RTR, "$rtrlist") || die "Nie można otworzyć pliku $rtrlist";
open (CSV, ">$workingdir/RESULT.csv") || die "Nie można otworzyć pliku
RESULT.csv";
while (<RTR>) {
   chomp($rtr="$_");
   @ifIndex=`$snmpwalk $rtr .1.3.6.1.2.1.4.20.1.2`;
   @ipAddress=`$snmpwalk $rtr .1.3.6.1.2.1.4.20.1.1`;
   @ipMask=`$snmpwalk $rtr .1.3.6.1.2.1.4.20.1.3`;
   $arraynum=0;
   print CSV "\n$rtr\n";
   print CSV "Interfejs, Adres IP, Maska, MTU, Szybkość, Admin, Stan\n";
   for $ifnumber (@ifIndex) {
      chomp(($foo, $ifnum) = split(/= /, $ifnumber));
      $ifDescription=`$snmpget $rtr ifDescr.$ifnum`;
      $ifMTU=`$snmpget $rtr ifMtu.$ifnum`;
      $ifSpeed=`$snmpget $rtr ifSpeed.$ifnum`;
      $ifAdminstatus=`$snmpget $rtr ifAdminStatus.$ifnum`;
      $ifOperstatus=`$snmpget $rtr ifOperStatus.$ifnum`;
      chomp(($foo, $ipaddr) = split(/: /, $ipAddress[$arraynum]));
      chomp(($foo, $mask) = split(/: /, $ipMask[$arraynum]));
      chomp(($foo, $ifdes, $foo) = split(/"/, $ifDescription));
      chomp(($foo, $mtu) = split (/= /, $ifMTU));
      chomp(($foo, $speed) = split (/: /, $ifSpeed));
      chomp(($foo, $admin) = split (/= /, $ifAdminstatus));
      chomp(($foo, $oper) = split (/= /, $ifOperstatus));
      if ( $speed > 3194967295 ) { $speed = 0 };
      $admin =~ s/\(.*\)//;
      $oper  =~ s/\(.*\)//;
      if ( $oper eq "dormant" ) { $oper = "up(spoofing)"};
      $speed = $speed/1000;
      if ( $speed > 1000) {
         $speed = $speed/1000;
         $speed =~ s/$/ Mbit\/s/;
      }
      else {
      $speed =~ s/$/ Kbit\/s/;
      }
      print CSV "$ifdes,$ipaddr,$mask,$mtu,$speed,$admin,$oper\n";
      $arraynum++;
   }
}
close(RTR);
close(CSV);
```

Analiza

Skrypt *netstat.pl* pobiera informacje o adresach IP podsieci z routerów umieszczonych na liście. Wykorzystuje w swoim działaniu protokół SNMP. Gwarantuje poprawność i aktualność danych. Informacje o wszystkich interfejsach routera są zapisywane w postaci pliku CSV.

Działanie skryptu *netstat.pl* wymaga istnienia w katalogu */usr/local/bin* interpretera języka Perl oraz pakietu NET-SNMP. Więcej informacji na temat języka Perl i pakietu NET-SNMP zamieszczono w dodatku A. Jeżeli wymienione programy znajdują się w innym katalogu, należy odpowiednio zmodyfikować skrypt.

Przed użyciem skryptu trzeba zdefiniować wartości dwóch zmiennych — `$workingdir` i `$snmpro`. Zmienna `$workingdir` musi zawierać nazwę katalogu, w którym przechowywany jest plik skryptu i jego pliki wynikowe. Z kolei zmienna `$snmpro` zawiera ciąg tekstowy przeznaczonej tylko do odczytu danych wspólnoty SNMP routerów. Zakłada się, że podany ciąg tekstowy wspólnoty jest jednakowy we wszystkich urządzeniach.

