helion.pl
KSIĘGARNIA INTERNETOWA · WYDAWNICTWO

IDŹ DO
PRZYKŁADOWY ROZDZIAŁ
SPIS TREŚCI

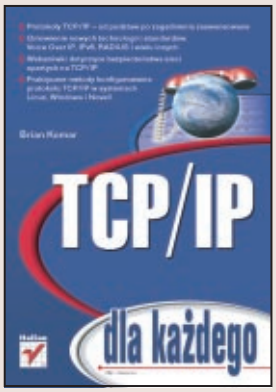
KATALOG KSIĄŻEK
KATALOG ONLINE
ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK
DODAJ DO KOSZYKA

CENNIK I INFORMACJE
ZAMÓW INFORMACJE O NOWOŚCIACH
ZAMÓW CENNIK

CZYTELNIA
FRAGMENTY KSIĄŻEK ONLINE

TCP/IP dla każdego



Autor: Brian Komar
Tłumaczenie: Paweł Koronkiewicz
ISBN: 83-7197-782-4
Tytuł oryginału: TY TCP/IP Networking in 21 Days
Format: B5, stron: 654

„TCP/IP dla każdego” przekaże Ci całą niezbędną wiedzę do administrowania siecią TCP/IP. Ujmuje zarówno zagadnienia podstawowe opisując dokładnie sam protokół, jak i kwestie bardziej skomplikowane, w tym konfigurowanie systemu DNS czy zarządzanie SNMP. Uwzględnione zostały najnowsze, normalizowane dopiero w ostatnich latach technologie, takie jak:

- Internet Protocol Security (IPSec)
- IPv6
- Voice Over IP
- Bezprzewodowe sieci lokalne
- Uwierzytelnianie RADIUS
- Infrastruktura klucza publicznego (PKI)

Książka omawia zarówno teoretyczne podstawy funkcjonowania sieci opartych na TCP/IP, jak i praktyczne sposoby konfigurowania protokołów w różnorodnych systemach operacyjnych stosowanych we współczesnych sieciach. Zgodnie z tytułem, jest to publikacja kierowana do szerokiego grona Czytelników: początkujących i zaawansowanych, dla administratora-praktyka i dla studenta. Temu ostatniemu z pewnością pomogą sprawdzające pytania, które podsumowują każdy rozdział książki.

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

Spis treści

O Autorze **15**

Wstęp **17**

Rozdział 1. Historia Internetu **21**

Jak powstał Internet? 21
Etap I — ARPAnet 22
Etap II — NSFNET 23
Internet dzisiaj 24
Kto jest odpowiedzialny za protokół TCP/IP? 24
Internet Architecture Board (IAB) 25
Requests for Comments (RFCs) 27
Cykl „dojrzewania" RFC 27
Jak uzyskać dokument RFC? 29
Podsumowanie 30
Pytania sprawdzające 31
W następnym rozdziale 31

Rozdział 2. Typy sieci i architektura systemów otwartych **33**

Typy sieci 33
Sieci lokalne (LAN) 33
Sieci rozległe (WAN) 41
Koncepcja systemów otwartych 48
Korzystanie z modeli warstwowych 49
Model odniesienia OSI 50
Model warstw TCP/IP 54
Porównanie modeli OSI i TCP/IP 58
Podsumowanie 59
Pytania sprawdzające 59
W następnym rozdziale 60

Rozdział 3. Adresy protokołu IP **61**

Adres IP 61
Jak zapisywany jest adres? 62
Klasy adresów IP 64
Ogólne zasady adresowania IP 66
Specjalne adresy IP 66
Znaczenie masek podsieci 67
Proces AND 68
Typowe problemy z maskowaniem 69

Adresy w sieci lokalnej ... 69
Przyszłość adresowania IP (IPv6) ... 71
Podsumowanie ... 71
Pytania sprawdzające ... 72
W następnym rozdziale ... 73

Rozdział 4. Rodzina TCP/IP — podstawowe protokoły ... 75

Podstawowe protokoły modelu warstw IP ... 76
Protokoły w warstwie międzysieciowej ... 76
Komunikacja połączeniowa i bezpołączeniowa ... 90
TCP — Transmission Control Protocol ... 90
UDP — User Datagram Protocol ... 91
Korzystanie z portów i gniazd ... 91
TCP — Transmission Control Protocol ... 93
Format nagłówka TCP ... 94
Ustanawianie sesji TCP, czyli trójstopniowa wymiana potwierdzeń ... 96
Zamykanie sesji TCP ... 97
Przepływ informacji w trakcie sesji TCP ... 98
Okna przesuwne TCP ... 99
Stany połączenia TCP ... 101
Zarezerwowane porty TCP ... 102
UDP — User Datagram Protocol ... 104
Format nagłówka UDP ... 105
Komunikacja przy użyciu UDP ... 105
Zarezerwowane porty UDP ... 106
Określanie, które porty są używane ... 106
Narzędzia trybu tekstowego ... 106
Korzystanie z narzędzi graficznych ... 110
Podsumowanie ... 111
Pytania sprawdzające ... 112
W następnym rozdziale ... 112

Rozdział 5. Sztuka maskowania podsieci ... 113

Maski modyfikowane ... 113
Właściwy podział sieci ... 115
Określanie liczby podsieci ... 115
Określanie liczby dostępnych stacji ... 118
Określanie dostępnych pul adresów dla wybranej maski ... 119
Tworzenie tabeli przeliczeniowej podsieci ... 121
Użycie tabeli przeliczeniowej do określania adresów klasy A ... 123
Użycie tabeli przeliczeniowej do określania adresów klasy B ... 124
Użycie tabeli przeliczeniowej do określania adresów klasy C ... 124
VLSM — zróżnicowana długość maski ... 126
Przykład trasowania VLSM ... 127
Wymagane warunki implementacji VLSM ... 129
Classless Inter-Domain Routing (bezklasowe trasowanie międzydomenowe) ... 130
Podsumowanie ... 131
Pytania sprawdzające ... 131
W następnym rozdziale ... 134

Rozdział 6. Odwzorowywanie adresów IP i nazw logicznych ... 135

Odwzorowanie adresów IP na adresy MAC ... 135
Odwzorowanie nazw logicznych na adresy IP ... 136

Odwzorowanie hostnames....137
Przestrzeń nazw domen....137
Proces odwzorowywania nazwy stacji....139
Podział ról w systemie DNS....140
Rodzaje zapytań DNS....144
Poprawianie wydajności DNS....147
Dynamiczne uaktualnianie zasobów DNS....148
Odwzorowanie nazw NetBIOS....153
Proces odwzorowywania nazw NetBIOS....155
Transakcje w sieciach NetBIOS....156
Serwery nazw NetBIOS....157
Porównanie serwerów NBNS i DNS....158
Pliki konfiguracyjne TCP/IP....159
HOSTS....159
NETWORKS....160
SERVICES....160
PROTOCOL....161
LMHOSTS....161
RESOLV.CONF....163
Podsumowanie....163
Pytania sprawdzające....163
W następnym rozdziale....164

Rozdział 7. Konfigurowanie serwerów DNS....165

Rejestrowanie nazwy DNS....165
Formaty komunikatów DNS....167
Rekordy zasobów....170
Konfigurowanie serwera DNS zgodnego z BIND....172
Plik named.conf....172
Konfigurowanie serwera DNS systemu Windows....184
Konfigurowanie serwera....184
Konfigurowanie przekazywania zapytań DNS....186
Tworzenie stref wyszukiwania odwrotnego....187
Tworzenie pliku wyszukiwania prostego....190
Tworzenie dalszych rekordów zasobowych....191
Problemy z DNS i narzędzie NSLOOKUP....192
Podsumowanie....194
Pytania sprawdzające....194
W następnym rozdziale....195

Rozdział 8. Konfigurowanie serwerów nazw NetBIOS....197

Format komunikatu NetBIOS....197
Wprowadzanie usług NetBIOS w sieci TCP/IP....199
Instalowanie Windows Internet Name Service (WINS)....200
Konfigurowanie stacji klienckich pod kątem ograniczenia komunikacji NetBIOS....201
Konfigurowanie środowiska serwera WINS....202
Dodawanie mapowań statycznych dla klientów nieobsługujących WINS....202
Rozwiązywanie problemów z NetBIOS — polecenie NBTSTAT....210
Odchodzenie od systemu NetBIOS....211
Podsumowanie....211
Pytania sprawdzające....211
W następnym rozdziale....212

Rozdział 9. Protokoły trasowania 213
Podstawowe zasady trasowania 213
Konfiguracje trasowania 215
Podstawowe problemy z trasowaniem 216
Zabezpieczanie systemu trasowania 218
Trasowanie statyczne 219
Protokoły trasowania 221
Protokoły bram zewnętrznych 222
Protokoły bram wewnętrznych 228
Rozwiązywanie problemów z trasowaniem 240
Microsoft Windows 240
Cisco IOS 241
Podsumowanie 241
Pytania sprawdzające 242
W następnym rozdziale 243

Rozdział 10. RARP, BOOTP i DHCP — protokoły automatycznego konfigurowania stacji 245
Zastosowanie automatycznej konfiguracji 245
Protokół RARP 246
Protokół BOOTP 247
Realizacja wymogu niezawodności w procesie BOOTP 247
Format komunikatu BOOTP 248
Protokół DHCP 249
Proces DHCP 250
Odnawianie DHCP 251
Konfigurowanie serwera DHCP 252
Strategie wdrażania DHCP 255
Podsumowanie 257
Pytania sprawdzające 258
W następnym rozdziale 258

Rozdział 11. Uwierzytelnianie w sieci TCP/IP 259
Uwierzytelnianie tekstowe 259
Network Information System (NIS) 261
Role serwerów NIS 261
Baza danych NIS 261
Domeny NIS 262
System Kerberos 262
Składniki systemu Kerberos 262
Proces uwierzytelniania Kerberos 263
Zalety systemu Kerberos 265
Infrastruktura klucza publicznego 265
Uzyskiwanie certyfikatu cyfrowego 266
Sprawdzanie certyfikatu 268
Mapowanie certyfikatów do kont użytkowników 269
Uwierzytelnianie WWW 271
Uwierzytelnianie anonimowe 271
Uwierzytelnianie podstawowe 272
Uwierzytelnianie skrócone 273
Uwierzytelnianie zintegrowane 274
Uwierzytelnianie certyfikatowe 275

Podsumowanie ... 276
Pytania sprawdzające ... 276
W następnym rozdziale ... 277

Rozdział 12. Szyfrowanie przesyłanych danych ... 279

Szyfrowanie danych w warstwie aplikacji ... 279
SSL i TLS ... 280
S/MIME i PGP ... 282
Zabezpieczanie danych w warstwie IP ... 282
Authentication Header (AH) ... 283
Encapsulating Security Payload (ESP) ... 284
Tryby IPSec ... 286
Procedury komunikacji IPSec ... 288
Planowane zmiany w IPSec ... 292
Problem ... 292
Rozwiązanie — nowy nagłówek ESP ... 293
Podsumowanie ... 295
Pytania sprawdzające ... 295
W następnym rozdziale ... 296

Rozdział 13. Bezpieczeństwo sieci ... 297

Zagrożenia dla bezpieczeństwa sieciowego ... 297
Jawne przesyłanie hasła ... 298
Rozpowszechnienie oprogramowania monitorującego ... 298
Podrabianie adresów ... 299
Słabe punkty w konfiguracji zabezpieczeń ... 300
Przygotowanie założeń ochrony sieci ... 301
Zapory firewall ... 302
O czym jeszcze należy pamiętać? ... 303
Typowe rozwiązania ... 303
Inne techniki ochrony sieci ... 310
Network Address Translation (NAT) ... 310
Intrusion Detection Systems (IDS) ... 311
Tworzenie „strefy zdemilitaryzowanej” ... 311
Zapory trójstronne ... 312
DMZ jako strefa środkowa ... 312
Wady i zalety różnych rozwiązań DMZ ... 313
Podsumowanie ... 314
Pytania sprawdzające ... 314
W następnym rozdziale ... 315

Rozdział 14. Aplikacje uruchamiania zdalnego ... 317

Protokół Telnet ... 318
Proces negocjowania opcji ... 319
Standardowe funkcje sterujące ... 321
Opisy znaków sterujących ASCII ... 322
Sekwencje unikowe Telnetu ... 323
Łączenie z serwerem usługi Telnet ... 324
Zdalne polecenia systemu UNIX ... 326
Ustawienia bezpieczeństwa ... 326
rlogin (remote login) ... 327
rsh (remote shell) ... 328
rexec (remote execute) ... 329

Usługi terminalowe jako narzędzie administracji zdalnej....329
Dostępne rozwiązania....330
Serwer terminali....331
Usługi terminalowe firmy Microsoft....331
Opcjonalne usługi TCP/IP....333
Instalowanie usług opcjonalnych w Windows 2000....334
Sprawdzanie instalacji usług opcjonalnych....335
Podsumowanie....336
Pytania sprawdzające....336
W następnym rozdziale....337

