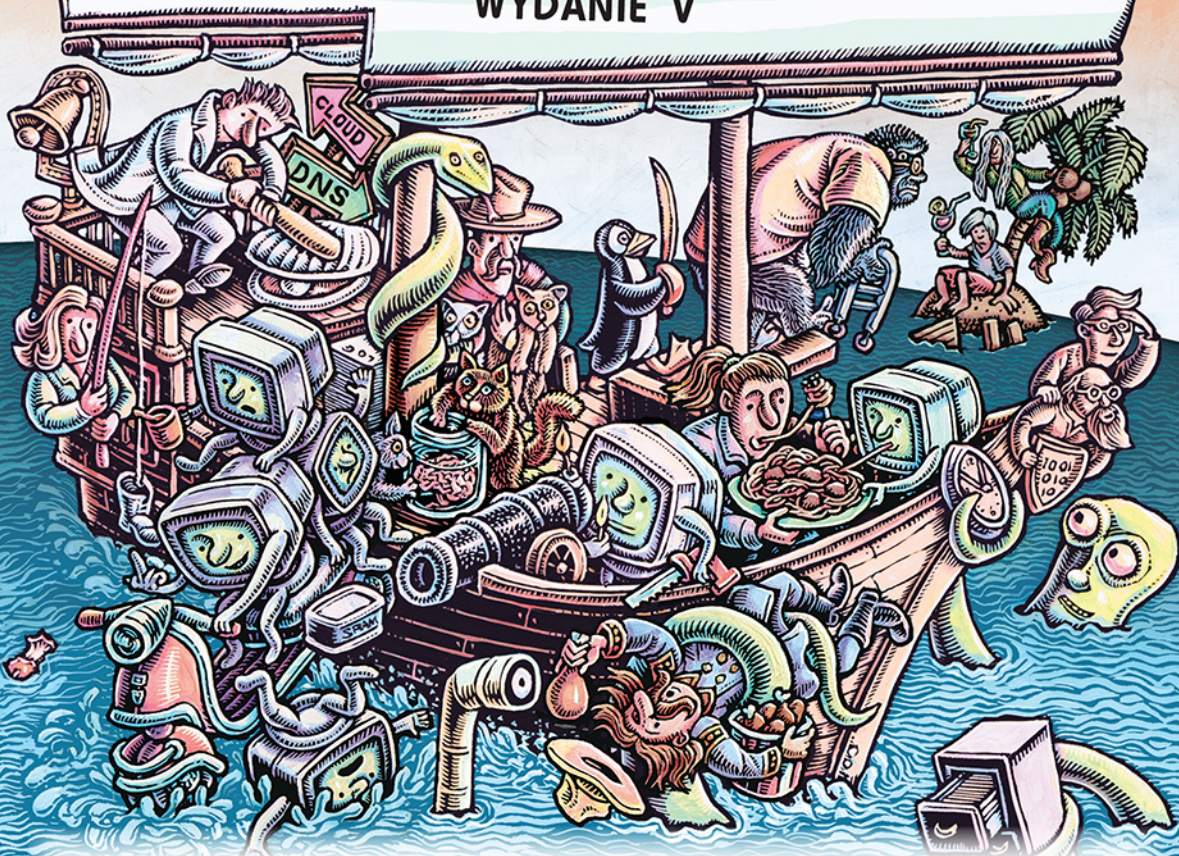


# UNIX<sup>®</sup> I LINUX<sup>®</sup>

## PRZEWODNIK ADMINISTRATORA SYSTEMÓW

WYDANIE V



**EVI NEMETH • GARTH SNYDER • TRENT R. HEIN**  
**BEN WHALEY • DAN MACKIN**

James Garnett, Fabrizio Branca, Adrian Mouat

**Helion**

Tytuł oryginału: UNIX and Linux System Administration Handbook (5th Edition)

Tłumaczenie: Leszek Sagalara

ISBN: 978-83-283-4175-3

Authorized translation from the English language edition, entitled: UNIX AND LINUX SYSTEM ADMINISTRATION HANDBOOK, Fifth Edition; ISBN 0134277554; by Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley, and Dan Mackin; published by Pearson Education, Inc, publishing as Addison-Wesley Professional. Copyright © 2018 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. Polish language edition published by HELION S.A. Copyright © 2018.

Ubuntu is a registered trademark of Canonical Limited, and is used with permission.  
Debian is a registered trademark of Software in the Public Interest Incorporated.  
CentOS is a registered trademark of Red Hat Inc., and is used with permission.  
FreeBSD is a registered trademark of The FreeBSD Foundation, and is used with permission.  
The Linux Tux logo was created by Larry Ewing, lewing@isc.tamu.edu.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Wydawnictwo HELION  
ul. Kościuszki 1c, 44-100 GLIWICE  
tel. 32 231 22 19, 32 230 98 63  
e-mail: [helion@helion.pl](mailto:helion@helion.pl)  
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!  
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres  
<http://helion.pl/user/opinie/unlip5>  
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

# SPIS TREŚCI

<b>Pamięci Evi</b>	<b>35</b>
<b>Wstęp</b>	<b>37</b>
<b>Słowo wstępne</b>	<b>39</b>
<b>Podziękowania</b>	<b>41</b>

## I. PODSTAWY ADMINISTROWANIA

<b>1</b>	<b>Od czego zacząć?</b>	<b>45</b>
1.1.	Podstawowe obowiązki administratora	46
	Kontrola dostępu	46
	Podłączanie sprzętu	46
	Automatyzacja zadań	47
	Nadzorowanie kopii zapasowych	47
	Instalacja i aktualizowanie oprogramowania	47
	Monitorowanie	48
	Rozwiązywanie problemów	48
	Zarządzanie lokalną dokumentacją	48
	Uważne monitorowanie stanu zabezpieczeń	48

- Dostosowywanie wydajności 49
- Opracowanie polityki serwera 49
- Współpraca z dostawcami 49
- Udzielanie pomocy użytkownikom 49
- 1.2. Podstawowe narzędzia administratora 50
- 1.3. Dystrybucje systemu Linux 51
- 1.4. Przykładowe systemy używane w tej książce 52
  - Przykładowe dystrybucje systemu Linux 53
  - Przykładowa dystrybucja systemu Unix 54
- 1.5. Notacja i konwencje typograficzne 54
- 1.6. Jednostki 56
- 1.7. Strony podręcznika systemowego i inne rodzaje dokumentacji 57
  - Organizacja podręcznika systemowego 57
  - man — czytanie stron podręcznika systemowego 58
  - Miejsce przechowywania stron podręcznika 58
- 1.8. Inna dokumentacja autorytatywna 59
  - Przewodniki dotyczące określonych systemów 59
  - Dokumentacja dotycząca określonych pakietów 59
  - Książki 60
  - RFC i inne dokumenty internetowe 60
- 1.9. Inne źródła informacji 60
  - Bądź na bieżąco 61
  - Dokumenty HOWTO i witryny informacyjne 61
  - Konferencje 62
- 1.10. Sposoby wyszukiwania i instalacji oprogramowania 62
  - Jak sprawdzić, czy oprogramowanie jest już zainstalowane? 63
  - Instalowanie nowego oprogramowania 64
  - Instalacja oprogramowania ze źródeł 66
  - Instalacja ze skryptu WWW 67
- 1.11. Gdzie hostować? 68
- 1.12. Specjalizacje i dyscypliny pokrewne 69
  - DevOps 69
  - Inżynierowie ds. niezawodności 69
  - Inżynierowie ds. bezpieczeństwa 69
  - Administratorzy sieci 69
  - Administratorzy baz danych 70
  - Inżynierowie sieciowych centrów operacyjnych 70
  - Technicy centrów danych 70
  - Architekci 70
- 1.13. Zalecana literatura 70
  - Administracja systemu i DevOps 71
  - Niezbędne narzędzia 71

## 2 Rozruch i demony zarządzania systemem 73

- 2.1. Przegląd procesu rozruchowego 74
- 2.2. Oprogramowanie sprzętowe systemu 75
  - BIOS a UEFI 75
  - BIOS 76
  - UEFI 76
- 2.3. Programy rozruchowe 78
- 2.4. GRUB 78
  - Konfiguracja programu GRUB 79
  - Wiersz poleceń programu GRUB 80
  - Opcje jądra systemu Linux 80
- 2.5. Rozruch systemu FREEBSD 81
  - Ścieżka BIOS-u — boot0 81
  - Ścieżka UEFI 82
  - Konfiguracja programu loader 83
  - Polecenia programu loader 83
- 2.6. Demony zarządzania systemem 83
  - Zadania procesu init 84
  - Implementacje procesu init 85
  - Tradycyjny init 85
  - systemd kontra reszta świata 86
  - init oszczędny i przykładowie ukarany 86
- 2.7. systemd w szczegółach 87
  - Jednostki i pliki jednostek 87
  - systemctl — zarządzanie systemd 88
  - Statusy jednostek 89
  - Jednostki celu 91
  - Zależności pomiędzy jednostkami 93
  - Kolejność wykonywania 94
  - Przykład bardziej złożonego pliku jednostki 94
  - Usługi lokalne i dostosowywanie 95
  - Zastrzeżenia związane z usługami i kontrolą rozruchu 96
  - Rejestrowanie zdarzeń w systemd 98
- 2.8. init i skrypty startowe w systemie FreeBSD 99
- 2.9. Procedury ponownego uruchamiania i zamykania systemu 101
  - Wyłączanie fizycznych systemów 101
  - Wyłączanie systemów chmurowych 101
- 2.10. Strategie postępowania w przypadku problemów z rozruchem 102
  - Tryb pojedynczego użytkownika 102
  - Tryb pojedynczego użytkownika w systemie FreeBSD 104
  - Tryb pojedynczego użytkownika z programem GRUB 104
  - Odzyskiwanie systemów chmurowych 104

### 3 Kontrola dostępu i uprawnienia administratora 107

- 3.1. Standardowa kontrola dostępu w systemie Unix 108
  - Kontrola dostępu w systemie plików 109
  - Prawa własności do procesów 109
  - Konto użytkownika root 110
  - setuid i setgid 111
- 3.2. Zarządzanie kontem użytkownika root 111
  - Logowanie na konto użytkownika root 111
  - su — zmiana tożsamości użytkownika 112
  - sudo — su z ograniczeniami 113
  - Wyłączanie konta użytkownika root 120
  - Konta systemowe inne niż root 121
- 3.3. Rozszerzenia standardowego modelu kontroli dostępu 122
  - Wady modelu standardowego 122
  - PAM 123
  - Kerberos — sieciowe uwierzytelnianie kryptograficzne 124
  - Listy kontroli dostępu do systemu plików 124
  - Możliwości systemu Linux 124
  - Przestrzenie nazw w systemie Linux 125
- 3.4. Nowoczesne mechanizmy kontroli dostępu 125
  - Oddzielne ekosystemy 126
  - Obowiązkowa kontrola dostępu (MAC) 126
  - Kontrola dostępu oparta na rolach 127
  - Security-enhanced Linux (SELinux) 128
  - AppArmor 129
- 3.5. Zalecana literatura 131

### 4 Kontrolowanie procesów 133

- 4.1. Elementy składowe procesu 134
  - PID — numer identyfikacyjny procesu 134
  - PPID — identyfikator procesu macierzystego 135
  - UID i EUID — rzeczywisty i efektywny identyfikator użytkownika 135
  - GID i EGID — rzeczywisty i efektywny identyfikator grupy 136
  - Uprzejmość 136
  - Terminal sterujący 136
- 4.2. Cykl życia procesu 137
  - Sygnały 137
  - Polecenie kill — wysłanie sygnałów 140
  - Stany procesów i wątków 141

- 4.3. Polecenie ps — monitorowanie procesów 142
- 4.4. Interaktywne monitorowanie procesów — polecenie top 144
- 4.5. Polecenia nice i renice — zmiana priorytetów przełączania 146
- 4.6. System plików /proc 147
- 4.7. Śledzenie sygnałów i funkcji systemowych — polecenia strace i truss 148
- 4.8. Procesy niekontrolowane 150
- 4.9. Procesy okresowe 152
  - cron — harmonogram poleceń 152
  - Powszechne zastosowania zaplanowanych zadań 160

## 5 System plików

**163**

- 5.1. Ścieżki dostępu 165
- 5.2. Montowanie i odmontowywanie systemów plików 166
- 5.3. Organizacja drzewa plików 168
- 5.4. Typy plików 171
  - Zwykłe pliki 172
  - Katalogi 172
  - Dowiązania twarde 173
  - Pliki urządzeń znakowych i blokowych 173
  - Gniazda lokalne 174
  - Nazwane potoki 175
  - Dowiązania symboliczne 175
- 5.5. Atrybuty plików 176
  - Bity uprawnień 176
  - Bit setuid i setgid 177
  - Bit lepkości 177
  - Polecenie ls — wyświetlanie listy i sprawdzanie plików 177
  - Polecenie chmod — zmiana uprawnień 179
  - Polecenia chown i chgrp — zmiana właściciela i grupy 181
  - Polecenie umask — ustawianie uprawnień domyślnych 181
  - Dodatkowe opcje w systemie Linux 182
- 5.6. Listy kontroli dostępu (ACL) 183
  - Mała uwaga 184
  - Rodzaje ACL 184
  - Implementacje ACL 185
  - Obsługa ACL w systemie Linux 186
  - Obsługa ACL w systemie FreeBSD 186
  - Listy ACL w stylu POSIX 186
  - Listy ACL w stylu NFSv4 190