Działanie skryptu polega na przesyłaniu zapytań do kolejnych routerów z listy zapisanej w pliku *RTR_LIST* umieszczonej w katalogu roboczym skryptu. Lista routerów powinna się składać z nazw urządzeń lub ich adresów IP zapisanych w oddzielnych liniach pliku. W liniach tych nie wolno umieszczać komentarzy lub innych danych. Wynik działania skryptu jest zapisywany w pliku o nazwie *RESULT.csv* umieszczonym w katalogu roboczym.

Zawartość pliku *RESULT.csv* można importować do arkusza kalkulacyjnego. Wynik wykonania operacji powinien być zbliżony do prezentowanego w tabeli 2.3.

Prezentowany skrypt analizuje informacje o wszystkich interfejsach pracujących w protokole IP. Uwzględnia interfejsy wyłączone (*down*) przez administratora, interfejsy pętli zwrotnych (*loopback*), adresy HSRP oraz interfejsy o nieprzydzielonym adresie IP. Skrypt nie analizuje interfejsów, które nie pracują w protokole IP, i elementów składowych interfejsów.

Z uwagi na fakt, że wykorzystano jedynie wartości MIB otwartego standardu SNMP, prezentowane rozwiązanie można zastosować do pobierania danych z dowolnego urządzenia pracującego zgodnie z założeniami protokołu SNMP, w tym z urządzeń firm innych niż Cisco.

Zobacz również

Dodatek A.

Tabela 2.3. Przykładowy efekt wykonania skryptu netstat.pl

Detroit

Interfejs	Adres IP	Maska	MTU	Szybkość	Admin	Stan
Serial0/0	10.1.1.1	255.255.255.252	1500	768 Kbps	up	up
Loopback0	10.2.2.2	255.255.255.252	1514	0 Kbps	up	up
FastEthernet1/0	172.22.1.4	255.255.255.0	1500	100 Mbps	up	up
Ethernet0/0	172.25.1.8	255.255.255.0	1500	10 Mbps	down	down

Toronto

Interfejs	Adres IP	Maska	MTU	Szybkość	Admin	Stan
BRI0	10.1.99.55	255.255.255.0	1500	64 Kbps	down	down
Ethernet0	172.25.1.7	255.255.255.0	1500	10 Kbps	up	up
Loopback0	172.25.25.6	255.255.255.255	1514	0 Kbps	up	up

Boston

Interfejs	Adres IP	Maska	MTU	Szybkość	Admin	Stan
Serial0.1	172.20.1.2	255.255.255.252	0	28 Kbps	up	up
Ethernet0	172.20.10.1	255.255.255.0	1500	10 Mbps	up	up
Loopback0	172.20.100.1	255.255.255.255	1514	0 Kbps	up	up

2.14. Generowanie raportu zawierającego informacje o tablicy routingu

Problem

Chcemy pobrać z jednego z routerów przechowywaną przez niego tablicę routingu.

Rozwiązanie

Przedstawiony w przykładzie 2.2 skrypt *rt.pl* wykorzystuje protokół SNMP do pozyskiwania z określonego routera danych o tablicy routingu. Następnie przekazuje zgromadzone informacje na standardowe wyjście (STDOUT). Nazwa routera lub jego adres IP muszą być przekazane jako parametr wywołania skryptu.

Przykład 2.2. rt.pl