Rozdział 15. Protokoły przesyłania plików....339

File Transfer Protocol (FTP)....339
Podstawowe polecenia i kody odpowiedzi FTP....341
Zagadnienia bezpieczeństwa FTP....345
Typowa sesja FTP....347
Trivial File Transfer Protocol (TFTP)....348
Formaty komunikatów TFTP....349
Łączenie z serwerem TFTP....350
Oprogramowanie klienta TFTP....352
Typowe zastosowania TFTP....352
Remote Copy Protocol (RCP)....353
HyperText Transfer Protocol (HTTP)....353
HTTP a bezpieczeństwo....355
Network File System (NFS)....357
Remote Procedure Calls (RPC)....358
External Data Representation (XDR)....360
Portmapper....360
Procedury wywoływane przez NFS....360
Metody uwierzytelniania....362
Instalowanie systemu plików....364
Blokowanie plików pod kontrolą NFS....365
Web Distributed Authoring and Versioning (WebDAV)....365
Zabezpieczenia w protokole WebDAV....366
Współpraca EFS i WebDAV....367
Podsumowanie....368
Pytania sprawdzające....368
W następnym rozdziale....369

Rozdział 16. Poczta elektroniczna w sieci TCP/IP....371

Poczta elektroniczna w skrócie....371
Simple Mail Transfer Protocol (SMTP)....373
Proces SMTP....373
Inne wymagania SMTP....374
Zabezpieczanie sesji SMTP....375
Post Office Protocol 3 (POP3)....377
Sesja protokołu POP3....377
Zabezpieczanie uwierzytelniania POP3....379
Internet Message Access Protocol 4 (IMAP4)....380
Atrybuty wiadomości IMAP4....380
Stany sesji IMAP4 i związane z nimi polecenia....381
Zabezpieczanie uwierzytelniania IMAP....384

Lightweight Directory Access Protocol (LDAP) ... 385
Od X.500 do LDAP ... 385
Operacje LDAP ... 386
Zabezpieczanie katalogu LDAP ... 389
Załączniki poczty elektronicznej ... 389
BinHex ... 390
uuencode/uudecode ... 390
Multipurpose Internet Mail Extentions (MIME) ... 391
Przykładowy nagłówek MIME ... 394
Zabezpieczanie poczty elektronicznej ... 395
Protokoły zabezpieczania treści wiadomości ... 395
Podpisy cyfrowe ... 396
Szyfrowanie poczty elektronicznej ... 397
Podsumowanie ... 399
Pytania sprawdzające ... 399
W następnym rozdziale ... 400

Rozdział 17. Zarządzanie siecią — protokół SNMP ... 401

Zarządzanie siecią ... 401
Systemy zarządzania i agenty SNMP ... 402
Wspólnoty SNMP ... 402
Składniki systemu SNMP ... 403
Struktura informacji zarządzania (SMI) ... 404
Protokół SNMP ... 406
Baza informacji zarządzania (MIB) ... 407
Wdrażanie systemu zarządzania SNMP ... 409
Instalowanie agenta SNMP ... 409
Konfigurowanie agenta SNMP ... 410
Korzystanie z konsoli SNMP ... 412
Zarządzanie SNMP w praktyce ... 414
Podsumowanie ... 415
Pytania sprawdzające ... 415
W następnym rozdziale ... 415

Rozdział 18. TCP/IP na łączach telefonicznych ... 417

Przyłączanie do sieci poprzez linię telefoniczną ... 417
Serial Line Internet Protocol (SLIP) ... 418
Point-to-Point Protocol (PPP) ... 419
Połączenia tunelowane ... 422
Point-to-Point Tunneling Protocol (PPTP) ... 423
Layer Two Tunneling Protocol (L2TP) ... 426
Point-to-Point Protocol over Ethernet (PPPoE) ... 431
Uwierzytelnianie klientów PPP ... 434
Dostępne protokoły uwierzytelniania ... 434
Uwierzytelnianie scentralizowane i zdecentralizowane ... 435
Podsumowanie ... 437
Pytania sprawdzające ... 438
W następnym rozdziale ... 438

Rozdział 19. IP w sieciach ATM i bezprzewodowych oraz przesyłanie głosu ... 439

Wykorzystanie TCP/IP w sieci ATM ... 439
Pakiet ATMARP ... 440
Tworzenie logicznych podsieci IP ... 442

Serwer ATMARP ... 443
Rejestrowanie adresu IP na serwerze ATM ... 443
Żądania ATMARP ... 444
Czas utrzymywania wpisów w tabeli ATMARP ... 444
TCP/IP w sieciach bezprzewodowych ... 445
Konfiguracje sieci bezprzewodowych ... 445
Adresowanie MAC ... 446
Zabezpieczenia sieci bezprzewodowych ... 447
Przesyłanie głosu w sieciach IP ... 452
Konwersja głosu na postać cyfrową ... 452
Problemy ograniczające transmisję głosową poprzez IP ... 452
Kierunki rozwoju technologii przesyłania głosu przez IP ... 453
Podsumowanie ... 454
Pytania sprawdzające ... 454
W następnym rozdziale ... 454

Rozdział 20. Konfigurowanie serwerów i stacji roboczych do korzystania z TCP/IP ... 455

Instalowanie TCP/IP w Windows 9x ... 455
Dodawanie karty sieciowej ... 455
Dodawanie stosu protokołu TCP/IP ... 456
Konfigurowanie protokołu TCP/IP ... 457
Konfigurowanie TCP/IP w systemie Windows 2000 ... 461
Konfigurowanie protokołu TCP/IP ... 461
Dalsze ustawienia TCP/IP ... 462
Konfigurowanie TCP/IP w systemie Linux ... 463
Czynności konfiguracyjne w trakcie instalowania systemu ... 463
Modyfikowanie konfiguracji TCP/IP ... 465
Konfigurowanie TCP/IP na serwerze UNIX ... 468
Dołączanie karty sieciowej do stacji systemu UNIX ... 468
Konfigurowanie karty sieciowej ... 469
Przeglądanie plików konfiguracyjnych TCP/IP ... 470
Konfigurowanie demona Internetu ... 471
Konfigurowanie trasowania ... 473
Konfigurowanie TCP/IP na serwerze systemu NetWare 6 ... 474
Instalowanie protokołu TCP/IP ... 474
Weryfikowanie konfiguracji TCP/IP ... 477
Konfigurowanie NetWare/IP ... 478
Konfigurowanie DNS i DSS ... 479
Konfigurowanie serwera DHCP systemu NetWare ... 482
Podsumowanie ... 483
Pytania sprawdzające ... 484
W następnym rozdziale ... 484

Rozdział 21. IPv6, przyszłość TCP/IP? ... 485

Podstawowe zmiany wprowadzane przez IPv6 ... 485
Formaty adresów IPv6 ... 486
Reprezentacje adresów IPv6 ... 487
Specjalne adresy IPv6 ... 488
Format nagłówka IPv6 ... 489
Nagłówki dodatkowe IPv6 ... 490
Nagłówek Hop-by-Hop Options ... 491
Nagłówek Destination Options ... 492

Nagłówek Routing ... 492
Nagłówek Fragment ... 493
Nagłówek opcji Authentication ... 495
Nagłówek ESP ... 496
Nagłówek No Next ... 497
Przejście z IPv4 do IPv6 ... 497
Wykorzystanie dualnej warstwy IP ... 499
Opcje tunelowania IPv6 przez IPv4 ... 500
Transport Relay Translator (TRT) ... 501
Podsumowanie ... 502
Pytania sprawdzające ... 502

Dodatek A Lista dokumentów RFC ... 503

Protokoły grupy Internet Standard ... 503
Protokoły grupy Internet Standard specyficzne dla typu sieci ... 506
Protokoły grupy Draft Standard ... 506
Protokoły grupy Proposed Standard ... 508
Protokoły eksperymentalne ... 528
Protokoły informacyjne ... 532
Protokoły historyczne ... 535
Dokumenty robocze — Internet Draft ... 537

Dodatek B Odpowiedzi na pytania sprawdzające ... 539

Rozdział 1 ... 539
Rozdział 2 ... 540
Rozdział 3 ... 541
Rozdział 4 ... 543
Rozdział 5 ... 545
Rozdział 6 ... 547
Rozdział 7 ... 549
Rozdział 8 ... 551
Rozdział 9 ... 552
Rozdział 10 ... 553
Rozdział 11 ... 554
Rozdział 12 ... 555
Rozdział 13 ... 556
Rozdział 14 ... 558
Rozdział 15 ... 559
Rozdział 16 ... 561
Rozdział 17 ... 563
Rozdział 18 ... 564
Rozdział 19 ... 565
Rozdział 20 ... 566
Rozdział 21 ... 567

Dodatek C Identyfikatory obiektów Internet MIB-II ... 569

Grupa System ... 569
Grupa Interfaces ... 570
Grupa Address Translation ... 572
Grupa IP ... 572
Grupa ICMP ... 575
Grupa TCP ... 577
Grupa UDP ... 578

Grupa EGP 579
Grupa Transmission 580
Grupa SNMP 581

Dodatek D Słowniczek 583

Skorowidz 625

Rozdział 15.

Protokoły przesyłania plików

W niniejszym rozdziale przedstawimy przegląd istotniejszych protokołów przesyłania plików w sieci TCP/IP. Należą do nich:

- *File Transfer Protocol* (FTP),
- *Trivial File Transfer Protocol* (TFTP),
- *Remote Copy Protocol* (RCP),
- *HyperText Transfer Protocol* (HTTP),
- *Network File System* (NFS),
- *Web Distributed Authoring and Versioning* (WebDAV).

Jako uzupełnienie omówienia typowych sesji połączeniowych, zagadnień bezpieczeństwa i charakterystyki protokołów, przedstawione zostaną wybrane, najczęściej obecnie stosowane serwery WWW.

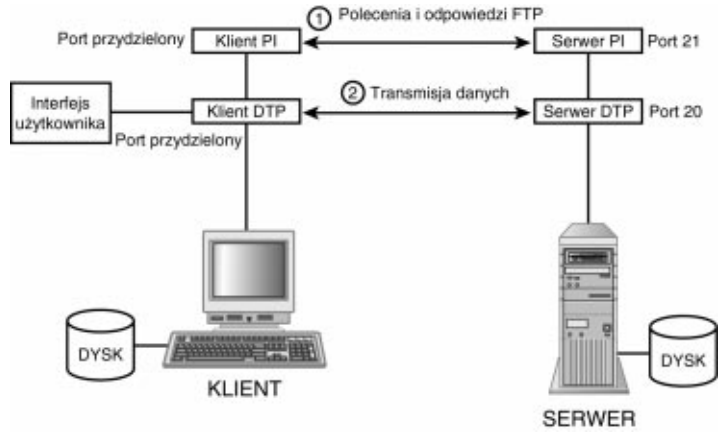
File Transfer Protocol (FTP)

Protokół przesyłania plików jest jednym z najpopularniejszych protokołów wykorzystywanych do przenoszenia plików pomiędzy stacjami w sieci TCP/IP. Główną jego zaletą jest oparcie funkcjonowania na protokole transportu TCP, co zapewnia niezawodne, wymagające ustanowienia sesji połączeniowej przesyłanie.

Protokół FTP wykorzystuje w transmisji danych dwa procesy:

- Proces przesyłania danych (ang. DTP — *Data Transfer Process*) zapewnia faktyczną transmisję danych pomiędzy klientem a serwerem FTP.
- Interpretator protokołu (ang. PI — *Protocol Interpreter*) jest wykorzystywany do przesyłania poleceń pomiędzy klientem a serwerem. Inicjuje on proces FTP i zarządza usługą DTP od strony klienta.

Faktycznie sesja FTP składa się więc z dwóch osobnych sesji łączących klienta z serwerem (patrz rysunek 15.1).



Rysunek 15.1. *Sesja FTP*

W trakcie ustanawiania między klientem i serwerem sesji realizowane są następujące operacje:

1. Na początku ustanawiana jest sesja pomiędzy usługami PI klienta i serwera. Interpretator protokołu po stronie użytkownika inicjuje połączenie sterujące pomiędzy stacjami. Za jego pośrednictwem klient FTP przesyła polecenia do serwera, a serwer potwierdza ich wykonanie. Strona użytkownika korzysta z portu o numerze przydzielonym losowo, natomiast po stronie serwera wykorzystywany jest port TCP 21.
2. Żądanie przesłania danych powoduje zainicjowanie przez proces DTP serwera połączenia z usługą DTP klienta. Za jego pomocą przesyłane są wyłącznie dane.

Połączenie sterujące zostaje zachowane również w trakcie przesyłania danych. Oba połączenia, sterujące i danych, pozwalają na przesyłanie danych w dwóch kierunkach.

Passive FTP

Jest jeszcze jeden sposób inicjowania transmisji FTP. Wcześniej napisaliśmy, że typowe połączenie FTP rozpoczyna inicjowanie przez klienta komunikacji poprzez połączenie sterujące. W drugim kroku serwer FTP inicjuje przesyłanie danych do klienta.

Alternatywą jest *przesyłanie pasywne* (ang. *FTP passive transfer*). W tym przypadku wymiana danych również rozpoczyna się od ustanowienia komunikacji poprzez port TCP 21. Różnice zaczynają się w momencie przesłania przez klienta polecenia `PASV`.

Odpowiedzią serwera FTP na polecenie `PASV` jest przesłanie numeru portu (od 1024 wzwyż), na którym rozpoczyna on nasłuch w oczekiwaniu połączeń klienta. Drugą różnicę wyznacza strona inicjująca połączenie danych — wykorzystując numer portu, otrzymany w odpowiedzi na polecenie `PASV`, klient może otworzyć połączenie danych z serwerem.

Wiele organizacji nie zezwala na przesyłanie pasywne ze względu na wymóg, aby serwer FTP prowadził nasłuch na losowo wybranym porcie o numerze od 1024 wzwyż. Otwieranie takich portów zapory zewnętrznej sieci jest najczęściej zbyt dużym zagrożeniem.