## 6 Oprogramowanie — instalacja i zarządzanie 195

- 6.1. Instalacja systemów operacyjnych 196
  - Instalacja przez sieć 196
  - Konfigurowanie PXE 198
  - Kickstart — zautomatyzowany instalator systemów Red Hat i CentOS 198
  - Automatyczna instalacja przy użyciu instalatora Ubuntu 201
  - Rozruch sieciowy przy użyciu narzędzia Cobbler,  
linuksowego serwera uruchomieniowego typu open source 203
  - Automatyzacja instalacji FreeBSD 203
- 6.2. Zarządzanie pakietami 204
- 6.3. Systemy zarządzania pakietami w Linuksie 206
  - rpm — zarządzanie pakietami RPM 206
  - dpkg — zarządzanie pakietami .deb 207
- 6.4. Wysokopoziomowe systemy zarządzania pakietami w systemie Linux 208
  - Repozytoria z pakietami 209
  - RHN — Red Hat Network 210
  - APT — Advanced Package Tool 210
  - Konfigurowanie repozytorium 212
  - Przykład pliku /etc/apt/sources.list 212
  - Własny serwer lustrzany jako lokalne repozytorium 213
  - Automatyzacja APT 214
  - yum — zarządzanie wydaniem opartym na formacie RPM 215
- 6.5. Zarządzanie oprogramowaniem w systemie FreeBSD 215
  - System bazowy 216
  - pkg — menedżer pakietów FreeBSD 217
  - Kolekcja portów 218
- 6.6. Lokalizowanie i konfigurowanie oprogramowania 219
  - Organizacja procesu lokalizowania 219
  - Strukturyzacja aktualizacji 220
  - Ograniczanie pola gry 220
  - Testowanie 221
- 6.7. Zalecana literatura 221

## 7 Pisanie skryptów i powłoka 223

- 7.1. Filozofia pisania skryptów 224
  - Pisz mikroskrypty 224
  - Poznaj dobrze kilka narzędzi 225
  - Automatyzuj wszystko 226
  - Nie optymalizuj przedwcześnie 226



- Wybierz właściwy język skryptowy 227
- Reguły poprawnego pisania skryptów 228
- 7.2. Podstawy powłoki 230
  - Edycja poleceń 231
  - Potoki i przekierowania 231
  - Zmienne i oznakowanie 233
  - Zmienne środowiskowe 234
  - Popularne polecenia filtrujące 235
- 7.3. Skrypty w powłoce sh 238
  - Wykonywanie poleceń 239
  - Od poleceń do skryptów 240
  - Wejście i wyjście 242
  - Spacje w nazwach plików 243
  - Argumenty wiersza poleceń i funkcje 243
  - Przepływ sterowania 245
  - Pętle 247
  - Działania arytmetyczne 249
- 7.4. Wyrażenia regularne 249
  - Proces dopasowywania 250
  - Znaki dosłowne 250
  - Znaki specjalne 250
  - Przykłady wyrażeń regularnych 252
  - Przechwytywanie 253
  - Zachłanność, lenistwo i katastrofalne wycofania 254
- 7.5. Programowanie w języku Python 255
  - Python 3 255
  - Python 2 czy Python 3? 256
  - Python — szybki start 256
  - Obiekty, łańcuchy, liczby, listy, słowniki, krotki i pliki 258
  - Przykład sprawdzania poprawności wejścia 260
  - Pętle 261
- 7.6. Programowanie w języku Ruby 262
  - Instalacja 263
  - Ruby — szybki start 263
  - Bloki 265
  - Symbole i hasze opcji 266
  - Wyrażenia regularne w języku Ruby 267
  - Ruby jako filtr 268
- 7.7. Zarządzanie bibliotekami i środowiskiem języków Python i Ruby 269
  - Wyszukiwanie i instalowanie pakietów 269
  - Tworzenie odtwarzalnych środowisk 270
  - Wiele środowisk 271

- 7.8. Kontrola wersji przy użyciu Git 274
  - Przykład prostego repozytorium Git 276
  - Zastrzeżenia dotyczące Git 278
  - Społecznościowe tworzenie kodu z systemem Git 278
- 7.9. Zalecana literatura 279
  - Powłoki i tworzenie skryptów 279
  - Wyrażenia regularne 280
  - Python 280
  - Ruby 281

## 8 Zarządzanie użytkownikami

**283**

- 8.1. Mechanika konta 284
- 8.2. Plik `etc/passwd` 285
  - Nazwa użytkownika 285
  - Zaszyfrowane hasło 286
  - Numer UID (identyfikator użytkownika) 288
  - Domyślne numery GID 289
  - Pole GECOS 289
  - Katalog domowy 290
  - Powłoka logowania 290
- 8.3. Plik `/etc/shadow` w systemie Linux 290
- 8.4. Pliki `/etc/master.passwd` i `/etc/login.conf` w systemie FreeBSD 292
  - Plik `/etc/master.passwd` 292
  - Plik `/etc/login.conf` 293
- 8.5. Plik `/etc/group` 294
- 8.6. Ręczne dodawanie użytkowników 296
  - Edycja plików `passwd` i `group` 296
  - Ustawianie hasła 297
  - Tworzenie katalogu domowego i instalowanie plików startowych 298
  - Ustawianie uprawnień i praw własności do katalogu domowego 300
  - Konfigurowanie ról i uprawnień administracyjnych 300
  - Finalizacja 300
- 8.7. Skrypty do dodawania użytkowników: `useradd`, `adduser` i `newusers` 301
  - Polecenie `useradd` w systemie Linux 301
  - Polecenie `useradd` w systemach Debian i Ubuntu 302
  - Polecenie `useradd` w systemie FreeBSD 303
  - Polecenie `newusers` w systemie Linux
    - hurtowe dodawanie użytkowników 304
- 8.8. Bezpieczne usuwanie kont i plików użytkowników 304
- 8.9. Blokowanie kont użytkowników 305

- 8.10. Minimalizowanie ryzyka za pomocą PAM 306
- 8.11. Scentralizowane zarządzanie kontami 307
  - LDAP a Active Directory 307
  - Systemy pojedynczego logowania na poziomie aplikacji 307
  - Systemy zarządzania tożsamością 308

## 9 Chmura obliczeniowa

311

- 9.1. Chmura w kontekście 312
- 9.2. Platformy chmur obliczeniowych 314
  - Chmury publiczne, prywatne i hybrydowe 314
  - Amazon Web Services 315
  - Google Cloud Platform 316
  - DigitalOcean 316
- 9.3. Podstawy usługi chmurowej 317
  - Dostęp do chmury 318
  - Regiony i strefy dostępności 319
  - Wirtualne serwery prywatne 320
  - Sieci 321
  - Pamięć masowa 321
  - Tożsamość i autoryzacja 322
  - Automatyzacja 323
  - Funkcje bezserwerowe 323
- 9.4. Wirtualne serwery prywatne — szybki start 324
  - Amazon Web Services 324
  - Google Cloud Platform 328
  - DigitalOcean 329
- 9.5. Kontrola kosztów 331
- 9.6. Zalecana literatura 333

## 10 Rejestrowanie zdarzeń

335

- 10.1. Położenie plików z dziennikami 338
  - Specjalne pliki dzienników 338
  - Przeglądanie dzienników w rejestratorze systemd 340
- 10.2. Rejestrator systemd 340
  - Konfiguracja rejestratora systemd 341
  - Dodatkowe opcje filtrujące rejestratora 342
  - Współlistnienie z programem syslog 343

- 10.3. syslog 344
  - Czytanie komunikatów systemu syslog 344
  - Architektura systemu rsyslog 345
  - Wersje systemu rsyslog 346
  - Konfiguracja systemu rsyslog 346
  - Przykłady pliku konfiguracyjnego 355
  - Bezpieczeństwo komunikatów systemu syslog 357
  - Diagnostyka konfiguracji systemu syslog 359
- 10.4. Rejestrowanie komunikatów jądra i uruchamiania systemu 359
- 10.5. Pliki dzienników — zarządzanie i rotowanie 360
  - logrotate — międzyplatformowe zarządzanie dziennikami 360
  - newsyslog — zarządzanie dziennikami w systemie FreeBSD 362
- 10.6. Zarządzanie dziennikami na dużą skalę 362
  - Zestaw narzędzi ELK 362
  - Graylog 363
  - Rejestrowanie zdarzeń jako usługa 363
- 10.7. Strategie rejestrowania 364

## 11 Sterowniki i jądro

**367**

- 11.1. Obowiązki administratora systemu związane z jądrem 368
- 11.2. Numerowanie wersji jądra 369
  - Wersje jądra w systemie Linux 369
  - Wersje jądra w systemie FreeBSD 370
- 11.3. Urządzenia i ich sterowniki 370
  - Pliki urządzeń i numery urządzeń 371
  - Wyzwania związane z zarządzaniem plikami urządzeń 372
  - Ręczne tworzenie plików urządzeń 372
  - Nowoczesne zarządzanie plikami urządzeń 373
  - Zarządzanie urządzeniami w systemie Linux 373
  - Zarządzanie urządzeniami w systemie FreeBSD 378
- 11.4. Konfigurowanie jądra w systemie Linux 380
  - Dostrajanie parametrów jądra systemu Linux 380
  - Budowanie własnego jądra 381
  - Dodawanie sterownika urządzenia w systemie Linux 385
- 11.5. Konfiguracja jądra w systemie FreeBSD 385
  - Dostrajanie parametrów jądra FreeBSD 385
  - Budowanie jądra w systemie FreeBSD 386

- 11.6. Ładowalne moduły jądra 387
  - Ładowalne moduły jądra w systemie Linux 387
  - Ładowalne moduły jądra w systemie FreeBSD 389
- 11.7. Rozruch 389
  - Komunikaty rozruchowe systemu Linux 390
  - Komunikaty rozruchowe systemu FreeBSD 394
- 11.8. Uruchamianie niestandardowych jąder w chmurze 395
- 11.9. Błędy jądra 396
  - Błędy jądra w systemie Linux 397
  - Panika jądra w systemie FreeBSD 399
- 11.10. Zalecana literatura 400

## 12 Drukowanie

**401**

- 12.1. CUPS 402
  - Interfejsy podsystemu drukowania 403
  - Kolejka drukowania 403
  - Wiele drukarek i kolejek 404
  - Instancje drukarek 404
  - Przeglądanie drukarek sieciowych 404
  - Filtry 405
- 12.2. Administracja serwerem CUPS 406
  - Konfiguracja sieciowego serwera wydruków 407
  - Automatyczna konfiguracja drukarki 407
  - Konfiguracja drukarki sieciowej 407
  - Przykłady konfiguracji drukarek 408
  - Wyłączenie usługi 408
  - Inne zadania konfiguracyjne 409
- 12.3. Rozwiązywanie problemów 409
  - Ponowne uruchamianie demona wydruku 409
  - Pliki dzienników 410
  - Połączenia w drukowaniu bezpośrednim 411
  - Problemy z drukowaniem sieciowym 411
- 12.4. Zalecana literatura 412

## II. SIECI

### 13 Sieci TCP/IP

415

- 13.1. TCP/IP i jego związek z internetem 416
  - Kto zarządza internetem? 416
  - Standardy sieciowe i dokumentacja 417
- 13.2. Podstawy sieci 418
  - IPv4 i IPv6 419
  - Pakiety i enkapsulacja 421
  - Ramkowanie w sieciach Ethernet 422
  - Maksymalna jednostka transmisji (MTU) 422
- 13.3. Adresowanie pakietów 423
  - Adresowanie sprzętowe (MAC) 423
  - Adresowanie IP 424
  - „Adresowanie” za pomocą nazw 425
  - Porty 425
  - Rodzaje adresów 426
- 13.4. Adresy IP — szczegółowe informacje 426
  - Klasy adresów IPv4 427
  - Podział na podsieci w IPv4 428
  - Sztuczki i narzędzia do wyliczania podsieci 429
  - CIDR — bezklasowe trasowanie międzydomenowe 430
  - Przydzielanie adresów 431
  - Adresy prywatne i NAT 431
  - Adresowanie IPv6 433
- 13.5. Wyznaczanie tras 437
  - Tablice tras 438
  - Przekierowania ICMP 439
- 13.6. Protokoły ARP (IPv4) i ND (IPv6) 440
- 13.7. DHCP — protokół dynamicznej konfiguracji hostów 441
  - Oprogramowanie DHCP 441
  - Sposób działania DHCP 442
  - Oprogramowanie DHCP w wersji ISC 443
- 13.8. Kwestie bezpieczeństwa 444
  - Przekazywanie pakietów IP 444
  - Przekierowania ICMP 445
  - Wybór trasy przez nadawcę 445
  - Pakiety ping na adres rozgłoszeniowy  
i inne formy ukierunkowanego rozgłaszania 445