```
#!/usr/bin/perl
#
#           rt.pl -- skrypt pobiera z routera jego
```

```
#                    tablicę routingu.
#
# Ustawienie parametrów środowiska
$snmpro="ORARO";
#
$x=0;
$snmpwalk="/usr/local/bin/snmpwalk -v 1 -c $snmpro";
$snmpget="/usr/local/bin/snmpget -v 1 -c $snmpro";
chomp ($rtr=$ARGV[0]);
if ( $rtr eq "" ) {die "$0: Nie wskazano routera\n"};
print "Adres docelowy\tMaska\t\tNastępny router";
print "\t\t Protokół\tInterfejs\n";
@iftable=`$snmpwalk $rtr ifDescr`;
for $ifnum (@iftable) {
    chomp (($intno, $intname) = split (/ = /, $ifnum));
    $intno=~s/.*ifDescr\.//;
    $intname=~s/"//gi;
    $int{$intno}=$intname;
}
@ipRouteDest=`$snmpwalk $rtr ipRouteDest`;
@ipRouteMask=`$snmpwalk $rtr ipRouteMask`;
@ipRouteNextHop=`$snmpwalk $rtr ipRouteNextHop`;
@ipRouteProto=`$snmpwalk $rtr ipRouteProto`;
@ipRouteIfIndex=`$snmpwalk $rtr ipRouteIfIndex`;
#@ipRouteMetric1=`$snmpwalk $rtr ipRouteMetric1`;
for $intnum (@ipRouteIfIndex) {
    chomp (($foo, $int) = split (/= /, $intnum));
    chomp (($foo, $dest) = split (/: /, @ipRouteDest[$x]));
    chomp (($foo, $mask) = split (/: /, @ipRouteMask[$x]));
    chomp (($foo, $nhop) = split (/: /, @ipRouteNextHop[$x]));
    chomp (($foo, $prot) = split (/= /, @ipRouteProto[$x]));
    #chomp (($foo, $metr) = split (/= /, @ipRouteMetric1[$x]));
    $int1 = $int{$int};
    if ($int1 eq '') {$int1="Local"};
    $prot=~s/\(.*//; $prot=~s/ciscoIgrp/\(e\)igrp/;
    printf ("%-15s %-15s %-15s %7s %-25s\n",$dest, $mask, $nhop, $prot,
$int1);
    $x++;
}
```

Analiza

Skrypt *rt.pl* został napisany w języku Perl i wykorzystuje pakiet NET-SNMP. Jego działanie polega na pozyskiwaniu za pomocą protokołu SNMP informacji o tablicy routingu wskazanego routera. Interpreter języka Perl i pakiet NET-SNMP muszą się znajdować w katalogu */usr/local/bin*. Więcej informacji na temat samego języka oraz pakietu NET-SNMP zamieszczono w dodatku A.

Przed uruchomieniem skryptu należy zdefiniować wartość zmiennej `$snmpro`, która przechowuje ciąg tekstowy wspólnoty SNMP:

```
Freebsd% ./rt.pl toronto
Adres docelowy  Maska           Następny router  Protokół     Interfejs
10.1.1.0        255.255.255.252 172.25.1.5       ospf         Ethernet0
10.2.2.2        255.255.255.255 172.25.1.5       ospf         Ethernet0
172.16.2.0      255.255.255.0   172.25.1.5       ospf         Ethernet0
```

```
172.20.0.0      255.255.0.0     172.25.1.5       local        Local
172.20.1.0      255.255.255.252 172.25.1.5       ospf         Ethernet0
172.20.10.0     255.255.255.0   172.25.1.5       ospf         Ethernet0
172.20.100.1    255.255.255.255 172.25.1.5       ospf         Ethernet0
172.22.0.0      255.255.0.0     172.25.1.5      (e)igrp       Ethernet0
172.22.1.0      255.255.255.0   172.25.1.5       ospf         Ethernet0
172.25.1.0      255.255.255.0   172.25.1.7       local        Ethernet0
172.25.2.0      255.255.255.252 172.25.1.5      (e)igrp       Ethernet0
172.25.25.1     255.255.255.255 172.25.1.5      (e)igrp       Ethernet0
172.25.25.6     255.255.255.255 172.25.25.6      local        Loopback0
172.25.26.4     255.255.255.252 172.25.1.5      (e)igrp       Ethernet0
172.25.26.5     255.255.255.255 172.25.1.5       ospf         Ethernet0
Freebsd%
```

Wyniki udostępniane przez skrypt są raczej łatwe w interpretacji, poza trasami statycznymi i połączeniami bezpośrednimi, które wymagają krótkiego wyjaśnienia.

W przypadku tras statycznych w kolumnie *Protokół* występuje wartość *local*, a w kolumnie *Interfejs — Local*. Połączenia bezpośrednie są również oznaczane wartością *local* w polu *Protokół*, ale w kolumnie interfejsu zamieszczana jest nazwa rzeczywistego interfejsu związanego z daną trasą.

Przykładowo, wiersz *172.20.0.0 255.255.0.0* odpowiada trasie statycznej:

```
172.20.0.0      255.255.0.0     172.25.1.5       local        Local
```

podczas gdy wiersz *172.25.1.0 255.255.255.0* odpowiada połączeniu bezpośredniemu:

```
172.25.1.0      255.255.255.0   172.25.1.7       local        Ethernet0
```

Ze względu na fakt, że skrypt wykorzystuje jedynie powszechnie znane wartości MIB SNMP, można go użyć do pobierania informacji o routingu IP również z innych urządzeń obsługujących protokół SNMP, w tym także z urządzeń pochodzących od innych dostawców niż Cisco.

Zobacz również

Dodatek A.

2.15. Generowanie raportu zawierającego informacje z tablicy ARP

Problem

Chcemy pobrać z jednego z routerów tablicę ARP, która zawiera dane o adresie MAC skojarzonym z określonym adresem IP lub o adresie IP odpowiadającym danemu adresowi MAC.

Rozwiązanie

Zadanie przedstawionego w przykładzie 2.3 skryptu *arpt.pl* polega na pobieraniu tablicy ARP z określonego routera lub ze stacji o określonym IP i wyświetlaniu wyników na standardowym wyjściu (STDOUT). Nazwa lub adres IP routera powinny być podane jako parametr linii poleceń.

Przykład 2.3. arpt.pl

```
#!/usr/local/bin/perl
#
#         arpt.pl -- skrypt pobiera tablicę ARP z określonego routera.
#
# Ustawienie parametrów środowiska
$snmpro="ORARO";
#
$snmpwalk="/usr/local/bin/snmpwalk -v 1 -c $snmpro";
$snmpget="/usr/local/bin/snmpget -v 1 -c $snmpro";
chomp ($rtr=$ARGV[0]);
if ( $rtr eq "" ) {die "$0: Nie wskazano routera \n"};
@iftable=`$snmpwalk $rtr ifDescr`;
for $ifnum (@iftable) {
    chomp (($intno, $intname) = split (/ = /, $ifnum));
    $intno=~s/.*ifDescr\.//;
    $intname=~s/"//gi;
    $arpint{$intno}=$intname;
}
printf ("%-22.22s %-10.10s  %-25.25s\n", Adres IP, Adres MAC, Interfejs);
@atTable=`$snmpwalk $rtr .1.3.6.1.2.1.3.1.1.1`;
for $atnum (@atTable) {
    chomp (($atip, $atint) = split (/ = /, $atnum));
    $atip =~ s/.*atIfIndex\.[0-9]+\.1\.//;
    $atphys=`$snmpget $rtr atPhysAddress.$atint.1.$atip`;
    chomp(($foo, $phys) = split(/: /, $atphys));
    $phys=~s/ /-/gi; chop ($phys);
    $phys=~tr/A-Z/a-z/;
    $int=$arpint{$atint};
    printf ("%-15.15s %17.17s  %-25.25s\n", $atip, $phys, $int);
}
```

Analiza

Zadaniem skryptu *arpt.pl* jest pobieranie zawartości tablicy ARP ze wskazanego routera, przy wykorzystaniu do tego celu protokołu SNMP. Wynik działania kodu przekazywany jest na standardowe wyjście STDOUT. Do uruchomienia skryptu konieczne jest posiadanie interpretera języka Perl oraz pakietu NET-SNMP. Obydwa programy powinny się znajdować w katalogu */usr/local/bin*. Więcej informacji o języku Perl i pakiecie NET-SNMP zamieszczono w dodatku A.

Przed uruchomieniem skryptu należy prawidłowo zdefiniować ciąg tekstowy wspólnoty SNMP (przechowywany w zmiennej `$snmpro`):