Podstawowe polecenia i kody odpowiedzi FTP

Przedstawimy teraz podstawowe polecenia wykorzystywane w trakcie sesji FTP. Można wśród nich wyróżnić następujące kategorie:

- polecenia sterujące dostępem,
- polecenia sterujące przesyłaniem,
- polecenia przesyłania plików,
- polecenia zarządzania plikami i katalogami,
- polecenia pomocy i kontroli stanu sesji,
- odpowiedzi serwera FTP.

Polecenia sterujące dostępem

Polecenia sterujące dostępem znajdują zastosowanie podczas ustanawiania i kończenia sesji FTP łączącej stacje klienta i serwera. Są one zwykle wprowadzane w standardowej kolejności. Najlepszym przykładem jest polecenie `USER`, po którym powinno nastąpić `PASS` — najpierw podajemy nazwę konta, potem odpowiednie hasło. Listę poleceń przedstawia tabela 15.1.

Tabela 15.1. *Polecenia FTP sterujące dostępem*

Polecenie	Opis
`OPEN [`*`stacja`*`]`	Ustanawia sesję z usługą FTP stacji o nazwie *stacja*
`USER [`*`nazwa`*`]`	Określa użytkownika korzystającego z poleceń FTP. Jest to pierwsze polecenie przesyłane po nawiązaniu połączenia sterującego
`PASS [`*`hasło`*`]`	Polecenie wprowadzane bezpośrednio po `USER`. Służy przekazaniu hasła użytkownika do serwera FTP za pośrednictwem protokołu Telnet. Chociaż zasadniczo klient FTP nie wyświetla hasła na ekranie, standardowa specyfikacja FTP przewiduje przesyłanie go przez sieć w sposób jawny
`ACCT [`*`konto`*`]`	Wymagany przez niektóre serwery FTP opcjonalny parametr określający konto użytkownika na tychże serwerach. Podawana nazwa nie musi mieć związku z poleceniem `USER`
`SMNT`	Polecenie `Structure Mount` (instaluj strukturę) pozwala na zainstalowanie struktury danych odmiennego systemu plików bez ponawiania logowania do serwera
`REIN`	Polecenie `Reinitialize` (inicjuj ponownie) kończy sesję dla bieżącego konta użytkownika. Transmisja wszystkich danych wejściowych i wyjściowych (z wyjątkiem trwającego przesyłania) zostaje wstrzymana. Użytkownik wraca do tego samego punktu, w którym znajdował się w momencie nawiązania połączenia z serwerem
`QUIT`	Kończy sesję łączącą stacje klienta i serwera. Część oprogramowania tak samo interpretuje polecenie `BYE`

Zabezpieczanie haseł przez wprowadzenie protokołu IPSec

Zabezpieczenia IPSec (ang. *IP Security*), omawiane w rozdziale 12., mogą zostać wykorzystane do szyfrowania wszystkich przesyłanych pomiędzy klientem i serwerem danych. Wymaga to zdefiniowania zasady stosowania nagłówków ESP dla wszystkich połączeń z serwerem FTP przez porty TCP 20 i 21.



Anonimowe sesje FTP

Protokół FTP pozwala na nawiązywanie połączeń anonimowych. Wykorzystywaną wówczas nazwą konta jest `ANONYMOUS` lub `FTP`. Zwyczajowo serwer FTP prosi użytkownika korzystającego z takiego konta o podanie swojego adresu e-mail jako hasła.

Polecenia sterujące przesyłaniem

Polecenia określające sposób przesyłania danych pozwalają zmienić ustawienia dotyczące sesji. Ustawienia domyślne pozostają zazwyczaj na serwerze niezmienione. Ich zmiana jest konieczna jedynie wtedy, gdy używane oprogramowanie wymaga wprowadzenia nowszych metod transmisji. Zestawienie poleceń jest przedstawione w tabeli 15.2.

Tabela 15.2. *Polecenia FTP sterujące przesyłaniem*

Polecenie	Opis
`PORT ##`	Pozwala wybrać gniazdo TCP używane po stronie klienta. Parametr ## powinien zawierać 32-bitowy adres IP stacji, z którą nawiązane będzie połączenie DTP oraz 16-bitowy numer portu
`PASV`	Polecenie `Passive` (pasywny) zmienia zachowanie serwera przy ustanawianiu sesji danych. Zamiast inicjowania połączenia serwer będzie jedynie monitorował port danych i oczekiwał na ustanowienie sesji przez klienta
`TYPE`	Polecenie `Representation Type` (typ reprezentacji) określa format reprezentacji danych serwera: ASCII, EBCDIC lub Image
`STRU`	Polecenie `File Structure` (struktura pliku) jest pojedynczym kodem określającym domyślną strukturę plików: `files` (pliki), `records` (rekordy) lub `pages` (strony)
`MODE`	Polecenie `Transfer Mode` (tryb przesyłania) określa tryb transmisji danych: `Stream` (strumień), `Block` (blok) lub `Compressed` (skompresowane)

Polecenia przesyłania plików

Po ustanowieniu sesji z serwerem FTP i określeniu ustawień przesyłania danych można rozpocząć właściwą transmisję. W tabeli 15.3 zamieszczone są najpowszechniej stosowane polecenia przesyłania plików udostępniane przez oprogramowanie klientów FTP.

Polecenia zarządzania plikami i katalogami

Po połączeniu z serwerem FTP, poza samym kopiowaniem plików, potrzebne może być również wykonanie kilku operacji, takich jak zmiana katalogu bieżącego, zmiana nazwy czy utworzenie katalogu. W tabeli 15.4 przedstawione jest ich obszerne zestawienie.

Tabela 15.3. *Polecenia FTP służące do przesyłania plików*

Polecenie	Opis
`ASCII`	Włącza tryb transmisji ASCII, odpowiedni dla plików tekstowych. Jest on używany domyślnie
`BINARY`	Włącza tryb binarny transmisji, który powinien być stosowany do przesyłania wszystkich plików innych niż pliki tekstowe (takich jak graficzne, skompresowane czy wykonywalne)
`TYPE`	Pozwala sprawdzić, który z trybów przesyłania został ustawiony — binarny czy ASCII
`RECV [`*`pzdalny`*`] [`*`plokalny`*`]`	Kopiuje plik *pzdalny* z serwera do pliku lokalnego *plokalny*. Jeżeli nazwa pliku lokalnego nie zostanie podana, plik zostanie skopiowany z zachowaniem nazwy oryginalnej
`SEND [`*`plokalny`*`] [`*`pzdalny`*`]`	Kopiuje lokalny plik *plokalny* do pliku *pzdalny* na serwerze. Jeżeli nazwa pliku na serwerze nie zostanie podana, plik zostanie skopiowany z zachowaniem nazwy lokalnej
`GET [`*`pzdalny`*`] [`*`plokalny`*`]`	Funkcjonuje tak samo jak polecenie `RECV`
`PUT [`*`plokalny`*`] [`*`pzdalny`*`]`	Funkcjonuje tak samo jak polecenie `SEND`
`MGET [`*`pzdalny`*`]`	`Multiple Get` (wielokrotne `GET`) funkcjonuje podobnie jak polecenie `GET`, ale pozwala na kopiowanie większej ilości plików, dzięki możliwości zastosowania do określenia grupy plików symboli wieloznacznych
`MPUT [`*`plokalny`*`]`	`Multiple Put` (wielokrotne `PUT`) funkcjonuje podobnie jak polecenie `PUT`, ale pozwala na kopiowanie większej ilości plików, dzięki możliwości zastosowania symboli wieloznacznych do określenia grupy plików
`PROMPT`	Włącza lub wyłącza wyświetlanie komunikatów z żądaniem potwierdzenia kopiowania po wprowadzeniu poleceń `MGET` i `MPUT`. Gdy są one włączone, pojawia się zapytanie, czy kopiowanie plików ma być kolejno potwierdzane. Gdy są wyłączone, żadne potwierdzenia nie są wymagane

Tabela 15.4. *Polecenia FTP służące do zarządzania plikami i katalogami*

Polecenie	Opis
`DELETE [`*`pzdalny`*`]`	Usuwa plik o nazwie *pzdalny* z serwera FTP
`MDELETE [`*`pzdalny`*`] [...]`	Usuwa wszystkie pliki odpowiadające opartemu na symbolach wieloznacznych wzorcowi *pzdalny*
`LCD [`*`katalog`*`]`	Polecenie `Local Change Directory` (zmień katalog lokalny) zmienia lokalny katalog bieżący. Jego podstawowym zastosowaniem jest określanie katalogu, do którego kopiowane będą pliki z serwera
`CD [`*`katalog`*`]`	Polecenie `Change Directory` (zmień katalog) zmienia katalog bieżący na serwerze
`CDUP`	Polecenie `Change Directory Up` (zmień katalog „do góry”) zmienia katalog bieżący na serwerze na katalog o poziom wyżej. Zostało ono wprowadzone ze względu na zróżnicowanie reprezentacji katalogu nadrzędnego w różnych systemach
`MKDIR katalog`	Polecenie `Make Directory` (utwórz katalog) zakłada nowy katalog na serwerze FTP

Tabela 15.4. *Polecenia FTP służące do zarządzania plikami i katalogami — ciąg dalszy*

Polecenie	Opis
`RMDIR katalog`	Polecenie `Remove Directory` (usuń katalog) usuwa katalog na serwerze FTP
`DIR`	Wyświetla ze szczegółami zawartość bieżącego katalogu serwera. Może ona zostać zapisana do pliku
`LS`	Polecenie `List` wyświetla zawartość bieżącego katalogu serwera w formie skróconej. Podstawowe przełączniki tego polecenia to `-F` i `-all`. Pierwszy powoduje wyświetlanie nazw podkatalogów ze znakiem łamania (`/`) na końcu. Użycie drugiego prowadzi do otrzymania takiego wyniku, jak po wprowadzeniu polecenia `DIR`
`PWD`	Wyświetla nazwę bieżącego katalogu serwera
`RENAME [`*`nazwa1`*`] [`*`nazwa2`*`]`	Zmienia nazwę pliku na serwerze z *`nazwa1`* na *`nazwa2`*

Polecenia pomocy i kontroli stanu sesji

Polecenia pomocy pozwalają uzyskać informacje o składni poleceń FTP, gdy nie jesteśmy pewni, jakie parametry możemy zastosować. Polecenia kontroli stanu sesji zapewniają możliwość sprawdzenia, jakie opcje sesji FTP zostały w danym momencie ustawione. Tabela 15.5 przedstawia typowe polecenia tego rodzaju.

Tabela 15.5. *Polecenia FTP związane z pomocą i kontrolą stanu sesji*

Polecenie	Opis
`!`	Powoduje wyjście z sesji FTP do lokalnego interpretatora poleceń lub powłoki systemowej. Powrót do sesji zapewnia zazwyczaj wprowadzenie polecenia `EXIT`
`?`	Wyświetla wszystkie polecenia dostępne w programie klienta FTP
`HELP`	Funkcjonuje tak samo jak polecenie `?`
`STATUS`	Wyświetla informacje o bieżącym stanie sesji FTP. Obejmują one tryb transmisji, stan połączenia, ustawienia wyświetlania komunikatów i wartość limitu czasu
`VERBOSE`	Przełącza tryb wyświetlania informacji (tzw. tryb informacji pełnej). Gdy jest włączony, użytkownik otrzymuje informacje o wszystkich odpowiedziach serwera FTP oraz szybkościach przesyłania

Kody odpowiedzi serwera FTP

Odpowiedzią serwera FTP na otrzymane od klienta polecenie jest kod odpowiedzi informujący, czy i jak zostało ono wykonane. W przypadku niewykonania polecenia odpowiedni kod błędu wskazuje, jakie działania powinien podjąć klient.

Wszystkie kody odpowiedzi są złożone z trzech pozycji alfanumerycznych. Pierwsza z nich zapewnia informację ogólną — zasadą jest tu wskazanie kolejnej właściwej w określonej sytuacji czynności. W tabeli 15.6 zestawionych jest 5 dopuszczalnych wartości pierwszej pozycji kodu odpowiedzi.

Tabela 15.6. *Wartości pierwszej pozycji kodu odpowiedzi serwera FTP*

Wartość	Nazwa	Opis
1yz	Pozytywna odpowiedź wstępna	Żądana akcja została podjęta. Kolejna odpowiedź zostanie wysłana przed osiągnięciem gotowości do przyjęcia kolejnego polecenia. Jednorazowo może zostać wysłana tylko jedna odpowiedź wstępna
2yz	Pozytywna odpowiedź końcowa	Żądana akcja została poprawnie wykonana i serwer gotów jest na przyjęcie od klienta kolejnego polecenia
3yz	Pozytywna odpowiedź pośrednia	Serwer zaakceptował polecenie, ale oczekiwane są dalsze dane. Jest to typowa odpowiedź, gdy oczekiwane jest określone następstwo poleceń, np. PASS po USER
4yz	Negatywna odpowiedź tymczasowa	Wydane serwerowi polecenie nie zostało poprawnie wykonane. Jest to jednak tymczasowy błąd w połączeniu i polecenie powinno zostać przesłane ponownie
5yz	Negatywna odpowiedź trwała	Żądana akcja nie mogła zostać wykonana. Typowe przyczyny takiej odpowiedzi to błąd we wpisywaniu polecenia oraz brak przypisania użytkownikowi odpowiednich uprawnień

Dalsze uszczegółowienie odpowiedzi serwera przynosi druga pozycja przesyłanego do klienta kodu. Dopuszczalne jej wartości przedstawia tabela 15.7.