- Falszowanie adresów IP 446
- Zapory sieciowe oparte na serwerze 446
- Wirtualne sieci prywatne 447
- 13.9. Podstawowa konfiguracja sieciowa 448
  - Przypisywanie nazwy komputera i adresu IP 448
  - Interfejs sieciowy i konfiguracja IP 449
  - Konfigurowanie tras 451
  - Konfigurowanie DNS 452
  - Konfigurowanie sieci w różnych systemach 453
- 13.10. Sieci w systemie Linux 454
  - NetworkManager 454
  - ip — ręczne konfigurowanie sieci 455
  - Konfigurowanie sieci w systemach Debian i Ubuntu 456
  - Konfiguracja sieci w systemach Red Hat i CentOS 456
  - Opcje sprzętu sieciowego w systemie Linux 458
  - Opcje TCP/IP w systemie Linux 459
  - Zmienne jądra związane z bezpieczeństwem 461
- 13.11. Sieci w systemie FreeBSD 461
  - ifconfig — konfigurowanie interfejsów sieciowych 462
  - Konfigurowanie sprzętu sieciowego w systemie FreeBSD 462
  - Konfiguracja sieci w systemie FreeBSD w czasie rozruchu 463
  - Konfiguracja TCP/IP w systemie FreeBSD 463
- 13.12. Rozwiązywanie problemów z siecią 464
  - Polecenie ping — sprawdzenie, czy host jest dostępny 465
  - Polecenie traceroute — śledzenie pakietów IP 467
  - Podśluchiwanie pakietów 470
- 13.13. Monitoring sieci 473
  - Polecenie smokeping — gromadzenie statystyk polecenia ping 473
  - iPerf — śledzenie wydajności sieci 474
  - Cacti — gromadzenie danych i tworzenie wykresów 475
- 13.14. Zapory sieciowe i NAT 476
  - iptables w systemie Linux — reguły, łańcuchy i tablice 476
  - Zapora IPFilter dla systemów Unix 481
- 13.15. Sieci w chmurze 484
  - Wirtualna chmura prywatna (VPC) w AWS 484
  - Sieci w GCP 490
  - Sieci w DigitalOcean 492
- 13.16. Zalecana literatura 493
  - Historia 493
  - Pozycje klasyczne i bibliie 493
  - Protokoły 493

**14 Sprzęt sieciowy****495**

- 14.1. Ethernet — sieć uniwersalna 496
  - Przesyłanie sygnałów w sieci Ethernet 496
  - Topologia Ethernetu 498
  - Skrajka nieekranowana 498
  - Włókna światłowodowe 500
  - Łączenie i rozszerzanie sieci Ethernet 501
  - Autouzgadnianie 503
  - Power over Ethernet 504
  - Ramki Jumbo 504
- 14.2. Sieci bezprzewodowe — internet dla nomadów 505
  - Standardy bezprzewodowe 505
  - Klient bezprzewodowy 506
  - Infrastruktura bezprzewodowa i punkty dostępu 506
  - Bezpieczeństwo sieci bezprzewodowych 509
- 14.3. SDN — programowalna sieć komputerowa 509
- 14.4. Testowanie i diagnostyka sieci 510
- 14.5. Układanie okablowania 510
  - Możliwości okablowania skrajką 511
  - Połączenia do biur 511
  - Standardy okablowania 511
- 14.6. Kwestie związane z projektowaniem sieci 512
  - Architektura sieci a architektura budynku 513
  - Rozbudowa 513
  - Przeciążenie 513
  - Konserwacja i dokumentacja 514
- 14.7. Kwestie związane z zarządzaniem 514
- 14.8. Zalecana literatura 515

**15 Wyznaczanie tras****517**

- 15.1. Przesyłanie pakietów — szczegóły 518
- 15.2. Demony i protokoły wyznaczania tras 521
  - Protokoły wektora odległości 522
  - Protokoły stanu łączy 523
  - Miary kosztu 523
  - Protokoły wewnętrzne i zewnętrzne 524
- 15.3. Prezentacja protokołów 524
  - RIP i RIPng — protokół informowania o trasach 524
  - OSPF — najpierw najkrótsza ścieżka 526



- EIGRP — rozszerzony protokół trasowania bramy wewnętrznej 526
- BGP — protokół bramy brzegowej 526
- 15.4. Grupowa koordynacja protokołów wyznaczania tras 527
- 15.5. Kryteria wyboru strategii wyznaczania tras 527
- 15.6. Demony trasujące 528
  - routed — przestarzała implementacja RIP 529
  - Quagga — dominujący demon trasujący 529
  - XORP — router w komputerze 530
- 15.7. Routery Cisco 530
- 15.8. Zalecana literatura 533

## 16 DNS — system nazw domenowych

**535**

- 16.1. Architektura DNS 536
  - Zapytania i odpowiedzi 536
  - Dostawcy usług DNS 537
- 16.2. Wyszukiwania w DNS 538
  - resolv.conf — konfigurowanie resolvera klienta 538
  - nsswitch.conf — kogo zapytać o nazwę? 538
- 16.3. Przestrzeń nazw DNS 539
  - Rejestracja nazwy domeny 540
  - Tworzenie własnych poddomen 540
- 16.4. Jak działa DNS 541
  - Serwery nazw 541
  - Serwery autorytatywne i buforujące 542
  - Serwery rekurencyjne i nierekurencyjne 542
  - Rekordy zasobów 543
  - Delegowania 543
  - Buforowanie i efektywność 545
  - Odpowiedzi wielokrotne i równoważenie obciążenia DNS metodą Round Robin 545
  - Diagnostyka przy użyciu narzędzi do odpytywania 546
- 16.5. Baza danych DNS 549
  - Polecenia dla analizatora w plikach strefowych 549
  - Rekordy zasobów 550
  - Rekord SOA 553
  - Rekordy NS 555
  - Rekordy A 556
  - Rekordy AAAA 556

- Rekordy PTR 557
- Rekordy MX 558
- Rekordy CNAME 559
- Rekordy SRV 560
- Rekordy TXT 561
- Rekordy SPF, DKIM i DMARC 562
- Rekordy DNSSEC 562
- 16.6. Oprogramowanie BIND 562
  - Komponenty BIND 563
  - Pliki konfiguracyjne 563
  - Instrukcja include 565
  - Instrukcja options 565
  - Instrukcja acl 571
  - Instrukcja key (TSIG) 571
  - Instrukcja server 572
  - Instrukcja masters 573
  - Instrukcja logging 573
  - Instrukcja statistics-channels 573
  - Instrukcja zone 574
  - Instrukcja controls dla rndc 577
- 16.7. Rozdzielony DNS i instrukcja view 578
- 16.8. Przykłady konfiguracji BIND 580
  - Strefa localhost 580
  - Mała firma zajmująca się sprawami bezpieczeństwa 581
- 16.9. Aktualizowanie plików strefowych 584
  - Przesyłanie informacji strefowych 584
  - Automatyczne aktualizacje 585
- 16.10. Kwestie związane z bezpieczeństwem DNS 587
  - Nowe spojrzenie na listy kontroli dostępu w BIND 588
  - Otwarty resolver 589
  - Uruchamianie w środowisku chroot 590
  - Bezpieczna komunikacja między serwerami za pomocą TSIG i TKEY 590
  - Konfigurowanie TSIG dla BIND 591
  - DNSSEC 593
  - Strategia dotycząca DNSSEC 594
  - Rekordy zasobów DNSSEC 594
  - Włączanie DNSSEC 596
  - Generowanie par kluczy 596
  - Podpisywanie stref 598
  - Łańcuch zaufania DNSSEC 600
  - Wymiana kluczy DNSSEC 600
  - Narzędzia DNSSEC 601
  - Usuwanie błędów w DNSSEC 603

- 16.11. Diagnostyka systemu BIND 604
  - Rejestrowanie w BIND 604
  - Sterowanie serwerem nazw za pomocą rndc 610
  - Wyszukiwanie niepoprawnych delegowań z poziomu wiersza poleceń 611
- 16.12. Zalecana literatura 613
  - Książki i inna dokumentacja 613
  - Zasoby sieciowe 613
  - Dokumenty RFC 613

## **17 Systemy pojedynczego logowania 615**

- 17.1. Podstawowe elementy SSO 616
- 17.2. LDAP — „lekkie” usługi katalogowe 617
  - Zastosowania LDAP 618
  - Struktura danych w katalogu LDAP 618
  - OpenLDAP — tradycyjna implementacja serwera LDAP na licencji open source 620
  - 389 Directory Server — alternatywna implementacja serwera LDAP na licencji open source 620
  - Zapytania LDAP 621
  - Konwertowanie plików passwd i group do LDAP 622
- 17.3. Wykorzystanie usług katalogowych do logowania 623
  - Kerberos 623
  - Demon sssd 626
  - Plik nsswitch.conf 627
  - PAM — uniwersalny mechanizm uwierzytelniania 627
  - Przykład konfiguracji PAM 629
- 17.4. Rozwiązania alternatywne 630
  - NIS — Network Information Service 630
  - rsync — bezpieczniejszy transfer plików 631
- 17.5. Zalecana literatura 631

## **18 Poczta elektroniczna 633**

- 18.1. Architektura systemów obsługi poczty elektronicznej 634
  - Klienci poczty 635
  - System przyjmujący 635
  - System transportowy 636
  - System dostarczania lokalnego 636
  - Skrzynki pocztowe 637
  - Systemy dostępne 637

- 18.2. Anatomia wiadomości pocztowej 637
- 18.3. Protokół SMTP 640
  - Wysłałeś mi EHLO 640
  - Kody błędów SMTP 641
  - Uwierzytelnianie SMTP 642
- 18.4. Mechanizmy antyspamowe i antywirusowe 643
  - Oszustwa 643
  - SPF i Sender ID 644
  - DKIM 644
- 18.5. Prywatność i szyfrowanie 645
- 18.6. Aliasy pocztowe 646
  - Odczyt aliasów z plików 648
  - Wysyłanie wiadomości do plików 648
  - Wysyłanie wiadomości do programów 649
  - Budowanie bazy aliasów 649
- 18.7. Konfiguracja serwera poczty 649
- 18.8. Sendmail 651
  - Plik switch 652
  - Uruchamianie serwera sendmail 652
  - Kolejki pocztowe 654
  - Konfiguracja serwera sendmail 655
  - Preprocesor m4 655
  - Elementy konfiguracji serwera sendmail 656
  - Plik konfiguracyjny zbudowany z przykładowego pliku .mc 657
  - Elementy konfiguracji 658
  - Tabele i bazy danych 658
  - Makra i funkcje ogólnego zastosowania 659
  - Konfiguracja klienta 664
  - Opcje konfiguracyjne m4 665
  - Mechanizmy antyspamowe serwera sendmail 667
  - Serwer sendmail i bezpieczeństwo 670
  - Testowanie i diagnostyka serwera sendmail 676
- 18.9. Exim 678
  - Instalacja serwera Exim 679
  - Uruchamianie serwera Exim 681
  - Narzędzia serwera Exim 681
  - Język konfiguracji serwera Exim 682
  - Plik konfiguracyjny serwera Exim 683
  - Opcje globalne 684
  - ACL (ang. access control lists) 686
  - Skanowanie treści na etapie ACL 689
  - Mechanizmy uwierzytelniające 689
  - Routery 690
  - Transporty 693

- Konfiguracja ponowień 694
- Konfiguracja przepisywania 695
- Lokalna funkcja skanująca 695
- Zapisywanie dzienników 695
- Diagnostyka 696
- 18.10. Postfix 697
  - Architektura serwera Postfix 697
  - Bezpieczeństwo 699
  - Polecenia i dokumentacja serwera Postfix 699
  - Konfiguracja serwera Postfix 700
  - Domeny wirtualne 704
  - Kontrola dostępu 706
  - Diagnostyka 709
- 18.11. Zalecana literatura 710
  - Literatura na temat serwera sendmail 710
  - Literatura na temat serwera Exim 711
  - Literatura na temat serwera Postfix 711
  - Dokumenty RFC 711

## 19 Hosting WWW

**713**

- 19.1. Protokół HTTP 714
  - URL — jednolity lokalizator zasobu 715
  - Struktura transakcji HTTP 716
  - curl — HTTP z wiersza poleceń 718
  - Ponowne użycie połączenia TCP 719
  - HTTP przez TLS 720
  - Wirtualne hosty 720
- 19.2. Podstawy oprogramowania WWW 721
  - Serwery WWW i oprogramowanie pośredniczące w ruchu HTTP 722
  - Balansery obciążenia 723
  - Pamięć podręczna 725
  - Sieci dostarczania treści (CDN) 728
  - Języki sieci WWW 729
  - Interfejsy programowania aplikacji (API) 731
- 19.3. Hosting WWW w chmurze 733
  - Budowa kontra zakup 733
  - Platforma jako usługa 734
  - Hosting treści statycznych 735
  - Bezserwerowe aplikacje WWW 735
- 19.4. Apache httpd 736
  - httpd w praktyce 736
  - Ustawienia konfiguracyjne httpd 737

- Konfigurowanie hostów wirtualnych 739
- Rejestrowanie zdarzeń 742
- 19.5. NGINX 743
  - Instalacja i uruchamianie serwera NGINX 743
  - Konfigurowanie serwera NGINX 744
  - Konfigurowanie TLS dla serwera NGINX 747
  - Równoważenie obciążenia z serwerem NGINX 747
- 19.6. HAProxy 748
  - Kontrolowanie stanu serwera 749
  - Statystyki serwera 750
  - Lepkie sesje 750
  - Terminacja TLS 751
- 19.7. Zalecana literatura 752