```
Freebsd% ./arpt.pl toronto
Adres IP            Adres MAC      Interfejs
172.22.1.1      00-01-96-70-b7-81  FastEthernet0/1
```

```
172.22.1.2      00-01-96-70-b7-81  FastEthernet0/1
172.22.1.3      00-01-96-70-b7-81  FastEthernet0/1
172.25.1.1      00-10-4b-09-57-00  FastEthernet0/0.1
172.25.1.5      00-01-96-70-b7-80  FastEthernet0/0.1
172.25.1.7      00-00-0c-92-bc-6a  FastEthernet0/0.1
172.25.1.254    00-00-0c-07-ac-01  FastEthernet0/0.1
172.16.2.1      00-01-96-70-b7-80  FastEthernet0/0.2
172.16.2.22     00-00-0c-07-ac-00  FastEthernet0/0.2
Freebsd%
```

Raport będący wynikiem działania skryptu składa się z informacji o adresach IP, adresach MAC i nazwach interfejsów odpowiadających poszczególnym wpisom w tablicy ARP. Aby wyszukać dane o konkretnym urządzeniu, można zastosować jedno z narzędzi przeszukiwania tekstu. W serwerze uniksowym byłby to program *grep*, którego wywołanie powinno wyglądać następująco:

```
Freebsd% ./arpt.pl toronto | grep 172.25.1.5
172.25.1.5      00-01-96-70-b7-80  FastEthernet0/0.1
Freebsd%
```

Tablice ARP routerów w sieci szkieletowej mogą mieć całkiem duże rozmiary, co znacznie utrudnia wyszukiwanie konkretnych wpisów ARP. Dzięki prezentowanemu skryptowi operacje przeszukiwania można wykonywać zdalnie, a dzięki poleceniu `grep` możliwe jest również wyszukiwanie adresów IP na podstawie znanych adresów MAC:

```
Freebsd% ./arpt.pl toronto | grep 00-01-96-70-b7-81
172.22.1.3      00-01-96-70-b7-81  FastEthernet0/0.1
Freebsd%
```

Proponowane rozwiązanie wykorzystuje jedynie wartości MIB otwartego standardu SNMP. Nic więc nie stoi na przeszkodzie, żeby zastosować je również w odniesieniu do innego urządzenia obsługującego protokół SNMP, w tym również we współpracy z jednostkami dostarczanymi przez firmy inne niż Cisco.

Zobacz również

Dodatek A.

2.16. Generowanie pliku nazw stacji

Problem

Chcemy utworzyć plik składający się z informacji o adresach IP i nazwach interfejsów wszystkich routerów działających w danej sieci.

Rozwiązanie

Prezentowany w przykładzie 2.4 skrypt języka Perl *host.pl* tworzy plik nazw stacji, w którym umieszczane są informacje o adresach IP wszystkich routerów znajdujących się na liście urządzeń. Skrypt został napisany w języku Perl i wymaga zainstalowania pakietu NET-SNMP. Podczas wywoływania programu nie trzeba podawać żadnych parametrów.

Przykład 2.4. host.pl