Tabela 15.7. *Wartości drugiej pozycji kodu odpowiedzi serwera FTP*

Wartość	Nazwa	Opis
x0z	Składnia	Błąd wynika z niewłaściwej składni albo też z wydania poprawnego polecenia, ale w nieodpowiednim momencie
x1z	Dane	Kod wykorzystywany w odpowiedziach na polecenia pomocy i kontroli stanu sesji, jak HELP i STATUS
x2z	Połączenia	Kod używany w odpowiedziach odnoszących się do stanu połączenia danych
x3z	Uwierzytelnianie i rejestracja	Kod wykorzystywany podczas procesu uwierzytelniania użytkownika
x4z	Nieokreślony	Kod jeszcze nieużywany
x5z	System plików	Informacja o stanie systemu plików serwera lub klienta FTP

Trzecia pozycja kodu odpowiedzi zapewnia dalsze sprecyzowanie podawanej informacji — ściśle zależy ona od dwóch pierwszych. Zaleca się przyjęcie ogólnej zasady dołączania do kodów błędów pola tekstowego, co zapewni pełne objaśnienie znaczenia kodu odpowiedzi. Szczegółowe zestawienie wartości wykorzystywanych na trzeciej pozycji kodu znaleźć można w dokumencie RFC 959, dostępnym pod adresem *www.ietf.org/rfc*.

Zagadnienia bezpieczeństwa FTP

RFC 2228 Dokument RFC 2228 nakreśla potrzebę powstania standardowego mechanizmu zabezpieczającego dla protokołu FTP. Obecnie hasła wykorzystywane w sesjach FTP są

przesyłane tekstem jawnym. Umożliwia to przechwytywanie ich przez tzw. szperacze sieciowe (ang. *network sniffers*).

Mechanizmy zabezpieczające powinny również zapewniać uwierzytelnianie serwerów dla uniknięcia podszywania się oraz szyfrowanie informacji przesyłanych kanałem danych.

RFC 2228 wprowadza dodatkowe polecenia, które mogą być używane w systemie FTP opcjonalnie. Ich zestawienie przedstawia tabela 15.8.

Tabela 15.8. *Polecenia rozszerzeń bezpieczeństwa FTP*

Polecenie	Pełna nazwa	Opis
AUTH	Mechanizm uwierzytelniania/ zabezpieczenia	Żądanie klienta dotyczące stosowania określonego mechanizmu ochrony przesyłanych danych. Może być przesyłane kilkakrotnie do czasu wynegocjowania między klientem a serwerem metody uwierzytelniania
ADAT	Dane uwierzytelniania/ zabezpieczeń	Polecenie pozwalające przesłać dodatkowe dane określające opcjonalne elementy wybranego mechanizmu zabezpieczeń
PROT	Poziom ochrony kanału danych	Określa poziom ochrony, który będzie stosowany przez stacje klienta i serwera w odniesieniu do kanału danych. Ustawienie Clear (jawny) wskazuje na zwykłe przesyłanie danych, Safe (bezpieczny) — sprawdzanie integralności danych, Confidential (poufny) — szyfrowanie, Private (prywatny) — jednoczesne szyfrowanie i sprawdzanie integralności
PBSZ	Rozmiar bufora ochrony	Określa największy dopuszczalny rozmiar kodowanych bloków danych, które będą przesyłane podczas wymiany plików. Podawany w bajtach
CCC	Kanał poleceń jawnych	Kończy wykorzystywanie określonego mechanizmu zabezpieczającego. Wykorzystywane najczęściej, gdy w sieci stosowane są zabezpieczenia na poziomie protokołu TCP. Po zakończeniu uwierzytelniania połączenia polecenie CCC umożliwia zakończenie dalszych procedur sprawdzania integralności danych i uwierzytelniania
MIC	Polecenie ochrony integralności	Wykorzystywane do transmisji danych przy poziomie ochrony ustawionym na Safe
CONF	Polecenie ochrony poufności	Wykorzystywane do transmisji danych przy poziomie ochrony ustawionym na Confidential.
ENC	Polecenie ochrony prywatności	Wykorzystywane do transmisji danych przy poziomie ochrony ustawionym na Private

Korzystanie z mechanizmów zabezpieczających inicjuje klient FTP, podając przy użyciu polecenia AUTH, z jakiego mechanizmu bezpieczeństwa zamierza korzystać. Serwer akceptuje ten mechanizm, odmawia jego stosowania lub też całkowicie odrzuca polecenie (gdy nie posiada mechanizmów zabezpieczeń FTP).

Jeżeli korzystanie z zabezpieczeń wymaga informacji dodatkowych, serwer żąda od klienta FTP użycia polecenia ADAT w celu ich przesłania. Wymiana informacji trwa do momentu określenia przyporządkowania zabezpieczeń.

Po ustanowieniu bezpiecznego połączenia może zostać określony typ ochrony danych. Służy do tego polecenie PROT. W zależności od wybranego poziomu, do realizacji bezpiecznej transmisji kanałem danych wykorzystywane będzie polecenie MIC, CONF lub ENC. Rozmiar pakietów danych określa się poleceniem PBSZ.

Nie każdy RFC jest wykorzystywany

Mimo że istnieją już dokumenty RFC dotyczące zabezpieczeń uwierzytelniania FTP, stosunkowo niewielka liczba klientów zapewnia obsługę nowych funkcji. Zalecić można używanie zabezpieczeń IPSec do szyfrowania całości sesji FTP lub wprowadzenie uwierzytelniania anonimowego, co wykluczy przesyłanie haseł tekstem jawnym.



Jeżeli stosujemy uwierzytelnianie anonimowe, wykluczone powinno zostać przechowywanie na serwerze FTP wszelkich istotnych danych. Podobna zasada dotyczy praw zapisu na serwerze. Anonimowy serwer FTP służyć powinien wyłącznie do udostępniania dokumentów publicznych.

Typowa sesja FTP

Poniżej przedstawiamy typową sesję FTP. Dla lepszego uświadomienia sobie źródła wyświetlanych informacji wszystkie wiersze zostały poprzedzone informacją KLIENT> lub SERWER> (znaki wpisywane przez użytkownika oznaczamy tłustym drukiem).

```
KLIENT> D:\>ftp
KLIENT> ftp> open bkdata
SERWER> Połączony z bkdata.
SERWER> 220 bkdata Microsoft FTP Service (Version 5.0).
KLIENT> Użytkownik (bkdata.przyklad.com:(none)): ftp
SERWER> 331 Anonymous access allowed, send identity (e-mail name) as
password.
KLIENT> Hasło:bkomar@komarconsulting.com
SERWER> 230-Witamy na serwerze FTP BKDATAb
SERWER> ================================
SERWER>  Dostęp anonimowy dozwolony!

SERWER> 230 Anonymous user logged in.
KLIENT> ftp> ls -F
SERWER> 200 PORT command successful.
SERWER> 150 Opening ASCII mode data connection for /bin/ls.
SERWER> lanma256.bmp
SERWER> Telnet Utilities/
SERWER> 226 Transfer complete.
SERWER> 33 bytes received in 0.01 seconds (3.30 Kbytes/sec)
KLIENT> ftp> cd "Telnet Utilities"
SERWER> 250 CWD command successful.
KLIENT> ftp> ls
SERWER> 200 PORT command successful.
SERWER> 150 Opening ASCII mode data connection for file list.
SERWER> teld4_x86_beta.exe
SERWER> telftp32.exe
```

```
SERWER> 226 Transfer complete.
SERWER> 34 bytes received in 0.01 seconds (3.40 Kbytes/sec)
KLIENT> ftp> binary
SERWER> 200 Type set to I.
KLIENT> ftp> get telftp32.exe
SERWER> 200 PORT command successful.
SERWER> 150 Opening binary mode data connection for telftp32.exe(743758
bytes).
SERWER> 226 Transfer complete.
SERWER> 743758 bytes received in 3.11 seconds (238.84 Kbytes/sec)
KLIENT> ftp> bye
SERWER> 221 Dziękujemy za wizytę!
```

Jak widać na przykładzie, zakres faktycznie wykorzystywanych poleceń ogranicza się do tych, których potrzebujemy, aby ściągnąć plik. Połączyliśmy się z serwerem *bkdata.example.com*. Wykorzystaliśmy możliwość uwierzytelnienia anonimowego, podając jako nazwę użytkownika `ftp`. Po ustanowieniu sesji użyliśmy polecenia `ls -F`, aby wyświetlić zawartość bieżącego katalogu serwera FTP. Skorzystanie z opcji `-F` pozwoliło odróżnić podkatalogi od plików. Po zmianie katalogu na *Telnet Utilities* ściągnięty został plik *telftp32.exe* w binarnym trybie transmisji.

Trivial File Transfer Protocol (TFTP)

Prosty protokół przesyłania plików (ang. TFTP — *Trivial File Transfer Protocol*) pozwala przesyłać pliki pomiędzy stacjami, wykorzystując protokół UDP. Nie zapewnia on tak wielu możliwości jak FTP, ale potrafi być skutecznym narzędziem odczytywania i zapisywania plików na innej stacji w sieci TCP/IP. Protokół TFTP jest często stosowany do pobierania kodu inicjalizacyjnego drukarek, koncentratorów i routerów. Implementacji TFTP używają również bezdyskowe stacje będące klientami DHCP lub BOOTP. Posiadają one jedynie informację o serwerze TFTP, z którego mogą pobrać plik zawierający odpowiednią procedurę inicjalizacji. Dość nowym przykładem zastosowania TFTP są usługi instalacji zdalnej (ang. RIS — *Remote Installation Services*) firmy Microsoft. Po zainicjowaniu komunikacji przez kartę sieciową wykorzystują one protokół TFTP do przesłania do komputera klienckiego danych niezbędnych do jego uruchomienia i zainstalowania systemu Windows.

Przesyłanie danych przy wykorzystaniu protokołu TFTP jest oparte na pakietach o stałej długości 512 bajtów. Jeżeli pakiet jest krótszy, oznacza to, że jest to ostatni pakiet transmisji. Po wysłaniu każdego pakietu do stacji docelowej, jego kopia jest przechowywana w przeznaczonym do tego buforze, aż do czasu otrzymania potwierdzenia poprawnego odebrania danych. Jeżeli stacja wysyłająca nie otrzyma potwierdzenia przed upływem limitu czasu retransmisji, pakiet zostaje wysłany ponownie. Daje to pewien dodatkowy poziom zabezpieczenia strumienia danych ponad właściwą sumę kontrolną UDP. Dla zapewnienia niezawodności transmisji wymagane jest przechowanie jedynie ostatniego pakietu — odbiór wcześniejszych musi zostać potwierdzony jeszcze przed rozpoczęciem przesyłania następnych.

Formaty komunikatów TFTP

Podczas sesji TFTP przesyłanych jest pięć typów komunikatów:

- RRQ — żądanie odczytu,
- WRQ — żądanie zapisu,
- DATA — dane,
- ACK — potwierdzenie,
- ERR — błąd.

Żądania zapisu i odczytu korzystają z tego samego formatu, przedstawionego na rysunku 15.2.

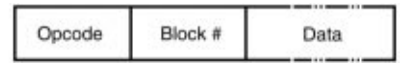
Rysunek 15.2. *Format pakietów RRQ i WRQ*



- Pole Opcode (kod operacji) zawiera wartość 1 przy żądaniu odczytu, a 2 przy żądaniu zapisu.
- Pole Filename (nazwa pliku) określa nazwę pliku pobieranego lub przekazywanego do serwera TFTP. Długość pola jest zmienna, a nazwa — zapisywana znakami NETASCII; znak o kodzie zero sygnalizuje koniec pola nazwy pliku.
- Pole Mode (tryb) określa typ transmisji. Do wartości dopuszczalnych należą przede wszystkim NETASCII i OCTET. Pierwszy z tych trybów jest wykorzystywany do przesyłania dokumentów tekstowych. Jest to 8-bitowy format ASCII, którego używa również protokół Telnet. Typ OCTET jest wykorzystywany do przesyłania danych binarnych przy użyciu pełnych 8 bitów. Znak o kodzie zero sygnalizuje koniec pola Mode.

Właściwe dane są przesyłane komunikatem DATA (patrz rysunek 15.3). Trafia do niego zawartość wskazanego przez pliku użytkownika.

Rysunek 15.3. *Komunikat DATA*



- Pole Opcode (kod operacji) jest ustawiane na wartość 3 — oznacza to komunikat zawierający dane.
- Pole Block # (numer bloku) dla pierwszego pakietu jest ustawiane na 1, a dla każdego kolejnego wartość ta jest zwiększana o 1.
- Pole Data (dane) może mieć do 512 bajtów długości. Jeżeli jest krótsze, to znaczy, że blok jest ostatnią częścią pliku. Jeżeli pole to ma 512 bajtów, oznacza to, że przed zakończeniem transmisji muszą zostać przesłane kolejne bloki danych.

Rysunek 15.4 przedstawia pakiet ACK. Odpowiedź w postaci pakietu ACK (potwierdzenia) lub pakietu ERR (informacji o błędzie) jest zwracana po odebraniu każdego pakietu danych lub żądania zapisu.