### III. PAMIĘĆ MASOWA

## 20 Pamięć masowa

755

- 20.1. Chcę tylko dodać dysk! 756
  - Linux 757
  - FreeBSD 758
- 20.2. Urządzenia pamięci masowej 759
  - Dyski twarde 760
  - Dyski SSD 763
  - Dyski hybrydowe 766
  - Technologia Advanced Format i 4-kilobajtowe bloki 767
- 20.3. Interfejsy urządzeń pamięci masowej 768
  - Interfejs SATA 768
  - Interfejs PCI Express 768
  - Interfejs SAS 769
  - USB 770
- 20.4. Podłączanie i niskopoziomowa obsługa dysków 771
  - Weryfikacja instalacji na poziomie sprzętowym 771
  - Pliki urządzeń dyskowych 772
  - Formatowanie i zarządzanie uszkodzonymi blokami 773
  - Bezpieczne wymazywanie dysków ATA 774
  - hdparm i camcontrol — ustawianie parametrów dysku i interfejsu 775
  - Monitorowanie dysku twardego za pomocą SMART 776

- 20.5. Obieranie cebuli, czyli programowa strona pamięci masowej 777
  - Elementy systemu pamięci masowej 777
  - Mapper urządzeń w systemie Linux 779
- 20.6. Partycjonowanie dysków 780
  - Tradycyjne partycjonowanie 781
  - Partycje MBR 782
  - GPT — tablica partycji GUID 783
  - Partycjonowanie w systemie Linux 784
  - Partycjonowanie w systemie FreeBSD 784
- 20.7. Zarządzanie woluminami logicznymi 784
  - Zarządzanie woluminami logicznymi w systemie Linux 785
  - Zarządzanie woluminami logicznymi w systemie FreeBSD 790
- 20.8. RAID — nadmiarowa macierz niedrogich dysków 790
  - RAID programowy a sprzętowy 790
  - Poziomy RAID 791
  - Przywracanie dysku po awarii 794
  - Wady RAID 5 794
  - mdadm — programowy RAID w systemie Linux 795
- 20.9. Systemy plików 799
- 20.10. Tradycyjne systemy plików — UFS, ext4 i XFS 800
  - Terminologia systemu plików 801
  - Polimorfizm systemu plików 802
  - Formatowanie systemu plików 802
  - fsck — sprawdzanie i naprawa systemu plików 802
  - Montowanie systemu plików 804
  - Ustawianie automatycznego montowania 804
  - Montowanie napędów USB 807
  - Zalecenia dotyczące obszaru wymiany 807
- 20.11. Systemy plików następnej generacji: ZFS i Btrfs 808
  - Kopiowanie przy zapisie 808
  - Wykrywanie błędów 809
  - Wydajność 809
- 20.12. ZFS — rozwiązanie wszystkich problemów z pamięcią masową 810
  - ZFS w systemie Linux 810
  - Architektura ZFS 811
  - Przykład: dodawanie dysków 812
  - Systemy plików i ich właściwości 812
  - Dziedziczenie właściwości 814
  - Osobne systemy plików dla każdego użytkownika 815
  - Kopie migawkowe i klony 815
  - Surowe woluminy 816
  - Zarządzanie pulą pamięci masowej 817

- 20.13. Btrfs — ograniczona wersja ZFS dla systemu Linux 819
  - Btrfs kontra ZFS 819
  - Konfigurowanie i konwertowanie pamięci masowej 820
  - Woluminy i podwoluminy 822
  - Migawki woluminów 823
  - Płytkie kopie 823
- 20.14. Strategia tworzenia kopii zapasowych 824
- 20.15. Zalecana literatura 825

## 21 NFS

**827**

- 21.1. Sieciowe systemy plików 828
  - Współzawodnictwo 828
  - Kontrola stanu 829
  - Problemy wydajności 829
  - Bezpieczeństwo 830
- 21.2. NFS 830
  - Wersje protokołu 831
  - Zdalne wywoływanie procedur 832
  - Protokoły transportowe 832
  - Stan 832
  - Eksporty systemu plików 833
  - Blokowanie plików 834
  - Bezpieczeństwo 834
  - Odwzorowanie tożsamości w wersji 4. 836
  - Dostęp z uprawnieniami root i konto nobody 837
  - Wydajność w wersji 4. 838
- 21.3. Serwery NFS 838
  - Plik exports w Linuksie 839
  - Plik exports w systemie FreeBSD 841
  - Demon nfsd 842
- 21.4. NFS po stronie klienta 844
  - Montowanie zdalnych systemów plików podczas rozruchu systemu 846
  - Ograniczanie eksportów do uprzywilejowanych portów 847
- 21.5. Odwzorowanie tożsamości w NFSv4 847
- 21.6. Statystyki połączeń NFS — nfsstat 848
- 21.7. Dedykowane serwery plików NFS 848
- 21.8. Montowanie automatyczne 849
  - Odwzorowania pośrednie 851
  - Odwzorowania bezpośrednie 851



- Odwzorowania główne 851
- Odwzorowania wykonywalne 852
- Widoczność zasobów montowanych automatycznie 852
- Automount i replikowane systemy plików 853
- Automatyczne użycie mechanizmu automount  
(wersja 3., wszystkie systemy oprócz Linuksa) 854
- Specyfika Linuksa 854
- 21.9. Zalecana literatura 855

## 22 SMB

**857**

- 22.1. Samba — serwer SMB dla systemów Unix 858
- 22.2. Instalacja i konfigurowanie serwera Samba 859
  - Współdzielenie plików z uwierzytelnianiem lokalnym 860
  - Współdzielenie plików za pomocą kont uwierzytelnianych  
przez Active Directory 861
  - Konfigurowanie udziałów 861
- 22.3. Montowanie plików udostępnionych przez SMB 863
- 22.4. Przeglądanie plików udostępnionych przez SMB 864
- 22.5. Zapewnienie bezpieczeństwa Samby 865
- 22.6. Usuwanie problemów z systemem Samba 865
  - Sprawdzanie stanu Samby za pomocą smbstatus 865
  - Konfigurowanie rejestrowania zdarzeń w Sambie 866
  - Zarządzanie zestawami znaków 867
- 22.7. Zalecana literatura 868

## IV. OPERACJE

### 23 Zarządzanie konfiguracją

**871**

- 23.1. Zarządzanie konfiguracją w pigułce 872
- 23.2. Niebezpieczeństwa związane z zarządzaniem konfiguracją 873
- 23.3. Elementy zarządzania konfiguracją 873
  - Operacje i parametry 874
  - Zmienne 875
  - Fakty 876
  - Obsługa zmian 876

	Powiązania	876
	Paczki i repozytoria paczek	877
	Środowiska	877
	Ewidencjonowanie i rejestracja klientów	878
23.4.	Porównanie popularnych systemów CM	879
	Terminologia	880
	Modele biznesowe	880
	Opcje architekuralne	880
	Opcje językowe	883
	Opcje zarządzania zależnościami	884
	Ogólne uwagi na temat systemu Chef	886
	Ogólne uwagi na temat systemu Puppet	886
	Ogólne uwagi na temat systemów Ansible i Salt	887
	YAML	887
23.5.	Wprowadzenie do systemu Ansible	889
	Ansible na przykładzie	890
	Ustawienia klienta	892
	Grupy klientów	894
	Przypisywanie zmiennych	895
	Grupy dynamiczne i obliczane	895
	Listy zadań	896
	Parametry state	898
	Iteracja	898
	Interakcja z Jinja	899
	Generowanie szablonów	899
	Powiązania — akcje i scenariusze	900
	Role	902
	Zalecenia dotyczące ustrukturyzowania bazy konfiguracyjnej	903
	Opcje dostępu Ansible	904
23.6.	Wprowadzenie do systemu Salt	906
	Ustawianie sługi	908
	Powiązania wartości zmiennych dla sług	909
	Dopasowywanie sług	910
	Stany w systemie Salt	912
	Salt i Jinja	913
	Identyfikatory stanów i zależności	914
	Funkcje stanowe i wykonawcze	916
	Parametry i nazwy	917
	Powiązania stanów ze sługami	919
	Wysokie stany	920
	Formuły Salt	921
	Środowiska	921
	Mapa drogowa dokumentacji	925

- 23.7. Porównanie systemów Ansible i Salt 926
  - Elastyczność i skalowalność procesu wdrażania 926
  - Wbudowane moduły i rozszerzalność 927
  - Bezpieczeństwo 927
  - Różności 928
- 23.8. Wzorce postępowania 929
- 23.9. Zalecana literatura 931

## 24 Wirtualizacja

**933**

- 24.1. Terminologia wirtualizacji 934
  - Hipernadzorczy 934
  - Migracja w locie 937
  - Obrazy maszyn wirtualnych 937
  - Konteneryzacja 938
- 24.2. Wirtualizacja w Linuksie 939
  - Xen 939
  - Instalacja gości w Xen 940
  - KVM 942
  - Instalacja gości w KVM 942
- 24.3. Bhyve w systemie FreeBSD 943
- 24.4. VMware 943
- 24.5. VirtualBox 944
- 24.6. Packer 944
- 24.7. Vagrant 946
- 24.8. Zalecana literatura 947

## 25 Kontenery

**949**

- 25.1. Pojęcia ogólne i podstawowe 950
  - Obsługa przez jądro 951
  - Obrazy 951
  - Sieć 952
- 25.2. Docker — silnik kontenerowy typu open source 953
  - Podstawowa architektura 953
  - Instalacja 955
  - Konfigurowanie klienta 955
  - Praca z kontenerem 956
  - Woluminy 959

- Kontenery danych 960
- Sieci w Dockerze 961
- Sterowniki pamięci masowej 963
- Opcje dockerd 964
- Budowanie obrazów 966
- Repozytoria 969
- 25.3. Kontenery w praktyce 971
  - Rejestrowanie zdarzeń 972
  - Porady dotyczące bezpieczeństwa 972
  - Rozwiązywanie problemów i usuwanie błędów 975
- 25.4. Grupowanie kontenerów i zarządzanie nimi 976
  - Krótki przegląd oprogramowania do zarządzania kontenerami 977
  - Kubernetes 977
  - Mesos i Marathon 978
  - Docker Swarm 979
  - ECS — obsługa kontenerów EC2 w AWS 980
- 25.5. Zalecana literatura 981

## 26 Ciągła integracja i ciągłe dostarczanie

983

- 26.1. Podstawy CI/CD 985
  - Zasady i praktyki 985
  - Środowiska 988
  - Przełączniki funkcji 989
- 26.2. Potoki 990
  - Proces budowania 990
  - Testowanie 991
  - Wdrażanie 993
  - Techniki wdrażania bez przestojów 994
- 26.3. Jenkins — serwer automatyzacji typu open source 995
  - Podstawowe pojęcia związane z Jenkinsem 995
  - Rozproszone procesy budowania 997
  - Potok jako kod 997
- 26.4. CI/CD w praktyce 998
  - UlsahGo, trywialna aplikacja internetowa 999
  - Testowanie jednostkowe UlsahGo 1000
  - Pierwsze kroki z potokiem Jenkinsa 1001
  - Budowanie obrazu DigitalOcean 1003
  - Zapewnienie pojedynczego systemu do testowania 1005
  - Testowanie kropli 1008
  - Wdrażanie UlsahGo do pary kropli i balansera obciążenia 1008
  - Zamknięcie potoku demonstracyjnego 1010

- 26.5. Kontenery a CI/CD 1010
  - Kontenery jako środowisko budowania 1011
  - Obrazy kontenerów jako artefakty budowania 1011
- 26.6. Zalecana literatura 1012

## 27 Bezpieczeństwo

**1013**

- 27.1. Elementy bezpieczeństwa 1015
- 27.2. Drogi do naruszenia bezpieczeństwa 1015
  - Socjotechnika 1015
  - Podatności oprogramowania 1016
  - Rozproszona odmowa usługi (DDoS) 1017
  - Nadużycia wewnętrzne 1018
  - Błędy konfiguracji sieci, systemu lub aplikacji 1018
- 27.3. Podstawowe środki bezpieczeństwa 1019
  - Aktualizacje oprogramowania 1019
  - Zbędne usługi 1020
  - Zdalne logowanie zdarzeń 1021
  - Kopie zapasowe 1021
  - Wirusy i robaki 1021
  - Rootkity 1022
  - Filtrowanie pakietów 1022
  - Hasła i uwierzytelnianie wieloskładnikowe 1023
  - Czułość 1023
  - Testy penetracyjne aplikacji 1024
- 27.4. Hasła i konta użytkowników 1024
  - Zmiany haseł 1025
  - Menedżery haseł 1025
  - Okres ważności haseł 1027
  - Konta współużytkowane 1027
  - Programy powłoki 1028
  - Użytkownicy typu root 1028
- 27.5. Narzędzia bezpieczeństwa 1028
  - Skaner portów sieciowych nmap 1028
  - Nessus — skaner sieciowy następnej generacji 1030
  - Metasploit — oprogramowanie do testów penetracyjnych 1031
  - Lynis — podręczny audyt bezpieczeństwa 1031
  - Wyszukiwanie słabych haseł — John the Ripper 1031
  - Programowalny system wykrywania włamań sieciowych — Bro 1032
  - Popularny system wykrywania włamań — Snort 1033
  - Wykrywanie włamań na poziomie hosta — OSSEC 1033
  - Fail2Ban — system reagowania na ataki brute-force 1036