```
#!/usr/local/bin/perl
#
#         host.pl -- skrypt tworzy plik nazw stacji, wykorzystując
#                    informacje pobrane z routerów umieszczonych na liście.
#
# Ustaw parametry środowiska
$workingdir="/home/cisco/net";
$snmpro="ORARO";
#
$rtrlist="$workingdir/RTR_LIST";
$snmpwalk="/usr/local/bin/snmpwalk -v 1 -c $snmpro";
$snmpget="/usr/local/bin/snmpget -v 1 -c $snmpro";
open (RTR, "$rtrlist") || die "Nie można otworzyć pliku $rtrlist";
open (RESULT, ">$workingdir/RESULT") || die "Nie można otworzyć pliku RESULT";
while (<RTR>) {
   chomp($rtr="$_");
   @ifIndex=`$snmpwalk $rtr ipAdEntIfIndex`;
   @ipAddress=`$snmpwalk $rtr ipAdEntAddr`;
   $rtr1=`$snmpget $rtr .1.3.6.1.4.1.9.2.1.3.0`;
   chomp(($foo, $RTR) = split (/"/, $rtr1));
   $arraynum=0;
   for $ifnumber (@ifIndex) {
      chomp(($foo, $ifnum) = split(/= /, $ifnumber));
      $ifDescription=`$snmpget $rtr ifName.$ifnum`;
      chomp(($foo, $ipaddr) = split(/: /, $ipAddress[$arraynum]));
      chomp(($foo, $ifdes) = split(/= /, $ifDescription));
      $name="$RTR-$ifdes";
      #$name=~s/\//-/;
      if ( $ifdes eq "Lo0" ) { $name=$RTR };
      print RESULT "$ipaddr\t\t$name\n";
      $arraynum++;
   }
}
close(RTR);
close(RESULT);
```

Analiza

W większości firm tablice nazw stacji budowane są ręcznie przez administratorów. Zazwyczaj są one później wykorzystywane przez serwery zarządzające konfiguracją sieci. Najczęściej z każdym routerem kojarzy się jeden dostępny bezpośrednio interfejs sieciowy. Prezentowany skrypt tworzy plik nazw, uwzględniając wszystkie znane adresy IP wszystkich rotuerów.

Poniżej zamieszczono przykład wynikowej listy generowanej przez skrypt *host.pl*:

```
10.1.1.1                miami-Se0/0
10.2.2.2                miami
172.20.6.8              miami-Se0/2
172.22.1.4              miami-Fa1/0
172.25.1.8              miami-Et0/0
10.1.1.2                toronto-Se0/1
172.20.1.1              toronto-Se0/0.2
172.22.1.1              toronto-Fa0/1
172.25.1.5              toronto-Fa0/0.1
172.25.2.1              toronto-Se0/0.1
172.25.25.1             toronto
172.25.26.5             toronto-Lo1
10.1.99.55              detroit-BR0
172.25.3.7              detroit-Et0
172.25.25.6             detroit
172.20.1.2              boston-Se0.1
172.20.10.1             boston-Et0
172.20.100.1            boston
```

Formatowanie danych wynikowych odpowiada standardowi stosowanemu w systemach Unix do tworzenia plików */etc/hosts*. Pobieranie adresów odbywa się przy wykorzystaniu protokołu SNMP. Do pobranych danych dopisywane są wartości nazw routerów oraz dane interfejsów. Dla każdego routera tworzony jest również wpis zawierający adres interfejsu `loopback0`, ale bez uwzględnienia nazwy interfejsu w nazwie stacji. W analizowanym przypadku jest to widoczne na przykładzie pozycji *boston*, gdzie nie został zastosowany mniej czytelny zapis *boston-Lo0*.

Tworzenie pliku nazw stacji jest wskazane z wielu powodów. Niekiedy routery przesyłają do serwera komunikaty (za pomocą protokołu SMTP lub narzędzia *syslog*), korzystając z adresów IP interfejsów zamiast z interfejsu pętli zwrotnej. Ponadto udostępnienie pliku nazw stacji sprawia, że wynik działania polecenia `traceroute` staje się łatwiejszy w analizowaniu:

```
Freebsd% traceroute miami
traceroute to miami (10.2.2.2), 64 hops max, 52 byte packets
 1  detroit-Et0      (172.25.3.7)  2.263 ms  2.210 ms  2.178 ms
 2  toronto-Fa0/0.1  (172.25.1.5)  3.042 ms  3.060 ms  3.846 ms
 3  boston-Se0.1     (172.20.1.2)  8.234 ms  8.245 ms  8.145 ms
 4  miami-Se0/2      (172.20.6.8)  9.893 ms  9.893 ms  9.432 ms
Freebsd%
```

Zastosowanie omawianego mechanizmu pozwala na rozszyfrowanie informacji o kolejnych urządzeniach, przez które przechodzi pakiet na drodze z zarządzanej stacji do routera w Miami. Zgromadzone dane umożliwiają nie tylko wskazanie routerów, przez które pakiety są przesyłane, ale także określenie interfejsów, za pomocą których informacje są przekazywane.

Skrypt nie uaktualnia bezpośrednio pliku */etc/hosts*. Dlatego prawdopodobnie konieczne będzie osobiste przeniesienie danych wynikowych do pliku */etc/hosts* określonego systemu. Pewnym ułatwieniem może być utworzenie nadrzędnego pliku z informacjami o stacjach, do którego dołączane będą dane pozyskane przez skrypt. Takie rozwiązanie umożliwia zastosowanie narzędzia *cron* do conocnego tworzenia aktualnej wersji tablicy adresów.

Przed uruchomieniem skryptu należy zmodyfikować wartości dwóch zmiennych. Pierwszą z nich jest `$workingdir`, która przechowuje nazwę katalogu roboczego programu. Druga zmienna to `$snmpro`. Jej wartością musi być ciąg tekstowy wspólnoty SNMP przeznaczonej tylko do odczytu. W skrypcie przyjęto założenie, że ten sam ciąg jest wykorzystywany we współpracy ze wszystkimi routerami.

Działanie kodu bazuje na odczytywaniu listy routerów i kierowaniu zapytań do kolejno wymienionych urządzeń. Plik zawierający wspomnianą listę powinien się znajdować w katalogu roboczym skryptu i nosić nazwę *RTR_LIST*. Poszczególnymi pozycjami listy mogą być adresy IP lub nazwy stacji, przy czym w jednej linii może się znajdować tylko jedna wartość. Nazwy stacji umieszczane na liście wynikowej są pobierane bezpośrednio z routerów, dlatego należy się upewnić, że każdemu urządzeniu przypisano unikatową nazwę. Wynik uruchomienia kodu znajduje się w pliku *RESULT* umieszczonym w katalogu roboczym.

Na koniec należy wspomnieć, że skrypt może generować nazwy, które nie będą zgodne z zaleceniem RFC 952 (ze specyfikacją tablicy nazw stacji wykorzystywaną w internecie — ang. *DoD Internet Host Table Specification*). Przyczyną niezgodności jest fakt umieszczania w nazwach stacji znaku ukośnika (np. *miami-Se0/2*), co w niektórych aplikacjach może być istotną przeszkodą. Dotyczy to szczególnie programów wykorzystujących adresy w formacie URL. Na przykład zapytanie skierowane do stacji o adresie *http://miami-Se0/2* z pewnością nie zostanie poprawnie zinterpretowane, gdyż ostatni znak (2) będzie uznawany za nazwę pliku. Niemniej większość powszechnie stosowanych aplikacji, takich jak *ping* i *Telnet*, bez problemu zaakceptuje taką postać adresu.

Osoby dbające o zgodność ze standardem lub korzystające z aplikacji, które nie obsługują wymienionych typów nazw, mogą wprowadzić niewielkie zmiany w kodzie skryptu pozwalające na zamianę ukośników na myślniki. W przedstawionym przykładzie listingu stosowna linia kodu została poprzedzona znakiem komentarza (`#`), który można usunąć, przekształcając wiersz:

```
#$name=~s/\//-/;
```

na:

```
$name=~s/\//-/;
```

Zobacz również

Dokument RFC 952.