Rysunek 15.4.
Pakiet ACK



Pakiet ACK ma dwa pola:

- Pole Opcode (kod operacji) jest ustawiane na wartość 3.
- Pole Block # (numer bloku) zawiera numer bloku odebrany w pakiecie, którego odbiór jest potwierdzany. Jeżeli potwierdzenie dotyczy pakietu WRQ, pole jest ustawiane na wartość 0, co sygnalizuje gotowość o transmisji danych.

Rysunek 15.5 przedstawia format komunikatu TFTP ERR.

Rysunek 15.5.
Pakiet ERR



- Pole Opcode (kod operacji) jest ustawiane na wartość 5.
- Wartość Error Code odpowiada jednej z przedstawionych w tabeli 15.9.

Tabela 15.9. *Kody błędów TFTP*

Kod błędu	Opis
0	Błąd niezdefiniowany. Więcej informacji o jego przyczynie można znaleźć w polu Error Message
1	Nie odnaleziono pliku. Została podana niewłaściwa nazwa
2	Dostęp zabroniony. Niewystarczające uprawnienia dostępu spowodowały błąd w transmisji danych
3	Brak miejsca na dysku lub możliwości alokacji przekroczone
4	Niedozwolona operacja TFTP
5	Nieznany identyfikator transmisji
6	Istnieje plik o podanej nazwie
7	Nie ma takiego użytkownika

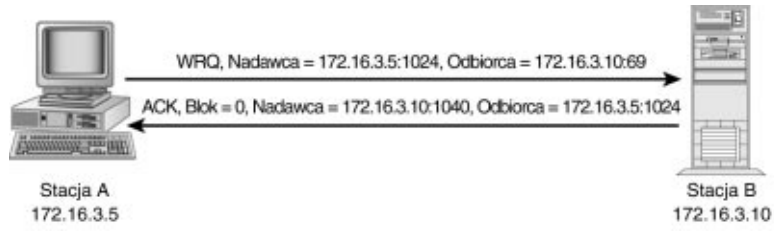
- Pole Error Message jest zapisywane w formacie NETASCII i uzupełnia kod błędu opisem tekstowym, użytecznym przy rozwiązywaniu problemu.
- Wartość 0. Pole Error Message ma zmienną długość i zawsze zakończone jest bajtem zerowym.

Łączenie z serwerem TFTP

Kiedy klient łączy się z serwerem TFTP, powtarzają się dwie podstawowe transakcje. Klient albo przekazuje dane do serwera, korzystając z żądania zapisu, albo pobiera dane z serwera, korzystając z żądania odczytu.

Rysunek 15.6 ilustruje przebieg komunikacji, kiedy do serwera przesyłane jest żądanie zapisu.

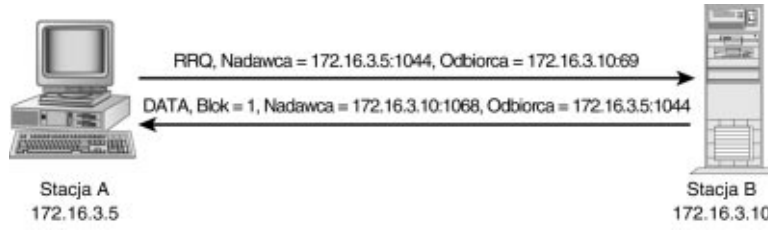
Rysunek 15.6. *Żądanie zapisu na serwerze TFTP*



Stacja wysyłająca (Stacja A) wysyła komunikat TFTP z kodem operacji ustawionym na 2 dla zasygnalizowania żądania zapisu. Portem źródłowym (zapisanym w komunikacie) będzie wybrany losowo port o numerze wyższym niż 1024. Portem przeznaczenia żądania zapisu jest zawsze port UDP o numerze 69. Serwer TFTP (Stacja B) odpowiada komunikatem potwierdzającym. Numerem bloku w tym komunikacie będzie 0. Informuje to o gotowości serwera TFTP do odbierania danych od stacji wysyłającej. Losowo wybrany port źródłowy komunikatu potwierdzającego będzie odtąd wykorzystywany do odbierania przychodzącej transmisji danych. Ta zmiana portu nasłuchu jest znaczną przeszkodą dla systemów firewall przy klasyfikowniu poprawności odpowiedzi na wcześniejsze żądanie TFTP. Dane będą przesyłane w pakietach 512-bajtowych. Pakiet o mniejszej długości będzie pakietem ostatnim.

Rysunek 15.7 przedstawia przebieg komunikacji, kiedy klient pobiera plik z serwera TFTP.

Rysunek 15.7. *Żądanie odczytu z serwera TFTP*



Stacja A przesyła komunikat żądania odczytu do stacji B. Kod operacji jest ustawiany na wartość 1, co sygnalizuje, że jest to komunikat `RRQ`. Portem źródłowym (zapisanym w komunikacie) jest losowo wybrany port UDP. Będzie on wykorzystywany do przesyłania danych. W tym przypadku jest to 1044. Standardowo, żądanie odczytu będzie przesyłane zawsze do portu o numerze 69 serwera. Serwer odpowiada pakietem `DATA`. Pierwszy taki pakiet będzie opisany numerem bloku 1 i będzie zawierał 512 bajtów danych. W pakiecie tym serwer podaje identyfikator swojego portu protokołu transportowego. Każdy kolejny pakiet danych będzie zawierał 512 bajtów danych i numer bloku o 1 wyższy od poprzedniego. Wielkość ostatniego pakietu będzie zazwyczaj mniejsza niż 512 bajtów.

Oprogramowanie klienta TFTP

W większości powłok systemu UNIX i systemów Windows narzędzie `TFTP` ma formę polecenia tekstowego pozwalającego na przesłanie danych do lub z serwera TFTP. Składnia polecenia jest następująca:

```
TFTP [-i] stacja [GET | PUT] źródło [miejsce_docelowe]
```

- `-i`. Użycie opcjonalnego przełącznika `-i` ustawia tryb przesyłania danych na `OCTET`, czyli transmisję binarną. Gdy nie przełącznik zostanie użyty, domyślnym trybem przesyłania jest `NETASCII`, odpowiedni dla plików tekstowych.
- *`stacja`*. Nazwa stacji serwera TFTP.
- `GET`. Transakcja polegająca na skopiowaniu pliku z serwera TFTP do stacji, na której zostało uruchomione oprogramowanie klienta.
- `PUT`. Transakcja polegająca na skopiowaniu pliku ze stacji, na której zostało uruchomione oprogramowanie klienta, do serwera TFTP.
- *`źródło`*. Nazwa pliku, który ma zostać pobrany z lub przekazany do serwera TFTP.
- *`miejsce_docelowe`*. Nazwa, która ma zostać użyta do zapisania pliku po jego przesłaniu. Często określa się ją jako *docelową nazwę pliku*.

Typowe zastosowania TFTP

Protokół TFTP nie jest zbyt często używany do „zwykłego" przesyłania danych. Jego najpowszechniejszym zastosowaniem jest pobieranie informacji konfiguracyjnych stacji. Dwie podstawowe sytuacje tego rodzaju to konfigurowanie routerów oraz konfigurowanie stacji BOOTP.

Routery mogą przechowywać parametry konfiguracyjne na serwerze TFTP. Jest to jedna z metod zapobiegania awariom i ich następstwom. W przypadku wyłączenia routera z użytkowania jego właściwe ustawienia mogą zostać pobrane z serwera TFTP przez router zastępczy (lub ten sam po wykonaniu naprawy).



Warto zadbać o to, żeby router zastępczy korzystał z tego samego oprogramowania systemowego i w miarę możliwości był nawet tą samą wersją routera. W przypadku routerów Cisco, każde nowe oprogramowanie systemowe routera pozwala na korzystanie z dodatkowych parametrów konfiguracyjnych.

Protokół BOOTP przewiduje odniesienie do serwera TFTP, który zawiera pliki konfiguracyjne klienta BOOTP. Po otrzymaniu z serwera BOOTP początkowej konfiguracji IP następuje kontakt z serwerem TFTP w celu pobrania systemu operacyjnego. Jest to wyjątkowo efektywna metoda przesyłania danych do klientów BOOTP. Na jej wykorzystanie pozwalają również niektóre wersje oprogramowania DHCP. Najlepszym przykładem połączonego zastosowania protokołów TFTP i DHCP jest usługa RIS firmy Microsoft służąca do instalowania systemów operacyjnych klientów — Windows 2000 Professional i Windows XP Professional.

Remote Copy Protocol (RCP)

Protokół zdalnego kopiowania (ang. RCP — *Remote Copy Protocol*) jest częścią pakietu *R-Utilities*. Pozwala on na kopiowanie katalogów i ich zawartości ze stacji odległej i na nią.

Podobnie jak w przypadku opisanych w poprzednim rozdziale narzędzi `RSH` i `RLOGIN`, polecenie `RCP` wykorzystuje pliki `hosts.equiv` i `.rhosts` do określenia, które stacje i którzy użytkownicy mają prawo do korzystania z polecenia.

Składnia polecenia `RCP` jest następująca:

```
RCP [-r] stacja1:plik1 stacja2:plik2
```

- `-r`. Opcja pozwalająca na kopiowanie z podkatalogami.
- `stacja1:plik1`. Plik, który będzie kopiowany. Jeżeli stacją źródłową jest komputer lokalny, wystarczy podać katalog i nazwę pliku.
- `stacja2:plik2`. Docelowy plik i katalog kopiowania.

Aby skopiować plik *budzet.txt* z lokalnego katalogu */usr/bkomar/raporty* do katalogu */usr/dneilan* na zdalnej stacji *DNEILAN*, należy wprowadzić następujące polecenie:

```
RCP /usr/bkomar/raporty/budzet.txt dneilan:/usr/dneilan
```

Podobnie, aby skopiować zawartość całego katalogu */data/ksiegowosc* do katalogu */data/budzet* na zdalnej stacji *IRONMAN*, użyjemy polecenia:

```
RCP -r /data/ksiegowosc ironman:/data/budzet
```

Jeżeli z kolei zamierzamy skopiować katalog */data/marketing* na stacji *DNEILAN* i wszystkie w nim zawarte podkatalogi do katalogu */data/marketing/2002* na stacji *IRONMAN*, polecenie będzie wyglądać następująco:

```
RCP -r dneilan:/data/marketing ironman:/data/marketing/2002
```

HyperText Transfer Protocol (HTTP)

RFC 2616

Protokół przesyłania hipertekstu (ang. HTTP — *HyperText Transfer Protocol*) jest protokołem wykorzystywanym do przeglądania WWW (ang. *World Wide Web* — „ogólnoświatowa pajęczyna"). Jest on oparty na mechanizmie pytanie-odpowiedź. Klient żąda przesłania strony z serwera WWW, odpowiedzią serwera jest przesłanie tejże strony.

Protokół HTTP pracuje w warstwie aplikacji. Klient wysyła zapytanie do serwera HTTP (standardowo do portu TCP 80). Serwer interpretuje zapytanie i wysyła właściwą odpowiedź. Faktyczny przebieg komunikacji ma naturę bezpołączeniową i bezstanową. Po udzieleniu odpowiedzi na zapytanie klienta połączenie jest zrywane do czasu otrzymania następnego zapytania. Wyjątkiem od tej zasady jest korzystanie przez klienta z mechanizmów utrzymywania połączenia przez klienta (ang. *keep-alive*) zapewnianych

przez HTTP 1.1. W tym przypadku klient nie ustanawia nowych sesji, ale utrzymuje istniejące połączenie.

W zapytaniach HTTP wykorzystuje się kilka metod. Należą do nich:

- GET,
- HEAD,
- POST,
- PUT,
- DELETE,
- TRACE,
- CONNECT.

Metoda GET wykorzystywana jest w zapytaniu HTTP w celu uzyskania określonej informacji. Pozwala ona na pewną elastyczność, opartą na instrukcjach IF. Ich wykorzystanie prowadzi do tzw. warunkowego GET. Jeżeli podany warunek jest spełniony, dane są przesyłane. Pozwala to klientom HTTP używać buforowanych kopii stron WWW, jeżeli nie zmieniły one swojej zawartości. Przyczynia się to do lepszego wykorzystywania pasma przesyłania sieci.

Metoda HEAD funkcjonuje podobnie jak GET z tym wyjątkiem, że treść przesyłki nie jest zwracana do klienta. Jest to więc metoda często stosowana do określania, czy łącze do określonego zasobu jest poprawne i czy zasób ten nie uległ zmianie. Wykrywanie zmian jest oparte na porównaniu danych wysłanych w nagłówku zapytania z zawartością nagłówka odpowiedzi.

Metoda POST pozwala zażądać od serwera przyjęcia załączonych danych jako nowej, skierowanej do niego publikacji. Można ją wykorzystać do wysyłania listów do grupy dyskusyjnej, odsyłania wypełnionych formularzy HTML serwerowi HTTP albo dodawania rekordów danych do bazy przechowywanej na serwerze.

Metodę PUT wykorzystuje się do umieszczania wysyłanych danych w ściśle określonym w zapytaniu miejscu. Różni się ona od metody POST możliwością określenia docelowej lokalizacji danych. Jeżeli dane już się w niej znajdują, nowa przesyłka powinna być traktowana jako aktualizacja danych istniejących.

Metoda DELETE jest żądaniem usunięcia przez serwer określonych zasobów. Czynność tę może powstrzymać interwencja administratora lub odpowiednie zabezpieczenie serwera. Odpowiedź pozytywna będzie przesłana jedynie wówczas, gdy serwer HTTP faktycznie może usunąć dane.