- 27.6. Narzędzia kryptograficzne 1036
  - Kryptografia klucza symetrycznego 1037
  - Kryptografia klucza publicznego 1037
  - Infrastruktura klucza publicznego 1038
  - TLS 1040
  - Kryptograficzne funkcje skrótu 1040
  - Generowanie liczb losowych 1042
  - Wybór oprogramowania kryptograficznego 1043
  - Polecenie openssl 1043
  - PGP — Pretty Good Privacy 1045
  - Kerberos — zunifikowane podejście do bezpieczeństwa sieciowego 1046
- 27.7. Bezpieczna zdalna powłoka SSH 1046
  - Podstawowe elementy OpenSSH 1047
  - Klient ssh 1048
  - Uwierzytelnianie za pomocą klucza publicznego 1050
  - ssh-agent 1051
  - Aliasy hostów w pliku ~/.ssh/config 1052
  - Multipleksacja połączeń 1053
  - Przekierowywanie portów 1054
  - sshd — serwer OpenSSH 1055
  - Weryfikacja klucza hosta za pomocą SSHFP 1056
  - Przesyłanie plików 1057
  - Inne metody bezpiecznego logowania 1057
- 27.8. Zapory sieciowe 1058
  - Zapory filtrujące pakiety 1058
  - Filtrowanie usług 1058
  - Zapory z kontrolą stanu 1059
  - Poziom bezpieczeństwa oferowany przez zapory sieciowe 1059
- 27.9. VPN (ang. Virtual Private Network) 1060
  - Tunelowanie IPsec 1060
  - Czy sam VPN wystarczy? 1061
- 27.10. Certyfikacja i standardy 1061
  - Certyfikacja 1061
  - Standardy bezpieczeństwa 1062
- 27.11. Źródła informacji o bezpieczeństwie 1064
  - SecurityFocus.com oraz listy dyskusyjne BugTraq i OSS 1065
  - Schneier on Security 1065
  - Raport firmy Verizon z dochodzeń w sprawach dotyczących naruszenia danych 1065
  - Instytut SANS 1065
  - Źródła związane z poszczególnymi dystrybucjami 1066
  - Inne listy e-mailowe i strony WWW 1066
- 27.12. Reakcja na atak 1066
- 27.13. Zalecana literatura 1068

## 28 Monitoring

1069

- 28.1. Przegląd monitoringu 1070
  - Instrumentacja 1071
  - Rodzaje danych 1071
  - Pobieranie i przetwarzanie 1072
  - Powiadomienia 1072
  - Panele i interfejsy użytkownika 1073
- 28.2. Kultura monitoringu 1073
- 28.3. Platformy monitorujące 1074
  - Platformy czasu rzeczywistego typu open source 1075
  - Platformy szeregów czasowych typu open source 1076
  - Platformy open source do tworzenia wykresów 1078
  - Komercyjne platformy monitorujące 1079
  - Hostowane platformy monitorujące 1079
- 28.4. Zbieranie danych 1080
  - StatsD — ogólny protokół przesyłania danych 1080
  - Pozyskiwanie danych z wyjścia poleceń 1082
- 28.5. Monitorowanie sieci 1083
- 28.6. Monitorowanie systemów 1084
  - Polecenia dla systemów monitorowania 1085
  - collectd — pozyskiwanie ogólnych danych systemowych 1086
  - sysdig i dtrace — śledzenie działań w systemie 1086
- 28.7. Monitorowanie aplikacji 1087
  - Monitorowanie dzienników 1087
  - Supervisor + Munin — proste rozwiązanie dla ograniczonych zastosowań 1088
  - Komercyjne narzędzia do monitorowania aplikacji 1088
- 28.8. Monitorowanie bezpieczeństwa 1089
  - Weryfikowanie integralności systemu 1089
  - Monitorowanie wykrywania włamań 1090
- 28.9. Protokół SNMP 1091
  - Organizacja SNMP 1092
  - Operacje protokołu SNMP 1093
  - Net-SNMP — narzędzia dla serwerów 1093
- 28.10. Kruczki i sztuczki 1095
- 28.11. Zalecana literatura 1096

**29 Wydajność****1097**

- 29.1. Filozofia dostrajania wydajności 1098
- 29.2. Metody poprawy wydajności 1099
- 29.3. Czynniki wpływające na wydajność 1101
- 29.4. Zabieranie cykli procesora 1102
- 29.5. Analizowanie problemów z wydajnością 1102
- 29.6. Kontrola wydajności systemu 1103
  - Inwentaryzacja sprzętu 1103
  - Gromadzenie danych o wydajności 1105
  - Analiza użycia procesora 1106
  - Zarządzanie pamięcią przez system 1108
  - Analiza użycia pamięci 1109
  - Analiza obciążenia wejścia-wyjścia 1111
  - Testowanie wydajności podsystemu dyskowego — program fio 1112
  - Gromadzenie statystyk w czasie i budowanie raportów — program sar 1113
  - Wybór planisty operacji wejścia-wyjścia w Linuksie 1113
  - Szczegółowe profilowanie systemu Linux — program perf 1114
- 29.7. Pomocy! Mój system nagle bardzo zwolnił! 1115
- 29.8. Zalecana literatura 1117

**30 Podstawy centrów danych****1119**

- 30.1. Szafy 1120
- 30.2. Zasilanie 1121
  - Wymagania zasilania szaf 1122
  - Jednostki mocy — kVA a kW 1123
  - Wydajność energetyczna 1123
  - Pomiary 1124
  - Koszt 1124
  - Zdalne sterowanie 1124
- 30.3. Chłodzenie i środowisko 1124
  - Szacowanie zapotrzebowania na chłodzenie 1125
  - Gorące i zimne korytarze 1126
  - Wilgotność 1128
  - Monitorowanie środowiska 1128
- 30.4. Poziomy niezawodności centrów danych 1129
- 30.5. Bezpieczeństwo centrów danych 1129
  - Lokalizacja 1130
  - Ogrodzenie 1130



- Dostęp do obiektu 1130
- Dostęp do szaf 1130
- 30.6. Narzędzia 1131
- 30.7. Zalecana literatura 1132

## **31 Metodologia i reguły w IT 1133**

- 31.1. Teoria wielkiej unifikacji — DevOps 1134
  - Zasady DevOps 1135
  - Administracja systemem w świecie DevOps 1138
- 31.2. Rejestracja zgłoszeń i system zarządzania zgłoszeniami 1139
  - Funkcje systemów zgłoszeniowych 1140
  - Przydzielanie zgłoszeń 1140
  - Akceptacja systemów zgłoszeniowych przez użytkowników 1141
  - Przykłady systemów zgłoszeniowych 1142
  - Przydzielanie zgłoszeń 1143
- 31.3. Utrzymanie lokalnej dokumentacji 1143
  - Infrastruktura jako kod 1144
  - Standaryzacja dokumentacji 1144
- 31.4. Utrzymanie niezależnych środowisk 1146
- 31.5. Przywracanie systemu po katastrofie 1147
  - Ocena ryzyka 1147
  - Plan naprawy 1148
  - Zespół do zwalczania skutków katastrof 1149
  - Incydenty bezpieczeństwa 1150
- 31.6. Reguły i procedury 1151
  - Różnice między regułami i procedurami 1151
  - Najlepsze praktyki tworzenia reguł 1152
  - Procedury 1152
- 31.7. Definiowanie poziomu usług (SLA) 1153
  - Zakresy i opisy usług 1154
  - Reguły ustalania priorytetów zadań 1155
- 31.8. Zgodność — regulacje i standardy 1156
- 31.9. Zagadnienia prawne 1159
  - Ochrona prywatności 1159
  - Wymuszanie stosowania reguł 1160
  - Kontrola = odpowiedzialność 1160
  - Licencje na oprogramowanie 1161
- 31.10. Organizacje, konferencje i inne zasoby 1162
- 31.11. Zalecana literatura 1163

<b>Krótką historia administracji systemami</b>	<b>1165</b>
<b>Kolofon</b>	<b>1175</b>
<b>O współpracownikach</b>	<b>1177</b>
<b>O autorach</b>	<b>1179</b>
<b>Skorowidz</b>	<b>1181</b>

# 28

## MONITORING



Dążenie do monitorowania jest cechą wyróżniającą profesjonalnego administratora systemów. Niedoświadczeni administratorzy często pozostawiają systemy bez nadzoru i pozwalają na „wykrywanie” usterek, gdy sfrustrowany, rozgniewany użytkownik dzwoni do działu pomocy technicznej, ponieważ nie jest w stanie wykonać powierzonego mu zadania. Nieco bardziej uświadomione zespoły administracyjne tworzą platformę monitorowania, ale wyłączają powiadomienia po godzinach pracy, ponieważ są one zbyt uciążliwe. W obu przypadkach następuje walka z pożarem i ośmieszenie. Takie podejścia mają negatywny wpływ na przedsiębiorstwo, komplikują działania naprawcze i narażają administratorów na krytykę.

Profesjonalni administratorzy systemów przyjmują monitoring za swoją religię. Każdy system przed uruchomieniem jest dodawany do platformy monitorującej, a zestaw narzędzi kontrolnych jest regularnie testowany i dostrajany. Mierniki i trendy są oceniane proaktywnie, tak aby można było wykryć problemy, zanim dotkną one użytkowników lub wywołają zagrożenie bezpieczeństwa danych.

Duży serwis strumieniowania wideo on-line, o którym zapewne słyszałeś, ceni swój system telemetrii tak bardzo, że prędzej dopuszczono by tam do przestoju w świadczeniu usługi niż przestoju w monitorowaniu. Bez monitoringu i tak byłoby wiadomo, co się stało.

Filozofia „najpierw monitoring” (wraz z towarzyszącymi mu narzędziami) uczyni Cię superbohaterem wśród administratorów systemów. Dzięki niej lepiej poznasz swoje oprogramowanie i aplikacje, sprawniej rozwiążesz małe problemy, zanim przerodzą się w katastrofalne w skutkach awarie, staniesz się bardziej efektywny w znajdowaniu warunków wystąpienia błędów i usuwaniu problemów, a także zrozumiesz kwestię wydajności złożonych systemów. Monitorowanie poprawi również jakość Twojego życia, pozwalając na rozwiązanie większości problemów w dogodnym dla Ciebie czasie, a nie o trzeciej nad ranem w okresie świąt Bożego Narodzenia.

## 28.1. PRZEGLĄD MONITORINGU

Celem monitorowania jest zapewnienie funkcjonowania infrastruktury IT jako całości zgodnie z oczekiwaniami oraz zebranie, w przystępnej i łatwo przyswajalnej formie, danych przydatnych do zarządzania i planowania. Proste, prawda? Ten ogólny opis obejmuje jednak potencjalnie dość rozległy obszar.

Rzeczywiste systemy monitoringu różnią się w każdym możliwym wymiarze, ale wszystkie mają tę samą podstawową strukturę:

- z systemów i urządzeń będących przedmiotem zainteresowania pozyskiwane są surowe dane;
- platforma monitorująca dokonuje przeglądu danych i określa, jakie działania będą odpowiednie, zazwyczaj poprzez stosowanie administracyjnie określonych reguł;
- surowe dane i wszelkie decyzje podjęte przez system monitorowania przesyłane są na zaplecze, gdzie prowadzone są odpowiednie działania.

Rzeczywiste systemy monitoringu bywają zarówno trywialnie proste, jak i niezwykle skomplikowane. Przykładowo poniższy skrypt napisany w Perlu zawiera wszystkie wymienione wyżej elementy:

```
#!/usr/bin/env perl
$loadavg = (split /\s,/, `uptime`)[10];
# Jeśli obciążenie jest większe niż 5, powiadom administratora
if ($loadavg > 5.0) {
    system 'mail -s "Obciążenie serwera jest za wysokie" dan@admin.com < /dev/null'
}
```

Skrypt ten uruchamia polecenie `uptime` w celu uzyskania średnich obciążeń systemu. Jeśli średnia wartość jednogodzinowego obciążenia jest większa niż 5.0, skrypt wysyła wiadomość e-mail do administratora. Dane, ocena, reakcja.

Dawno, dawno temu „wyszukana” konfiguracja monitoringu składała się z zestawu takich skryptów, które były uruchamiane przez demona `cron` i nakazywały modemu wysyłanie wiadomości na pagery administratorów. Obecnie na każdym etapie monitorowania potoku dostępnych jest wiele opcji.

Oczywiście nadal możesz pisać skrypty monitorujące i uruchamiać je przez program cron. Jeśli to naprawdę wszystko, czego potrzebujesz, za wszelką cenę staraj się zachować prostotę. Jeżeli jednak jesteś odpowiedzialny za więcej niż dwa serwery, takie doraźne podejście zwykle nie będzie wystarczające.

W dalszej części tego podrozdziału przyjrzymy się etapom potoku nieco dokładniej.

## Instrumentacja

Szeroki zakres danych, które mogą się okazać użyteczne dla Twojej organizacji, obejmuje parametry wydajnościowe (czas odpowiedzi, wykorzystanie, szybkość transferu), parametry dotyczące dostępności (dostępność i nieprzerwany czas pracy), pojemność, zmiany stanu, wpisy dzienników, a nawet biznesowe wskaźniki efektywności, takie jak średnia wartość koszyka zakupów lub współczynnik konwersji kliknięć.