Metoda TRACE pozwala na skontrolowanie poprawności odbierania danych przez serwer. Odpowiedź TRACE jest taka sama jak odebrane przez serwer zapytanie. Pozwala to na testowanie zapytań i rozwiązywanie problemów ich dotyczących.

Metoda CONNECT jest zarezerwowana do użytku przez mechanizm zabezpieczeń TSL (ang. *Transport Layer Security*).

Obecnym standardem protokołu HTTP jest HTTP 1.1. Do nowych cech tej wersji należą:

- *Trwałe połączenia*. HTTP w wersji 1.1 pozwala na obsługę wielu zapytań w jednym połączeniu. Wcześniejsze wersje wymagały ustanawiania osobnych połączeń dla każdej z grafik osadzonych na stronie WWW.
- *Przetwarzanie potokowe*. Dozwolone jest przesyłanie do serwera kolejnych zapytań jeszcze przed otrzymaniem odpowiedzi na zapytanie początkowe. Pozwala to na zwiększenie wydajności.
- *Dyrektywy buforowania*. Wprowadzenie dyrektyw buforowania pozwala na ignorowanie i optymalizowanie domyślnych algorytmów zarówno serwera, jak i klienta.
- *Nagłówki stacji*. Ta opcja HTTP 1.1 pozwala na wiązanie wielu nazw stacji z jednym adresem IP. Pozwala to uniknąć przypisywania kilku różnych adresów IP serwerowi WWW obsługującemu kilka serwerów wirtualnych. Nagłówek stacji pozwala określić, do którego wirtualnego serwera zapytanie jest skierowane.
- *Opcje* `PUT` i `DELETE`. Polecenia pozwalające zdalnemu administratorowi wysyłać i usuwać zawartość serwera WWW przy użyciu standardowej przeglądarki.
- *Przekierowania HTTP*. Mechanizm pozwalający administratorowi na skierowanie użytkownika do innej strony lub ośrodka WWW, gdy strona staje się niedostępna lub zostaje usunięta.



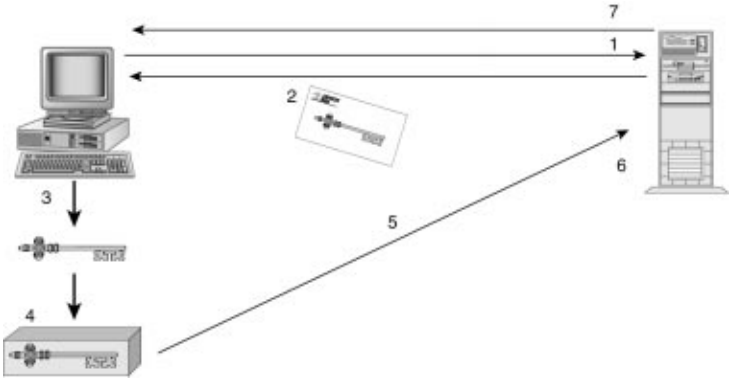
Obecnie stosuje się powszechnie trzy serwery HTTP. Najpopularniejszy jest *APACHE* Web server. Pracuje on w większości systemów UNIX i Linux oraz na platformach Windows. Pozostałe dwa to *SuiteSpot* firmy Netscape i *Internet Information Server* (IIS) firmy Microsoft. Netscape SuiteSpot pracuje na większości platform, w tym Windows NT, Linux i SCO UNIX. Microsoft IIS pracuje wyłącznie na platformach Windows NT, Windows 2000 Server i Windows .NET Server.

HTTP a bezpieczeństwo

RFC 2246 Do zabezpieczenia informacji przesyłanych połączeniami HTTP może zostać zastosowanych kilka metod. Dwie najpopularniejsze są oparte na *szyfrowaniu* transmisji danych i *uwierzytelnianiu* zarówno klienta, jak i serwera. Noszą one nazwy: *SSL* (ang. *Secure Sockets Layer* — warstwa gniazd bezpiecznych) i *TSL* (ang. *Transport Layer Security* — zabezpieczenia warstwy transportu).

Zarówno SSL, jak i TLS funkcjonują jako dodatkowa warstwa pomiędzy warstwą transportową TCP/IP, a warstwą aplikacji. Wszystkie transmisje między klientem a serwerem podlegają szyfrowaniu i deszyfrowaniu przez warstwę SSL lub TLS. Główna różnica między oboma protokołami polega na tym, że szeroko stosowany protokół SSL nie został zdefiniowany w dokumencie RFC. Protokół TLS został opisany w dokumencie nr 2246, a jego interakcje z protokołem HTTP 1.1 w dokumentach RFC 2817 i 2818.

Szyfrowanie SSL i TLS opiera się na cyfrowych certyfikatach SSL. Rysunek 15.8 ilustruje typowy przebieg komunikacji SSL.



Rysunek 15.8. *Proces wymiany potwierdzeń SSL*

1. Klient przesyła serwerowi żądanie przyznania zabezpieczeń.
2. Serwer przesyła klientowi jego certyfikat. Zawiera on klucz publiczny klienta oraz opcje szyfru.



W tym momencie, po otrzymaniu certyfikatu serwera, klient może zgłosić błąd. Ma on najczęściej jedną z dwóch przyczyn:

Pierwszą jest niezgodność nazwy podmiotu certyfikatu z pełną, kwalifikowaną nazwą serwera WWW. Tak dzieje się wtedy, gdy organizacja przeniesie witrynę WWW na inny serwer, o odmiennej nazwie.

Drugą przyczyną odmowy komunikacji może być pochodzenie komunikatu ze źródła nie wymienionego na przechowywanej przez klienta liście zaufanych wystawców certyfikatów. Aby certyfikat został uznany, musi istnieć nadrzędny organ certyfikacyjny, który zarówno klient, jak i serwer uznają za zaufany.

3. Klient generuje klucz główny.
4. Klient szyfruje klucz główny przy użyciu klucza publicznego serwera otrzymanego w kroku 2. Tak zabezpieczony klucz główny klienta może zostać odszyfrowany wyłącznie przy użyciu klucza prywatnego obecnego tylko na serwerze.
5. Klient przesyła zaszyfrowany klucz główny do serwera.
6. Serwer odszyfrowuje klucz główny klienta przy użyciu własnego klucza prywatnego.
7. Serwer uwierzytelnia się u klienta, przesyłając komunikat zaszyfrowany przy użyciu klucza głównego klienta. Obie strony połączenia zostają w ten sposób uwierzytelnione.

Klucz publiczny i klucz prywatny

Wiele mechanizmów szyfrowania korzysta z systemu klucza prywatnego i klucza publicznego. Pakiet zaszyfrowany przy użyciu klucza prywatnego może zostać odszyfrowany jedynie przy użyciu odpowiadającego mu klucza publicznego. Podobnie pakiet zaszyfrowany przy użyciu klucza publicznego może zostać odszyfrowany jedynie przy użyciu prywatnego.

Jak określić, czy używane jest SSL

Wszystkie strony WWW wyposażone w mechanizm SSL mają adres URL rozpoczynający się od *HTTPS://*. Na stronach, których adres nie zaczyna się w taki sposób, nie należy podawać żadnych cennych informacji, jak choćby numeru karty kredytowej. Przeglądarki WWW zapewniają również dodatkowe wskaźniki bezpiecznego połączenia. W przeglądarce Netscape Navigator, przy dolnej krawędzi okna, pojawia się kluczyk. Również w przeglądacre Internet Explorer u dołu okna pojawia się właściwy znacznik — kłódka.

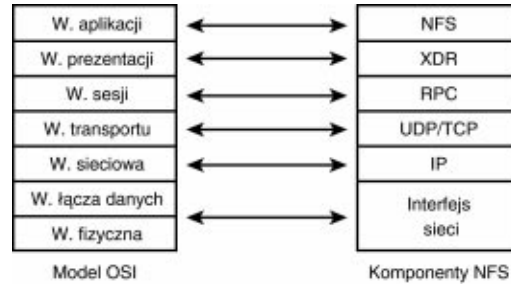
Wszelkie dalsze dane są szyfrowane i odszyfrowywane przy użyciu przesłanego serwerowi klucza głównego.

Network File System (NFS)

RFC 3010

Sieciowy system plików (ang. NFS — *Network File System*) został opracowany w firmie Sun Microsystems w celu umożliwienia klientom sieciowym przezroczystego dla nich dostępu do zdalnych systemów plików. Określenie „przezroczysty" oznacza tu, że klient nie zdaje sobie sprawy z tego, że korzysta z plików na serwerze zdalnym. Pliki dostępne są w sposób identyczny, jak pliki lokalne. W celu utworzenia uchwytu pliku łączącego z zasobem zdalnym, klient wykonuje operację instalowania (ang. *mounting*). Uchwyt zapewnia możliwość korzystania z sieciowego źródła plików podobnie jak z zasobu lokalnego. Ukoronowaniem tej operacji jest przypisanie źródłu zdalnemu litery dysku. Funkcje te są realizowane przez zdalne wywołania procedur (RPC) na poziomie sesji oraz mechanizm zewnętrznej reprezentacji danych w warstwie reprezentacji modelu OSI (patrz rysunek 15.9).

Rysunek 15.9.
Oprogramowanie NFS



Jako podstawowy protokół transportu NFS wykorzystuje UDP, choć od wersji NFS 3.0 (aktualnie najnowsza to NFS 4.0) może używać do tego celu również protokołu TCP. Klient w celu uzyskania dostępu do zdalnych źródeł plików łączy się z portem TCP lub UDP serwera NFS o numerze 2049.

Protokół NFS został zrealizowany w formie protokołu bezstanowego. Serwer nie śledzi wykonywanych operacji dyskowych. Do klienta należy odpowiedzialność za zarządzanie wszystkimi działaniami i ich śledzenie. Taki model realizowania operacji określa się terminem powtarzalny (ang. *idempotent*). Klient musi dokładnie opisać wykonywaną operację. Nie może liczyć na to, że serwer będzie pamiętał treść transakcji wcześniejszych.

Przykładem może być tu operacja zapisu, przy której klient określa:

- plik, w którym będą zapisywane dane (poprzez uchwyt),
- liczbę zapisywanych bajtów,
- początkowy punkt operacji.

Ponieważ operacje NFS opierają się zazwyczaj na protokole UDP, klient NFS przed przejściem do transakcji kolejnej oczekuje na potwierdzenie. Jeżeli nie zostanie ono otrzymane, po czasie mierzonym zegarem retransmisji klient ponawia próbę zrealizowania transakcji. Czynność ta jest powtarzana aż do przekroczenia określonej progowej liczby ponowień. Wówczas transakcja zostaje uznana za zakończoną niepowodzeniem.

Chociaż rozwiązanie takie może się wydawać nieefektywne, to pozwala na zachowanie względnej prostoty architektury serwera NFS. Serwer może ulec awarii lub zostać zrestartowany i w dalszym ciągu utrzymywać połączenia stacji klienckich. Powtarzają one próby wykonania operacji do czasu ponownej dostępności serwera.



NFS i TCP

Wersja NFS 3.0 wprowadza możliwość komunikacji klienta i serwera opartej na protokole transportowym TCP. Główną korzyścią płynącą ze stosowania TCP jest możliwość wykorzystywania do przesyłania transakcji mechanizmu „okien przesuwnych" (ang. *sliding windows*). Zamiast oczekiwania na potwierdzenie każdego żądania zapewniona jest możliwość efektywnego przesyłania kolejnych. Uzyskiwany wzrost wydajności w stosunku do UDP sięga 200%.

Remote Procedure Calls (RPC)

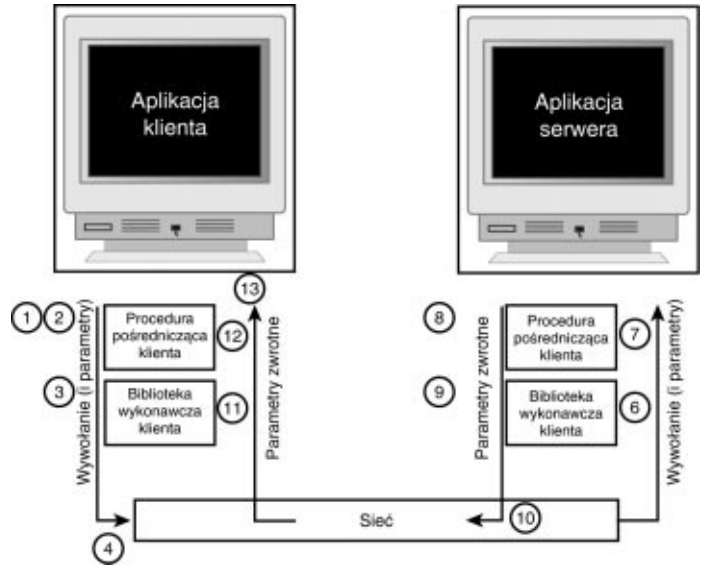
Zdalne wywołania procedur (RPC ang. — *Remote Procedure Calls*) pełnią znaczącą rolę w realizacji NFS. Pozwalają klientowi na wykorzystywanie plików serwera zdalnego. Zapewniają możliwość traktowania zdalnego procesu udostępniania plików podobnie jak zasobu lokalnego.

Mechanizm realizacji wywołań RPC jest podobny do wykorzystywanego w programowaniu wywoływania funkcji. Wykonanie RPC na stacji zdalnej przebiega na poziomie zabezpieczeń użytkownika wywołującego. Zapewnia to zachowanie wprowadzonych zasad ochrony również w sytuacji wykonywania procedury na stacji zdalnej.