Ponieważ wszystko, co można zrobić na komputerze, jest potencjalnym przedmiotem monitorowania, systemy monitoringu są zazwyczaj agnostyczne, jeśli chodzi o źródła danych. Często są one dostarczane z wbudowanym wsparciem dla różnych elementów wejściowych. Nawet źródła danych, które nie są obsługiwane bezpośrednio, mogą być wprowadzane za pomocą kilku wierszy kodu dostosowawczego lub oddzielnej bramki danych, takiej jak StatsD (patrz podrozdział 28.4).

Przy tak dużej ilości danych, które aż się proszą o to, żeby je zbierać, trudną częścią projektowania systemu zbierania danych może być zdecydowanie o tym, co należy ignorować. Unikaj gromadzenia danych, które nie mają jasnego i możliwego do zrealizowania celu. Nadmiarowe zbieranie danych obciąża zarówno system monitoringu, jak i monitorowane jednostki. Prowadzi również do ukrywania wartości, które są naprawdę ważne, lecz giną w gąszczu informacji.

Niestety, często nie jest łatwo odróżnić użyteczne dane od śmieci. Musisz stale na nowo oceniać to, co jest monitorowane, i zastanawiać się nad tym, jak dane te będą się zachowywać przez cały okres życia systemu.

## Rodzaje danych

Na najwyższym poziomie dane monitoringu można podzielić na trzy ogólne kategorie:

- **Wskaźniki czasu rzeczywistego**, charakteryzujące stan działania środowiska. Są to zazwyczaj liczby lub wartości logiczne. Ogólnie rzecz biorąc, obowiązkiem systemu monitorowania jest przetestowanie tych wskaźników w odniesieniu do oczekiwań i wygenerowanie alarmu w przypadku, gdy bieżąca wartość wykroczy poza określony wcześniej zakres lub próg.
- **Zdarzenia**, które często przybierają formę wpisów w plikach dziennika lub powiadomień typu „push” z podsystemów. Zdarzenia te, zwane czasami wskaźnikami opartymi na wzorcu, mogą sygnalizować zmianę stanu, stan alarmowy bądź inne działanie. Zdarzenia mogą być przetwarzane na dane numeryczne (np. sumę lub szybkość) lub bezpośrednio wywoływać odpowiedzi monitorujące<sup>1</sup>.
- **Zagregowane i podsumowane trendy historyczne**, które są często zbiorami uszeregowanych czasowo wskaźników czasu rzeczywistego. Pozwalają one na analizę i wizualizację zmian w czasie.

---

<sup>1</sup> Wiele punktów danych gromadzonych przez oprogramowanie monitorujące aplikację należy do kategorii „zdarzenie”, czasami dołączone są również dane ilościowe. Wzajemne powiązania pomiędzy zdarzeniami (np. „użytkownik przeglądał stronę *Ustawienia*, a następnie kliknął *Anuluj*, niczego nie zmieniając”) są często pomocne w badaniu. Platformy monitorowania ogólnego przeznaczenia nie są zbyt dobre w przypadku tego rodzaju powiązań, co jest jednym z powodów, dla których monitorowanie aplikacji stało się całkiem osobną kategorią.

## Pobieranie i przetwarzanie

Większość systemów monitoringu opiera się na centralnej platformie monitorującej, która absorbuje dane z monitorowanych systemów, przeprowadza odpowiednie przetwarzanie i stosuje reguły administracyjne w celu określenia, co powinno nastąpić w odpowiedzi.

Platformy pierwszej generacji, takie jak Nagios i Icinga, skupiały się na wykrywaniu pojawiających się problemów i reagowaniu na nie. Systemy te były rewolucyjne w swoich czasach i wprowadziły nas we współczesny świat monitoringu. Niemniej jednak z biegiem lat zostały one przyćmione przez rozwijającą się stopniowo w branży świadomość, że wszystkie monitorowane dane można uszeregować czasowo. Gdyby wartości nie różniły się od siebie, nie monitorowałbyś ich.

Oczywiste jest, że potrzebne było podejście bardziej ukierunkowane na dane. Jednak monitorowane dane są zazwyczaj tak obszerne, że nie można po prostu przenieść ich wszystkich do tradycyjnej bazy danych i pozwolić sobie na ich gromadzenie. Jest to recepta na niską wydajność i przepełnienie dysków.

Nowoczesne podejście polega na zorganizowaniu monitoringu wokół magazynu danych, który specjalizuje się w obsłudze danych uszeregowanych czasowo. Przez początkowy okres przechowywane są wszystkie dane, ale w miarę ich starzenia się magazyn stosuje coraz wyższe poziomy podsumowywania w celu ograniczenia wymagań odnośnie do pamięci masowej. Przykładowo magazyn może zachowywać dane z okresu godziny w rozdzielczości jednej sekundy, dane tygodniowe w rozdzielczości jednej minuty, a dane roczne w rozdzielczości godzinowej.

Dane historyczne są przydatne nie tylko do prezentacji na panelu, ale również jako punkt odniesienia dla porównań, np. czy obecny poziom błędów sieci przekracza średnią historyczną o 25% lub więcej.

## Powiadomienia

Gdy masz już platformę monitorującą, zastanów się dokładnie, co zrobić z wynikami monitorowania. Priorytetem jest zazwyczaj poinformowanie administratorów i programistów o problemie, który wymaga uwagi.

Powiadomienia muszą się wiązać z podjęciem działań. Zorganizuj swój system monitorowania tak, aby każdy, kto otrzyma powiadomienie, musiał potencjalnie coś zrobić w odpowiedzi, nawet jeśli działanie jest czymś tak ogólnym jak „sprawdzić później, aby się upewnić, że to zostało zrobione”. Powiadomienia, które są czysto informacyjne, wykształcają w personelu nawyk ich ignorowania.

W większości przypadków powiadomienia muszą wykraczać poza pocztę elektroniczną, aby były optymalnie skuteczne. W razie wystąpienia krytycznych problemów łatwym i wydajnym rozwiązaniem będą powiadomienia SMS (tj. wiadomości tekstowe) wysyłane na telefony komórkowe administratorów. Odbiorcy mogą ustawić dzwonek i głośność telefonów tak, aby w razie potrzeby obudziły ich w środku nocy.

Powiadomienia powinny być również zintegrowane z implementacją ChatOps Twojego zespołu. Mniej krytyczne powiadomienia (takie jak statusy zadań, nieudane logowania i powiadomienia informacyjne) mogą być wysyłane do jednego lub kilku czatów, tak aby zainteresowane strony mogły otrzymywać podzbiory powiadomień, które mogłyby je interesować.

➤ *Więcej komentarzy na temat ChatOps można znaleźć w podrozdziale 31.1.*

Poza tymi podstawowymi kanałami możliwości powiadamiania są właściwie nieograniczone. Przykładowo system oświetlenia LED zmieniający kolory w zależności od stanu systemu może być przydatny do szybkiego wskazywania stanu w centrum danych lub sieciowym centrum operacyjnym. Inne opcje reagowania na sytuacje sygnalizowane przez systemy monitorowania obejmują m.in.:

- zautomatyzowane działania, takie jak zrzucanie baz danych lub rotacja dzienników;
- telefon do administratora;
- zamieszczenie danych na tablicy ściennej w celu ich upublicznienia;
- przechowywanie uszeregowanych czasowo danych w bazie do późniejszej analizy;
- nierobienie niczego i umożliwienie samemu systemowi dokonania późniejszego przeglądu.

## Panele i interfejsy użytkownika

Oprócz ostrzegania o wyraźnie wyjątkowych okolicznościach jednym z głównych celów monitoringu jest przedstawienie stanu środowiska w sposób bardziej uporządkowany i łatwiejszy do przyswojenia niż szereg surowych danych. Takie wyświetlacze nazywane są ogólnie panelami (ang. *dashboards*).

Panele nawigacyjne są projektowane przez administratorów lub innych członków organizacji zainteresowanych konkretnymi aspektami środowiska. Panele wykorzystują kilka różnych technik przemiany surowych danych w infograficzne złoto.

Po pierwsze, są selektywne w tym, co prezentują. Koncentrują się na najważniejszych wskaźnikach dla danej dziedziny, ukazujących ogólny stan lub ogólną wydajność. Po drugie, dają wskazówki kontekstowe dotyczące znaczenia i importu prezentowanych danych. Przykładowo problematyczne liczby i stany są zazwyczaj wyświetlane na czerwono, a podstawowe wskaźniki są przedstawiane za pomocą czcionki o większych rozmiarach. Związki pomiędzy wartościami są prezentowane poprzez grupowanie. Po trzecie, pulpity wyświetlają serie danych jako wykresy, co ułatwia ich szybką ocenę.

Oczywiście, większość zebranych danych nigdy nie pojawia się na panelu. Dobrze jest też mieć system monitoringu, który posiada uniwersalny interfejs użytkownika, ułatwiający przeglądanie i modyfikowanie schematu danych, umożliwiając wykonywanie dowolnych zapytań bazodanowych oraz tworzenie wykresów dowolnie definiowanych sekwencji danych w locie.

## 28.2. KULTURA MONITORINGU

Ten rozdział dotyczy głównie narzędzi, ale kultura jest co najmniej równie ważna jak one. Wyruszając w podróż po świecie monitoringu, kieruj się poniższymi zasadami:

- Jeśli dany system lub dana usługa kogoś dotyczą lub ktoś jest od nich zależny, musi to być monitorowane. Koniec i kropka. Nic w środowisku, od którego zależy usługa lub użytkownik, nie może pozostać bez nadzoru.
- Jeżeli urządzenie produkcyjne, system lub usługa posiada cechy możliwe do monitorowania, powinny one być monitorowane. Nie pozwól, aby serwer z wymyślnym interfejsem zarządzania sprzętem spędził tygodnie na bezskutecznych próbach powiadomienia Cię o awarii wentylatora.
- Wszystkie konstrukcje o wysokiej dostępności muszą być monitorowane. Szkoda by się było dowiedzieć, że główny serwer nie działa, dopiero po tym, gdy awarii uległby także serwer rezerwowo.

- Monitorowanie nie jest opcjonalne. Każdy administrator systemów, deweloper, członek personelu operacyjnego, kierownik i kierownik projektu powinien w swoim planie pracy zarezerwować czas na monitorowanie.
- Dane z monitoringu (zwłaszcza dane historyczne) są przydatne dla wszystkich. Spraw, aby były one łatwo dostępne i widoczne, tak by każdy mógł z nich korzystać podczas analizy głównych przyczyn awarii, planowania, zarządzania cyklem życia systemów i wprowadzania ulepszeń architektonicznych. Nie szczędź swoich wysiłków ani zasobów na tworzenie i promowanie pulpitów monitorujących.
- Każdy powinien odpowiadać na ostrzeżenia. Monitorowanie nie jest problemem tylko pracowników operacyjnych. Wszystkie role techniczne powinny otrzymywać powiadomienia i współpracować przy rozwiązywaniu problemów. Takie podejście zachęca do rzetelnego analizowania ich przyczyn, niezależnie od tego, które osoby najbardziej nadają się do rozwiązania problemu.
- Poprawnie wdrożone monitorowanie wpływa pozytywnie na jakość życia administratora. Solidny system monitorowania uwalnia Cię od zamartwiania się o to, w jakim stanie są Twoje systemy, i daje innym możliwość wspierania Cię. Bez monitorowania i odpowiedniej dokumentacji będziesz musiał czuwać przy telefonie praktycznie 24 godziny na dobę przez 7 dni w tygodniu.
- Przeszkól osoby reagujące na sytuacje alarmowe, aby je naprawiały, a nie tylko łagodziły. Zidentyfikuj fałszywe lub zbyt natrączywe alarmy i dostosuj je tak, aby przestały się wyzwać nieprawidłowo. Fałszywe alarmy zachęcają wszystkich do ignorowania systemu monitoringu.

## 28.3. PLATFORMY MONITORUJĄCE

Jeśli masz zamiar monitorować wiele systemów i więcej niż kilka wskaźników, warto poświęcić trochę czasu na wdrożenie pełnoprawnej platformy monitorującej. Jest to system ogólnego przeznaczenia, który zbiera dane z wielu źródeł, ułatwia wyświetlanie i podsumowywanie informacji o stanie oraz wprowadza ustandaryzowany sposób definiowania działań i ostrzeżeń.

Dobra wiadomość jest taka, że istnieje wiele możliwości do wyboru. Gorszą wiadomością jest to, że nie istnieje — jak na razie — jedna, doskonała platforma. Wybierając któreś z dostępnych rozwiązań, należy wziąć pod uwagę następujące kwestie:

- **Elastyczność gromadzenia danych.** Każda platforma może pobierać dane z różnych źródeł. Nie oznacza to jednak, że wszystkie platformy są pod tym względem równoważne. Pomyśl, z jakich źródeł danych rzeczywiście chcesz korzystać. Czy będziesz musiał odczytywać dane z bazy danych SQL? Z rekordów DNS? Z połączenia HTTP?
- **Jakość interfejsu użytkownika.** Wiele systemów oferuje konfigurowalne graficzne interfejsy użytkownika lub interfejsy WWW. Większość dobrze sprzedających się pakietów zachwala swoją zdolność wykorzystania szablonów JSON do prezentacji danych. Interfejs użytkownika to nie tylko szum marketingowy; potrzebujesz interfejsu, który przekaże informacje w sposób jasny, prosty i zrozumiały. Może w Twojej organizacji potrzebujesz różnych interfejsów użytkownika dla różnych grup?