Rysunek 15.10 przedstawia przebieg komunikacji podczas realizowania przez stację zdalną wywołania RPC.

1. Aplikacja klienta przekazuje procedurze pośredniczącej (ang. *client stub*) zwykłe wywołanie funkcji.
2. Procedura pośrednicząca przekształca parametry wejściowe z lokalnej reprezentacji danych na reprezentację pośrednią, używaną w komunikacji pomiędzy stacjami korzystającymi z różniących się reprezentacji wewnętrznych. W tym przykładzie „reprezentacja danych" oznacza system wykorzystywany

Rysunek 15.10.
Komunikacja RPC



przez stację lokalną do przechowywania znaków i danych. Przykładem różniących się reprezentacji może być komunikacja stacji wykorzystującej system ASCII ze stacją wykorzystującą EBCDIC.

3. Procedura pośrednicząca wywołuje bibliotekę wykonawczą klienta (ang. *client runtime*). Jest to zazwyczaj biblioteka procedur pełniąca funkcje wymagane przez procedurę pośredniczącą.
4. Biblioteka wykonawcza przesyła komunikat zawierający kapsułkowaną reprezentację XDR wywołania funkcji do biblioteki wykonawczej serwera.
5. Biblioteka wykonawcza serwera przekazuje wywołanie procedurze pośredniczącej serwera.
6. Procedura pośrednicząca przekształca kapsułkowane parametry z postaci pośredniej na reprezentację danych wykorzystywaną przez stację serwera.
7. Procedura pośrednicząca następnie wywołuje aplikację serwera i przekazuje dane wejściowe klienta do dalszego przetwarzania.
8. Po zakończeniu przetwarzania aplikacja serwera zwraca zbiór wyników procedurze pośredniczącej.
9. Procedura pośrednicząca serwera przekształca zbiór wyników i pozostałe parametry z nim związane na reprezentację pośrednią, wykorzystywaną w transmisji. Komunikat zawierający kapsułkowane dane jest przekazywany bibliotece wykonawczej.
10. Biblioteka wykonawcza serwera przesyła komunikat do biblioteki wykonawczej klienta.
11. Biblioteka wykonawcza klienta przekazuje komunikat procedurze pośredniczącej.

12. Procedura pośrednicząca odczytuje zawarty w komunikacie zbiór wyników i przekształca na lokalną reprezentację danych klienta.
13. Procedura pośrednicząca zwraca zbiór wyników i dołączone do nich parametry funkcji wywołującej.

External Data Representation (XDR)

Zewnętrzna reprezentacja danych (ang. XDR — *External Data Representation*) zapewnia format pośredni, wykorzystywany przez wywołania RPC podczas przesyłania danych pomiędzy systemami klienta i serwera. Wprowadzenie formatu XDR było uwarunkowane koniecznością rozwiązania następujących problemów wymiany binarnych danych między systemami heterogenicznymi:

- różne uporządkowanie bajtów w ramach struktur wielobajtowych[1],
- różne reprezentacje typów danych (jak np. kody EBCDIC i ASCII),
- wyrównywanie struktur.

Format XDR zawsze pośredniczy w przesyłaniu danych pomiędzy systemami klienta i serwera. Zapewnia to, że system odbierający zawsze otrzyma dane w czytelnej dla niego postaci.

Portmapper

NFS nie jest jedyną aplikacją korzystającą z wywołań RPC. Ze względu na możliwość korzystania z wywołań zdalnych przez różne aplikacje jednocześnie, na wszystkich serwerach realizujących połączenia RPC z klientami obecny jest program mapujący — *portmapper*. Kiedy klient łączy się z serwerem w celu komunikowania się z aplikacją za pomocą RPC, portmapper rozpoczyna swoje działania od określenia, czy na serwerze została uruchomiona aplikacja. Jeżeli tak, systemowi klienta zwracany jest odpowiedni numer portu. Klient korzysta z niego do czasu zakończenia pracy aplikacji.

Portmapper używa portów UDP i TCP o numerze 111. Każda uruchamiana na serwerze usługa RPC informuje program mapujący o numerach wykorzystywanych portów.

Procedury wywoływane przez NFS

System NFS korzysta z określonych, przeznaczonych dla niego wywołań RPC. Zapewniają one realizację wymaganych przez klienty funkcji.

[1] W najprostszym przykładzie struktury wielobajtowej, jaką jest dwubajtowa liczba całkowita, bardziej znaczący bajt może znajdować się pod wyższym adresem (jak w procesorach Intela 80×86 — konwencja taka określana jest potocznie *little-endian*) albo pod adresem niższym (jak np. w procesorach Motoroli — *big-endian*); zasada ta dotyczy konsekwentnie wszystkich wielobajtowych (dwu-, cztero- i ew. ośmiobajtowych) liczb całkowitych w danym procesorze — *przyp. red.*

- NULL. Procedura, która nie powoduje wykonania żadnej czynności. Pozwala sprawdzić, czy serwer NFS odpowiada na wywołania.
- GETATTR. Procedura Get File Attributes (pobierz atrybuty pliku) pozwala uzyskać atrybuty znajdującego się na serwerze pliku. Należą do nich m.in. ochrona, właściciel, rozmiar i czasy dostępu.
- SETATTR. Procedura Set File Attributes (ustaw atrybuty pliku) zmienia atrybuty pliku.
- ROOT. Procedura ta nie jest już używana. Była wykorzystywana do instalowania systemów plików. Obecnie zapewnia to protokół montowania, opisany dalej.
- LOOKUP. Procedura Lookup Filename (wyszukaj nazwę pliku) przeszukuje dla klienta określony katalog. Zwraca zarówno uchwyt, który będzie wykorzystywany przez klienta w dostępie do pliku, jak i atrybuty.
- ACCESS. Procedura Access (dostęp) sprawdza prawa dostępu użytkownika.
- READLINK. Procedura Read From Symbolic Link (odczytaj z dowiązania symbolicznego) odczytuje wartość przechowywaną w dowiązaniu symbolicznym.
- READ. Procedurę Read From File (odczytaj z pliku) klient wykorzystuje do odczytywania zawartości pliku.
- WRITE. Procedurę Write To File (zapisz do pliku) klient wykorzystuje do zapisywania danych do pliku.
- CREATE. Procedura Create File (utwórz plik) tworzy w katalogu plik.
- MKDIR. Procedura Create Directory (utwórz katalog) tworzy katalog.
- SYMLINK. Procedura Create Symbolic Link (utwórz dowiązanie symboliczne) tworzy symboliczne dowiązanie istniejącego pliku.
- MKNOD. Procedura Create Special Device (utwórz urządzenie specjalne) tworzy pliki urządzeń oraz potoki nazwane.
- REMOVE. Procedura Remove File (usuń plik) usuwa plik.
- RMDIR. Procedura Remove Directory (usuń katalog) usuwa pusty katalog.
- RENAME. Procedura Rename (zmień nazwę) zmienia nazwę pliku lub katalogu.
- LINK. Procedura Link (dowiązanie) tworzy dowiązanie zwykłe istniejącego pliku.
- READDIR. Procedura Read From Directory (odczytaj katalog) pozwala przejrzeć zawartość katalogu. Wymaga ona wykorzystania informacji o stanie połączenia klienta z serwerem, aby zawartość katalogu mogła zostać zwrócona w całości. Zapewnia to „magiczne ciasteczko" (ang. *magic cookie*) zawierające wskaźnik do następnego pliku, którego dotyczą zwracane informacje. Funkcja zwraca nazwy plików i ich numery identyfikacyjne.
- READDIRPLUS. Procedura Extended Read From Directory (odczytaj rozszerzone informacje katalogu) również zwraca zawartość katalogu i funkcjonuje podobnie jak READDIR. Oprócz nazwy i numeru identyfikacyjnego pliku, zwraca jego atrybuty i uchwyt. Znacznie zwiększa efektywność przeglądania katalogów.

- FSSTAT. Procedura Get Dynamic File System Information (pobierz dynamiczne dane systemu plików) zwraca ulotne informacje o stanie systemu plików. Jeżeli serwer NFS nie obsługuje wszystkich atrybutów, zwraca wszystkie, które może. We wcześniejszych wersjach NFS polecenie to nosiło nazwę STATFS.
- FSINFO. Procedura Get Static File System Information (pobierz statyczne dane systemu plików) ściąga trwałe informacje o stanie systemu plików oraz ogólne informacje o implementacji serwera NFS.
- PATHCONF. Procedura Path Configuration (konfiguracja ścieżki) zwraca informacje specyficzne dla systemu POSIX. Obejmują one maksymalną liczbę łączy zwykłych, maksymalną długość nazwy obiektu, sposób obsługi obcięcia nazwy uchwytu oraz tryb rozróżniania wielkich i małych liter.
- COMMIT. Procedura Commit (zatwierdź) powoduje zapisanie danych wcześniej przekazanych wywołaniem WRITE, a przechowywanych w buforze (pamięci podręcznej) na dysku. Zabezpiecza to przed utratą danych oczekujących na zapisanie w buforze serwera.

Metody uwierzytelniania

Mechanizm RPC przewiduje uwierzytelnianie wywołań klienta. Dodatkowo w NFS 4.0 wprowadzona została obsługa wywołań zwrotnych, co pozwala potwierdzić, że klient przesyłający wywołanie RPC ma wystarczające uprawnienia. NFS 3.0 obsługuje następujące mechanizmy uwierzytelniania:

- AUTH_NONE,
- AUTH_UNIX,
- AUTH_SHORT,
- AUTH_DES,
- AUTH_SYS,
- AUTH_DH,
- AUTH_KRB4,
- RPCSEC_GSS.

AUTH_NONE

Uwierzytelnianie AUTH_NONE faktycznie nie istnieje, sprowadza się bowiem do braku jakiegokolwiek uwierzytelnienia. Jest często wykorzystywane do dostępu do danych przeznaczonych tylko do odczytu, które nie są pod żadnym względem poufne. Dostęp do takich zbiorów nie wymaga uwierzytelnienia użytkownika.

AUTH_UNIX

Metoda AUTH_UNIX jest oparta na tradycyjnym uwierzytelnianiu uniksowym. Wykorzystanie tego mechanizmu wymaga podania przez klienta NFS identyfikatora użytkownika (UID), identyfikatora grupy (GID) oraz informacji o grupie.

Kiedy stosowany jest mechanizm uwierzytelniania AUTH_UNIX, dobrze jest korzystać z usługi informacji NIS. Zapewniona jest w ten sposób wspólna dla wszystkich stacji i jednolita baza kont.

Podstawowym problemem związanym z mechanizmem AUTH_UNIX jest przesyłanie całości wymaganych do uwierzytelnienia danych tekstem jawnym. Program monitorujący sieć może przechwycić transmisję i wykorzystać uzyskane dane do naśladowania klienta w realizacji własnych wywołań.

AUTH_SHORT

W metodzie uwierzytelniania AUTH_SHORT sekwencja uwierzytelniająca klienta jest oparta na danych zwróconych przez serwer. Jest to zazwyczaj wykorzystywane przy kontynuacji połączenia RPC. Klient odwołuje się w wywołaniu do połączenia poprzedniego, dzięki czemu proces zostaje skrócony.

AUTH_DES

Metoda AUTH_DES wykorzystuje w transmisji pakietów uwierzytelnianie DES (ang. DES — *Data Encryption Standard* — standard szyfrowania danych). Wykorzystywany jest model klucza publicznego i prywatnego oraz kluczy sesji.

Zalety uwierzytelniania DES w stosunku do uwierzytelniania uniksowego są następujące:

- metoda uwierzytelniania nie jest związana z określonym systemem operacyjnym,
- mechanizm uwierzytelniania stosuje do przesłania niezbędnych danych transmisję szyfrowaną,
- aby podszyć się pod klienta, konieczne jest uzyskanie klucza prywatnego lub hasła sieciowego klienta.

AUTH_SYS

Metoda uwierzytelniania AUTH_SYS opiera się na danych wprowadzonych wcześniej, przy uwierzytelnianiu użytkownika przez system operacyjny.

AUTH_DH

W metodzie AUTH_DH klient generuje sekwencję uwierzytelniania, a serwer zwraca dane mechanizmu uwierzytelniania. Sposób ten stosuje się zasadniczo przy wznawianiu połączeń RPC — odwołanie do sesji wcześniejszej pozwala skrócić czas inicjowania komunikacji.

AUTH_KRB4

Mechanizm uwierzytelniania AUTH_KRB4 jest oparty na systemie Kerberos 4. Jego przewagą nad wszystkimi innymi metodami szyfrowania jest konieczność wykazania się posiadaniem klucza prywatnego bez jego jednoczesnego ujawnienia.

W systemie Kerberos klient przedstawia jedynie bilet sesji wydany przez Centrum dystrybucji kluczy (KDC). Dodatkową zaletą tego mechanizmu jest możliwość uwierzytelnienia również usługi. Klient może więc upewnić się, że łączy się z usługą właściwą, a nie oszustem.

RPCSEC_GSS

Opcja RPCSEC_GSS wprowadza w systemie NFS rozszerzalny mechanizm uwierzytelniania. Wykorzystanie API GSS i LIPKEY pozwala wykorzystywać protokoły takie jak Kerberos 5 i metody oparte na kluczu publicznym jako protokoły uwierzytelniania NFS.

RPCSEC_GSS nie jest kolejnym mechanizmem uwierzytelniania, ale interfejsem programowania apikacji, który umożliwia współpracę innych popularnych protokołów uwierzytelniania z usługą systemu plików oraz uwierzytelnianie obu stron komunikacji.

Instalowanie systemu plików

System NFS 4.0 przewiduje możliwość zapewnienia standardowego mapowania ścieżek sieciowych do obiektów plikowych klienta NFS. Jest to rodzima cecha sieciowego systemu plików nie wymagająca, jak we wcześniejszych wersjach NFS, odwoływania się do tzw. protokółu instalowania systemów plików.

Wcześniejsze wersje systemu NFS

W poprzednich wersjach NFS protokół instalowania systemów plików (ang. *mount protocol*) był osobnym protokołem wykorzystywanym w mechanizmie NFS do realizacji następujących zadań:

- określanie ścieżek dostępu do serwerów,
- zatwierdzanie użytkowników,
- weryfikowanie praw dostępu,
- zapewnianie klientom dostępu do katalogu root zdalnego systemu plików (przez podanie uchwytu do punktu dostępu).

Pozostawały one oddzielone od protokołu NFS, co zapewniało możliwość modyfikowania metod dostępu i sprawdzania poprawności bez konieczności zmieniania samego protokołu NFS. Protokół instalowania jako mechanizm uwierzytelniania wykorzystuje obecnie metody AUTH_NONE, AUTH_UNIX, AUTH_SHORT, AUTH_DES i AUTH_KERB (wcześniejsza wersja uwierzytelniania Kerberos).

Protokół NFS wykorzystuje następujące procedury RPC do instalowania i usuwania instalacji systemów plików:

- OPEN. Wykonuje wszystkie operacje wyszukiwania, tworzenia i udostępniania plików. Jest to pojedyncza procedura wykonująca wszystkie typy operacji.
- CLOSE. Powoduje wyjście ze stanu zainicjowanego procedurą OPEN.

Klient nawiązujący komunikację z serwerem NFS inicjuje połączenie poprzez utworzenie uchwytu plikowego (ang. *file handle*). W systemie NFS 4.0 pierwszym uchwytem klienta może być:

- *Uchwyt plikowy root*. „Korzeń" drzewa przestrzeni nazw systemu plików (z perspektywy klienta). Klient tworzy połączenie z uchwytem root, inicjując operację PUTROOTFH. Uzyskuje wówczas możliwość wyszukiwania i używania plików drzewa serwera NFS. Używa do tego celu procedury LOOKUP.
- *Publiczny uchwyt plikowy*. Uchwyt plikowy, który może być dowiązany do dowolnego innego obiektu systemu plików serwera NFS.



Jest dopuszczalne, żeby uchwyty root i publiczny odwoływały się do tego samego obiektu systemu plików. Utworzenie odpowiednio zabezpieczonej konfiguracji uchwytów należy do administratora NFS.

Blokowanie plików pod kontrolą NFS

W starszych wersjach (przed 4.0) NFS był protokołem bezstanowym — realizacja blokowania wymaga zastosowania protokołu dodatkowego. Istotą blokowania plików jest uniemożliwienie dostępu do pliku więcej niż jednemu klientowi jednocześnie (jeżeli nie jest to wymagane przez aplikację). Funkcję tę zapewnia systemowi NFS w wersji 3.0 protokół menedżera blokowania sieciowego (ang. NLM — *Network Lock Manager*).

W wersji 4.0, model blokowania uległ istotnej zmianie. W miejsce NLM wprowadzone zostało utrzymywanie stanu blokady oparte na dzierżawach. Dla każdego klienta przyłączanego do zasobu NFS serwer określa pewien okres, przez który inne klienty nie mogą uzyskiwać dostępu do tego samego zasobu. Dzierżawa może być ponawiana operacją NEW lub, prościej, poprzez wykonanie operacji odczytu z zasobu.

Jest to prostsza procedura niż w przypadku NLM (w NFS 3.0). Protokół NLM określa, który klient korzysta z blokady pliku. Blokada może być albo *wyłączna* (dostęp do pliku nie jest dozwolony do czasu usunięcia blokady), albo *ze współużytkowaniem* (wiele klientów może łączyć się z plikiem).

W protokole NLM wykorzystywane są również zegary. Zabezpieczają one przed trwałą blokadą pliku w przypadku zawieszenia się systemu klienta.

Web Distributed Authoring and Versioning (WebDAV)

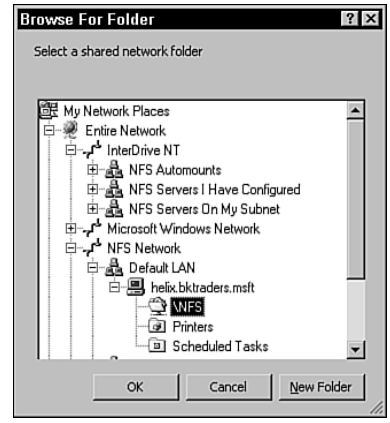
RFC 2518 *Rozproszone tworzenie i obsługa wersji w WWW* (ang. WebDAV — *Web Distributed Authoring and Versioning*) to oparty na protokole HTTP sposób udostępniania plików w sieci Internet. Jego przeznaczeniem jest zapewnianie chronionego udostępniania plików opartego na protokole, który może być przekazywany przez zapory firewall.

Wprowadzanie NFS w środowisku Windows

System Windows nie zawiera usług klienta i serwera NFS w standardowym pakiecie. Dostępne są jednak rozwiązania pochodzące z innych firm. Sama firma Microsoft oferuje Services for Unix 2.0.

Najpopularniejszym spośród oprogramowania klienckiego innych firm jest pakiet NetMANAGE ViewNOW InterDrive pracujący w systemach Windows 9x, Windows NT 4.0 i Windows 2000. Umożliwia on klientom instalowanie woluminów NFS przy użyciu standardowego interfejsu systemu (patrz rysunek 15.11).

Rysunek 15.11. *Przeglądanie zasobów NFS*



Udostępniony w systemie NFS katalog instalujemy, korzystając z polecenia Windows *Map Network Drive* (mapuj dysk sieciowy). Jest ono dostępne po kliknięciu prawym przyciskiem myszy ikony *Network Neighborhood* (otoczenie sieciowe) lub, w nowszych wersjach systemu, *My Network Places* (moje miejsca sieciowe). Zasoby NFS wyświetlane są podobnie jak zasoby sieci Windows czy Novell NetWare.

Podobnie działa oprogramowanie klienta i serwera dostępne w pakiecie Services for Unix 2.0.

Każde z tych rozwiązań umożliwia współpracę zarówno z serwerami, jak i klientami pod kontrolą systemu Unix.

WebDAV jest rozwiązaniem wieloplatformowym, wspieranym przez takie marki jak Apache, Adobe, Macromedia, Netscape, Oracle i Microsoft.



W Windows 2000, Windows XP i Windows .NET Server foldery WebDAV określane są w interfejsie użytkownika nazwą *Foldery sieci Web* (*Web folders*).

Po nawiązaniu połączenia z udziałem WebDAV korzystanie z protokołu HTTP pozostaje niewidoczne dla klientów publikujących i zarządzających zasobami na serwerze.

Zabezpieczenia w protokole WebDAV

Jedną z podstawowych przyczyn opracowania protokołu WebDAV była potrzeba rozwiązania problemów towarzyszących korzystaniu z transmisji FTP. Wprowadzono więc następujące ulepszenia:

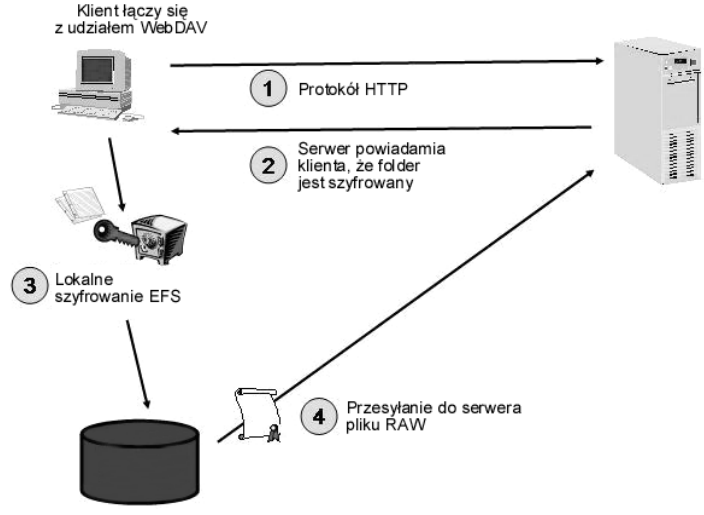
- *Ochrona hasła*. Wprowadzenie połączenia uwierzytelniania podstawowego z szyfrowaniem SSL pozwala szyfrować wszystkie hasła związane z dostępem do udziału WebDAV. Można również stosować uwierzytelnianie certyfikatowe.
- *Szyfrowanie transmisji danych*. Jeżeli na serwerze obsługującym udział WebDAV włączono SSL, wszystkie przesyłane do serwera WWW dane są szyfrowane.
- *Szyfrowanie danych przechowywanych*. System Windows .NET Server wprowadza możliwość połączenia WebDAV z systemem szyfrowania EFS. Oznacza to, że plik może zostać zaszyfrowany na kliencie zdalnym i w tej postaci przesłany do udziału WebDAV.

Współpraca EFS i WebDAV

System plików szyfrowania (ang. EFS — *Encrypting File System*) to technologia szyfrowania obsługiwana przez systemy Windows 2000 Professional i Windows XP Professional. W połączeniach z Windows .NET Server można wykorzystać EFS w transmisjach WebDAV. Plik zaszyfrowany na komputerze klienta może być dzięki temu bezpiecznie zapisany na serwerze WebDAV. Przesyłanie w postaci zaszyfrowanej zabezpiecza przed przechwyceniem danych.

Rysunek 15.12 przedstawia przykład wykorzystania EFS do szyfrowania danych kopiowanych na serwer WWW. Procedura jest następująca:

Rysunek 15.12. *EFS i WebDAV*



1. Klient łączy się z udziałem WebDAV, wywołując `http://serwer/udział` lub `\\serwer\udział` (gdzie *serwer* to nazwa serwera WebDAV, na którym znajduje się udział, a *udział* to nazwa udostępniania folderu WebDAV).

Klient przeprowadza procedurę uwierzytelniania odpowiednio do konfiguracji mechanizmów zabezpieczeń serwera WebDAV.

2. Jeżeli folder WebDAV został skonfigurowany tak, aby wymagał szyfrowania EFS, serwer powiadamia klienta o takiej sytuacji, wymagając szyfrowania każdego zapisywanego w folderze pliku.
3. Generowany jest klucz symetryczny i plik zostaje zaszyfrowany. Sam klucz szyfrowania pliku (FEK — ang. *File Encryption Key*) jest szyfrowany przy użyciu klucza prywatnego użytkownika i w tej postaci zapisywany w polu odszyfrowywania danych (DDF — ang. *Data Decryption Field*) pliku.
4. Zaszyfrowany plik jest przesyłany do procesu Wininet, po czym jako plik RAW ładowany do serwera.

Plik RAW występuje w lokalnym systemie plików serwera WWW jako pozbawiony cech szczególnych plik binarny. Jedynie dostęp poprzez udział WebDAV pozwala ujawnić jego atrybuty.

Podsumowanie

W rozdziale zostało omówionych wiele różnych protokołów przesyłania plików dostępnych w sieci TCP/IP. Najczęściej używanym przez czytelnika protokołem będzie prawdopodobnie FTP. Wraz ze wzrostem znaczenia zagadnień bezpieczeństwa sieci można jednak oczekiwać coraz powszechniejszego stosowania metod alternatywnych, jak narzędzia WebDAV. Warto pamiętać, że do przesyłania plików można wykorzystywać wiele narzędzi z kategorii freeware i shareware. Większość z nich jest znacznie bardziej intuicyjna w obsłudze niż opisane tutaj metody oparte na wierszu poleceń.

Pytania sprawdzające

1. Jakie są podstawowe różnice pomiędzy protokołami FTP i TFTP?
2. Co znaczy termin *anonimowe FTP*?
3. Porównaj bezpieczeństwo protokołów RCP i FTP.
4. Jakie dwa kanały komunikacyjne są wykorzystywane w sesji FTP? Jakie porty są używane podczas sesji po stronie serwera?
5. Opisz kolejne kroki w ustanawianiu sesji SSL.
6. Jakie funkcje zapewnia system NFS w sieci?
7. Jakie funkcje zapewnia serwerowi NFS protokół instalowania systemów plików?
8. Wyjaśnij pojęcie blokowania plików. Dlaczego jest ono ważne w środowisku sieciowym?

9. Dlaczego zabezpieczenia WebDAV są lepsze od zabezpieczeń FTP?
10. Opisz szyfrowanie w komunikacji WebDAV między systemami Windows XP i Windows .NET Server.
11. Który z różnych, obsługiwanych przez system NFS mechanizmów uwierzytelniania, zapewnia możliwość rozszerzania pod kątem protokołów zaprojektowanych w przyszłości?

W następnym rozdziale

W kolejnym rozdziale zajmiemy się pocztą elektroniczną. Kluczowym zagadnieniem będą dla nas protokoły transportu poczty. Należą do nich protokoły SMTP, POP3 i IMAP.

Przyjrzymy się także zagadnieniom związanym z załącznikami i zabezpieczaniem komunikacji.