- **Koszt.** Niektóre komercyjne pakiety zarządzania są bardzo kosztowne. Wiele korporacji ceni sobie to, że ich witryna jest zarządzana przez wysokiej klasy system komercyjny. Jeśli nie jest to tak ważne dla Twojej organizacji, przyjrzyj się darmowym rozwiązaniom, takim jak Zabbix, Sensu, Cacti i Icinga.
- **Automatyczne wykrywanie.** Wiele systemów oferuje możliwość „wykrywania” Twojej sieci. Poprzez rozesyłanie poleceń ping, żądania SNMP, przeszukiwania tablic ARP i zapytania DNS identyfikują one wszystkie lokalne hosty i urządzenia. Wszystkie wdrożenia wykrywające, jakie widzieliśmy, działały całkiem dobrze, ale ich dokładność jest mniejsza w sieciach złożonych lub silnie zabezpieczonych zaparami.
- **Funkcje sprawozdawcze.** Wiele produktów może wysyłać powiadomienia e-mail, integrować się z ChatOps, wysyłać wiadomości tekstowe i automatycznie generować zgłoszenia dla popularnych systemów śledzenia błędów. Upewnij się, że wybrana przez Ciebie platforma umożliwi elastyczne raportowanie. Kto wie, z jakimi urządzeniami elektronicznymi będziesz miał do czynienia za kilka lat?

## Platformy czasu rzeczywistego typu open source

Chociaż opisywane tu platformy — Nagios, Icinga i Sensu Core — robią wszystkiego po trochu, są znane z tego, że świetnie sobie radzą z pomiarami chwilowymi (lub opartymi na wskaźnikach).

Systemy te mają swoich zwolenników, ale jako narzędzia monitorowania pierwszej generacji stopniowo tracą przewagę nad systemami szeregów czasowych, które opisujemy w następnym punkcie. Większości ośrodków rozpoczynających od zera radzilibyśmy wybranie systemu opartego na szeregach czasowych.

### Nagios i Icinga

Nagios i Icinga specjalizują się w powiadamianiu o błędach w czasie rzeczywistym. Chociaż systemy te nie pomagają w określeniu, o ile zwiększyło się wykorzystanie przepustowości łącza w ciągu ostatniego miesiąca, mogą jednak wykryć, że Twój serwer WWW przeszedł w tryb off-line.

Nagios i Icinga były pierwotnie odnogami jednego drzewa źródłowego, ale współczesna Icinga 2 została przepisana zupełnie na nowo. Pod wieloma względami pozostaje jednak kompatybilna z Nagiosem.

Oba systemy zawierają mnóstwo skryptów do monitorowania usług we wszystkich kształtach i rozmiarach oraz posiadają rozbudowane możliwości monitoringu SNMP. Prawdopodobnie ich największą zaletą jest modułowy i wysoce konfigurowalny system konfiguracji, który pozwala na pisanie własnych skryptów do monitorowania wszelkich możliwych wskaźników.

Jeśli czujesz się na siłach i masz skłonności masochistyczne, możesz napisać własne wtyczki monitorujące w Perlu, PHP, Pythonie czy nawet w C. Wiele standardowych metod powiadamiania jest wbudowanych: poczta elektroniczna, raporty WWW, wiadomości tekstowe itp. I podobnie jak w przypadku wtyczek monitorujących, możesz łatwo wprowadzić własne skrypty powiadomień i akcji.

Zarówno Nagios, jak i Icinga dobrze się sprawdzają w sieciach obsługujących mniej niż tysiąc hostów i urządzeń. Systemy te są łatwe do dostosowania i rozszerzania, a także zawierają potężne funkcje, takie jak nadmiarowość, zdalne monitorowanie i eskalacja powiadomień.

Jeśli instalujesz nową infrastrukturę monitoringu od podstaw, polecamy raczej Icingę 2 niż Nagiosa. Jej baza kodu jest na ogół czystsza i szybko zyskuje zwolenników oraz wsparcie społeczności. Z funkcjonalnego punktu widzenia interfejs użytkownika Icingi jest bardziej przejrzysty i szybszy, a także jest w stanie automatycznie budować zależności usług, co może mieć zasadnicze znaczenie w złożonych środowiskach.

## Sensu

Sensu jest pełnowymiarową platformą monitorującą, dostępną zarówno w wersji open source (Sensu Core), jak i z płatnymi, komercyjnymi dodatkami. Posiada ultranowoczesny interfejs użytkownika i może uruchamiać wszelkie wtyczki monitorujące pochodzące z systemów Nagios, Icinga i Zabbix. Sensu zostało zaprojektowane jako zamiennik Nagiosa, więc kompatybilność z wtyczkami jest jedną z jego najbardziej atrakcyjnych cech. Sensu umożliwia łatwą integrację z powiadomieniami narzędzi Logstash i Slack, a jego instalacja jest wyjątkowo prosta.

## Platformy szeregów czasowych typu open source

Wykrywanie bieżących problemów i reagowanie na nie to tylko jeden z aspektów monitoringu. Często równie ważna jest znajomość zmian wartości w czasie i ich związków z innymi wartościami. W celu zaspokojenia tej potrzeby powstały cztery popularne platformy szeregów czasowych: Graphite, Prometheus, InfluxDB i Munin.

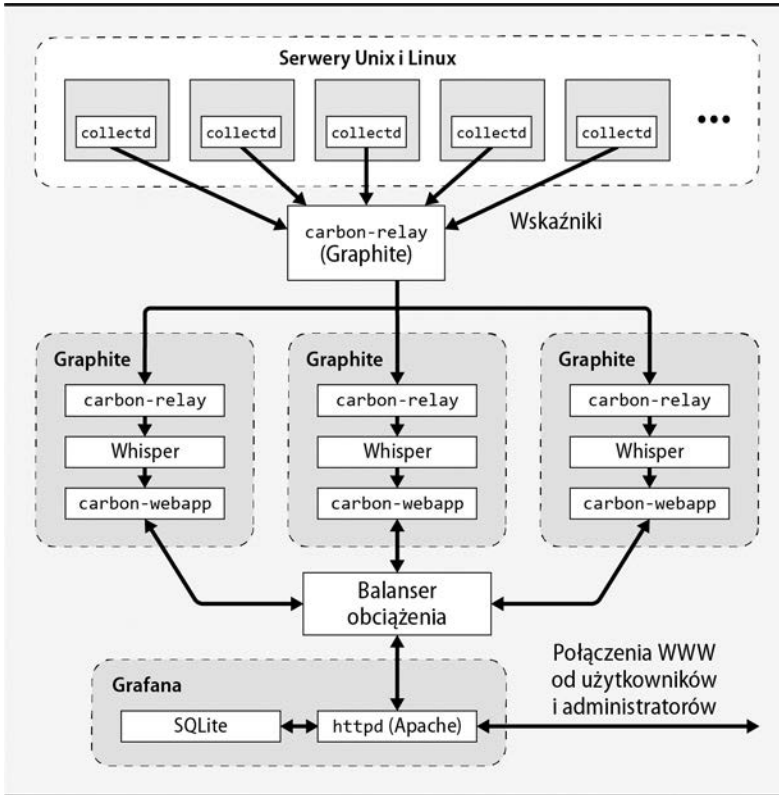
W tych systemach obliczem i sercem ekosystemu monitorowania jest baza danych. Różnią się one pod względem stopnia kompletności jako samodzielne systemy monitoringu i na ogół są zaprojektowane pod kątem świata bardziej modułowego niż ten, dla którego opracowano tradycyjne systemy, takie jak Icinga. Do zbudowania kompletnej platformy monitorowania konieczne może być dostarczenie dodatkowych komponentów.

## Graphite

Graphite był niewątpliwie awangardą platform monitorowania szeregów czasowych nowej generacji. Jego trzonem jest elastyczna baza danych szeregów czasowych z łatwym w użyciu językiem zapytań. Przyczyną popularności hashtagu #monitoringlove i ogromnego wpływu, jaki Graphite wywarł na metody prezentacji danych w interfejsach użytkownika, jest sposób, w jaki agreguje on i podsumowuje wskaźniki. Rozpoczął on odchodzenie od monitoringu w przedziałach minutowych na rzecz skali mierzonej w ułamkach sekundy.

Jak można odgadnąć z nazwy, Graphite zawiera funkcje graficzne do wizualizacji WWW. Ten aspekt pakietu został jednak nieco przyćmiony przez podobną cechę Grafany. Graphite jest dziś lepiej znany z innych komponentów, takich jak Carbon i Whisper, które stanowią rdzeń systemu zarządzania danymi.

Platformę Graphite można łączyć z innymi narzędziami w celu stworzenia skalowalnego, rozproszonego, klastrowego środowiska monitorującego, które jest w stanie pochłaniać i raportować setki tysięcy wskaźników. Taki schemat architektoniczny został przedstawiony na rysunku 28.1.



Rysunek 28.1. Klastrowa architektura platformy Graphite

## Prometheus

Naszą ulubioną platformą szeregów czasowych jest obecnie Prometheus. Jest to wszechstronna platforma, która zawiera zintegrowane ze sobą elementy zbierania danych, analizy trendów i ostrzegania. Komponenty te są przyjazne zarówno dla administratorów, jak i deweloperów, co sprawia, że jest to doskonałe rozwiązanie dla DevOps. Nie pozwala jednak na klastrowanie, co może oznaczać, że nie jest ono odpowiednie dla ośrodków wymagających wysokiej dostępności.

## InfluxDB

InfluxDB jest niezwykle przyjazną dla programistów platformą monitorowania szeregów czasowych, która obsługuje wiele języków programowania. Podobnie jak Graphite, InfluxDB to tak naprawdę tylko silnik bazodanowy szeregów czasowych. Będziesz musiał uzupełnić pakiet zewnętrznymi komponentami, takimi jak Grafana, aby stworzyć kompletny system monitoringu zawierający takie funkcje jak ostrzeganie.

Funkcje zarządzania danymi w InfluxDB są znacznie bardziej rozbudowane niż w przypadku rozwiązań wymienionych wyżej. Dodatkowe funkcje InfluxDB wprowadzają jednak pewną złożoność niepożądaną w typowych instalacjach.

InfluxDB ma dość kłopotliwą historię błędów i niezgodności. Jednak obecna wersja wydaje się stabilna i jest prawdopodobnie najlepszą alternatywą dla platformy Graphite, jeśli szukasz samodzielnego systemu zarządzania danymi.

## Munin

Munin był niegdyś dość popularny, szczególnie w Skandynawii. Jest on zbudowany na bazie sprytniej architektury, w której wtyczki do zbierania danych nie tylko dostarczają dane, ale także informują system, w jaki sposób należy je przedstawić. Mimo że Munin wciąż doskonale nadaje się do użytku, w przypadku nowych wdrożeń należy rozważyć bardziej nowoczesne rozwiązania, takie jak Prometheus. W pewnych sytuacjach Munin jest jednak nadal użytecznym narzędziem do monitorowania aplikacji; patrz podrozdział 28.7.

## Platformy open source do tworzenia wykresów

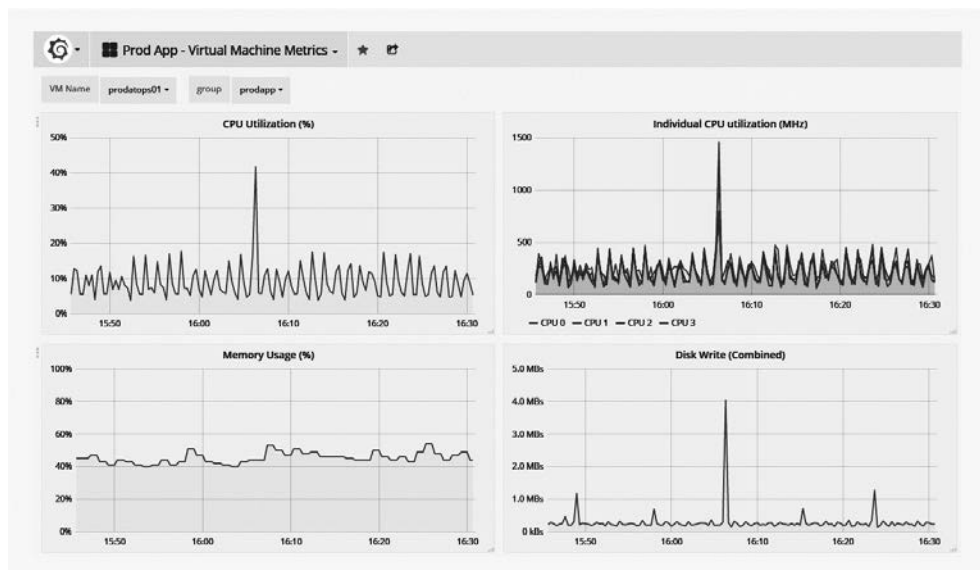
Dwa podstawowe sposoby tworzenia paneli i wykresów to funkcje graficzne wbudowane w platformę Graphite i nowszy pakiet, Grafana.

Graphite może przedstawiać graficznie dane pochodzące z innych magazynów niż Whisper (wbudowany komponent pamięci masowej pakietu Graphite), ale niekoniecznie jest to właściwy wybór.

Jako pakiet agnostyczny bazodanowo Grafana dość dobrze radzi sobie z obsługą obcych magazynów danych, w tym wszystkich wymienionych w poprzednim punkcie. Obsługuje ponad 30 baz źródeł. Pierwotnie Grafana miała być próbą poprawienia wykresów generowanych przez platformę Graphite, więc można z niej wygodnie korzystać również w tym środowisku.

Zarówno Grafite, jak i Grafana prezentują graficzny interfejs przypominający tablicę rozdzielczą, który może generować wizualizacje umożliwiające lepszy wgląd i łatwiejsze zarządzanie. Z ich pomocą można wyświetlać wszystko, począwszy od niskopoziomowych wskaźników systemowych, a skończywszy na wskaźnikach biznesowych. Testy porównawcze zazwyczaj wykazują przewagę Grafany ze względu na jej nieprzeciętny interfejs użytkownika i piękniejsze wykresy.

Na rysunku 28.2 przedstawiony został prosty pulpit Grafany.



Rysunek 28.2. Przykładowy pulpit Grafany

## Komercyjne platformy monitorujące

Setki firm sprzedają oprogramowanie monitorujące, a co tydzień pojawiają się na rynku nowi konkurenci. Jeśli szukasz rozwiązania komercyjnego, powinieneś przynajmniej rozważyć opcje wymienione w tabeli 28.1.

**Tabela 28.1.** Popularne komercyjne platformy monitorujące

Platforma	Adres URL	Komentarz
Datadog	<i>datadoghq.com</i>	platforma monitorowania aplikacji w chmurze długa lista obsługiwanych systemów, aplikacji i usług
Librato	<i>librato.com</i>	obsługa istniejących wtyczek open source w trybie Plug and Play
Monitus	<i>monitus.net</i>	platforma monitorująca dla handlu elektronicznego
Pingdom	<i>pingdom.com</i>	platforma monitorująca oparta na SaaS <sup>a</sup>
SignalFx	<i>signalfx.com</i>	platforma SaaS z integracją z wieloma chmurami
SolarWinds	<i>solarwinds.com</i>	solidny monitoring sieciowy
Sysdig Cloud	<i>sysdig.com</i>	specjalność: monitorowanie i ostrzeganie w Dockerze łatwa korelacja zdarzeń pomiędzy usługami
Zenoss	<i>zenoss.com</i>	niezwykle złożona alternatywa dla systemu Icinga

<sup>a</sup> Nie wymaga instalacji oprogramowania. Dobre rozwiązanie tylko dla aplikacji WWW.

Większość firm nie powinna budować własnego systemu monitorowania, niezależnie od tego, czy miałby się on znajdować w chmurze, w hipernadzorcy centrum danych, czy w szafie. Outsourcing jest tańszy i bardziej niezawodny. Dlatego warto rozważyć Datadog, Librato, SignalFx lub Sysdig Cloud, jeśli potrzebujesz systemu monitorowania dla popularnego zestawu aplikacji bądź serwerów.

Analizując komercyjną platformę monitorującą, często najpierw zwracasz uwagę na jej cenę. Pamiętaj jednak o tym, aby zbadać również szczegóły jej działania:

- Czy da się ją łatwo zintegrować z Twoim systemem zarządzania konfiguracją?
- W jaki sposób system wdraża nowe wtyczki lub testy do Twoich hostów? Czy są one przesyłane, czy pobierane?
- Czy dobrze integruje się z istniejącą platformą powiadamiania, jeśli taką posiadasz?
- Czy Twoje środowisko dopuszcza połączenia zewnętrzne ułatwiające wprowadzenie rozwiązań monitorujących opartych na chmurze?

To tylko kilka z pytań, które powinieneś sobie zadać, analizując platformy. Ostatecznie najlepsza jest taka platforma, która ma prostą konfigurację, jest atrakcyjna cenowo i będzie z łatwością przyjęta przez Twoich użytkowników.

## Hostowane platformy monitorujące

Jeśli nie jesteś zainteresowany konfigurowaniem i utrzymywaniem własnych narzędzi do monitorowania sieci, możesz rozważyć wybór platformy hostowanej (w chmurze). Istnieje wiele darmowych i komercyjnych rozwiązań, ale jednym z najpopularniejszych jest StatusCake (*statuscake.com*). Zewnętrzny dostawca ma ograniczone możliwości przeglądania wewnętrznych szczegółów Twojej sieci, ale rozwiązania hostowane dobrze się sprawdzają w zakresie walidacji stanu usług publicznych i witryn internetowych.

Dostawca hostowanej platformy monitorującej może również uwolnić Cię od ograniczeń związanych z funkcjonowaniem łącza internetowego Twojej organizacji. Jeśli do przesyłania powiadomień z wewnętrznego systemu monitorowania wykorzystujesz swoją sieć nadrzędną — co w końcu robi większość ośrodków — chciałbyś mieć pewność, że sama sieć nadrzędna będzie monitorowana i oprzyrządowana, aby w razie wystąpienia problemów można było zorganizować pracowników.

## 28.4. ZBIERANIE DANYCH

W poprzednim podrozdziale przyjrzelśmy się różnym pakietom, które mogą pełnić funkcję centralnego mechanizmu monitorowania. Wybór i wdrożenie jednego z tych systemów jest jednak tylko pierwszą częścią procesu konfiguracji. Musisz się jeszcze upewnić, że interesujące Cię dane i zdarzenia trafiają do centralnej platformy monitorującej.

Szczegóły procesu oprzyrządowania zależą od systemów, które chcesz monitorować, oraz od filozofii Twojej platformy monitorującej. W wielu przypadkach będziesz musiał napisać kilka prostych skryptów scalających, aby przekonwertować informacje o stanie na formę zrozumiałą dla platformy monitorującej. Niektóre platformy, takie jak Icinga, wyposażone są w zestaw wtyczek, które zbierają standardowe wskaźniki pomiarowe z powszechnie monitorowanych systemów. Inne platformy, takie jak Graphite i InfluxDB, w ogóle nie zapewniają mechanizmu wprowadzania danych i trzeba je uzupełnić o frontend, który wykona to zadanie.

W następnych punktach najpierw omówimy StatsD, frontend zbierania danych ogólnego przeznaczenia, a później przyjrzymy się pewnym narzędziom i technikom oprzyrządowania niektórych monitorowanych systemów.

### StatsD — ogólny protokół przesyłania danych

StatsD został napisany przez inżynierów z serwisu Etsy jako sposób na śledzenie wszystkiego w ich własnym środowisku. Jest to oparty na UDP frontend pośredniczący, który zrzuca wszelkie przechodzące przez niego dane do platformy monitorującej w celu ich wykorzystania, przeliczenia i wyświetlenia. Supermocą StatsD jest jego zdolność do przyjmowania i dokonywania obliczeń na podstawie arbitralnych statystyk.

Demon StatsD firmy Etsy został napisany w Node.js. Jednak obecnie nazwa „StatsD” odnosi się bardziej do protokołu niż do jednego z wielu pakietów oprogramowania, które go implementują (prawdę mówiąc, nawet wersja z Etsy nie jest oryginalna; inspiracją do jej utworzenia był podobnie nazwany projekt firmy Flickr). Powstały wdrożenia napisane w wielu różnych językach, ale tutaj skupimy się na wydaniu Etsy.

StatsD zależy od Node.js, więc przed przystąpieniem do instalacji StatsD upewnij się, że ten pakiet został zainstalowany i odpowiednio skonfigurowany. Implementacja Etsy nie wchodzi w skład repozytoriów pakietów większości systemów operacyjnych, choć inne wersje StatsD są często dostępne; upewnij się, że ich nie pomylisz. Najprościej będzie sklonować wersję Etsy bezpośrednio z repozytorium GitHub:

```
$ git clone https://github.com/etsy/statsd
```

StatsD jest niewiarygodnie modułowy i może dostarczać przychodzące dane do różnych zapleczy i klientów. Spójrzmy na prosty przykład, który wykorzystuje platformę Graphite w roli zaplecza.

Aby się upewnić, że Graphite i StatsD komunikują się poprawnie, musisz zmodyfikować Carbon, składnik magazynujący Graphite. Dokonaj edycji pliku `/etc/carbon/storage-schemas.conf` i dodaj fragment podobny do poniższego:

```
[stats]
pattern = ^stats.*
retentions = 10s:12h,1min:7d,10min:1y
```

Ta konfiguracja nakazuje, aby Carbon przez 12 godzin przechowywał dane z 10-sekundowych odstępów czasu. Carbon podsumowuje wygasające dane w odstępach jednodominutowych i przechowuje to podsumowanie przez kolejne 7 dni. Dane o szczególności 10-minutowej przechowywane są przez cały rok. W dokonywaniu tych wyborów nie ma żadnej magii; musisz określić, co odpowiada potrzebom w zakresie przechowywania danych w Twojej organizacji i jakie dane mają być zbierane.

Dokładna definicja podsumowania danych szeregów czasowych różni się w zależności od rodzaju danych. Przykładowo, jeśli zliczasz błędy sieciowe, prawdopodobnie chciałbyś, aby podsumowanie dodawało wartości do siebie. Jeżeli szukasz mierników, które przedstawiają obciążenie lub stopień wykorzystania danych, prawdopodobnie będziesz potrzebował średniej. Być może będziesz musiał również określić odpowiednie sposoby postępowania z brakującymi danymi.

Zasady te są określone w pliku `/etc/carbon/storage-aggregation.conf`. Jeśli nie masz jeszcze działającej instalacji Graphite, jego przykładowa konfiguracja będzie dla Ciebie dobrym punktem wyjścia:

```
/usr/share/doc/graphite-carbon/examples/storage-aggregation.conf.example
```

Poniżej przedstawiamy kilka sensownych ustawień domyślnych, które warto uwzględnić w pliku `storage-aggregation.conf`.

```
[min]
pattern = \.lower$
xFilesFactor = 0.1
aggregationMethod = min
[max]
pattern = \.upper(_\d+)?$
xFilesFactor = 0.1
aggregationMethod = max
[sum]
pattern = \.sum$
xFilesFactor = 0
aggregationMethod = sum
[count]
pattern = \.count$
xFilesFactor = 0
aggregationMethod = sum
[count_legacy]
pattern = ^stats_counts.*
xFilesFactor = 0
aggregationMethod = sum
[default_average]
pattern = .*
xFilesFactor = 0.3
aggregationMethod = average
```

Zauważ, że każdy blok konfiguracyjny posiada wzorzec wyrażenia regularnego, który próbuje dopasować nazwy serii danych. Bloki są odczytywane po kolei, a pierwszy dopasowany blok staje się specyfikacją kontrolną dla każdej serii danych. Przykładowo seria o nazwie *sample.count* odpowiada wzorcowi w bloku [count]. Wartości zostaną skumulowane poprzez zsumowanie punktów danych (`aggregationMethod = sum`).

Ustawienie `xFilesFactor` określa minimalną liczbę próbek potrzebnych do znaczącego zmniejszenia współczynnika próbkowania wskaźników. Jest ona wyrażona jako liczba z przedziału od 0 do 1 przedstawiająca wartość procentową wartości niezerowych, które muszą występować w warstwie o większej szczegółowości, aby warstwa sumowania nie miała wartości zerowej. Przykładowo ustawienie `xFilesFactor` w przedstawionych wyżej blokach [min] i [max] wynosi 10%, więc nawet pojedyncza wartość danych spełni to kryterium, biorąc pod uwagę nasze ustawienia w pliku *storage-schema.conf*. Wartość domyślna to 50%. Jeśli te ustawienia nie będą dobrze przemyślane, otrzymane dane będą niedokładne lub staną pominięte!

Możemy wysłać kilka danych testowych do StatsD za pomocą programu *netcat* (`nc`):

```
$ echo "sample.count:1|c" | nc -u -w0 statsd.admin.com 8125
```

Polecenie to przekazuje do zbioru danych *sample.count* wartość 1 jako licznik wskaźników (wskazany przez `c`). Pakiet przechodzi do portu 8125. w domenie *statsd.admin.com*; jest to port, na którym domyślnie nasłuchuje *statsd*. Jeśli ten pomiar pojawi się na panelu Graphite, będziesz gotowy do zbierania wszelkiego rodzaju monitorowanych danych za pośrednictwem jednego z wielu klientów StatsD. Listę klientów, które mogą się komunikować z programem StatsD, można znaleźć na jego stronie wiki w repozytorium GitHub ([github.com/etsy/statsd/wiki](https://github.com/etsy/statsd/wiki)). Możesz też napisać własnego klienta! Protokół jest prosty, a możliwości są nieograniczone.

## Pozyskiwanie danych z wyjścia poleceń

Jeśli możesz coś prześledzić z wiersza poleceń, możesz to również prześledzić na swojej platformie monitorującej. Wszystko, czego potrzebujesz, to kilka wierszy skryptu scalającego do wyodrębnienia interesujących Cię danych, które następnie przekształcasz na format akceptowany przez platformę monitorującą.

Przykładowo polecenie `uptime` pokazuje czas pracy systemu, liczbę zalogowanych użytkowników i średnie obciążenie w ciągu ostatniej minuty, ostatnich 5 i 15 minut.

```
$ uptime
07:11:50 up 22 days, 10:13, 2 users, load average: 1.20, 1.41, 1.88
```

Człowiek jest w stanie błyskawicznie przetworzyć te dane wyjściowe i zobaczyć, że średnie obciążenie wynosi obecnie 1,20. Jeśli chcesz napisać skrypt, aby regularnie sprawdzać tę wartość lub wprowadzać ją do innego procesu monitorowania, możesz użyć poleceń do manipulacji tekstem w celu wyodrębnienia żądanej wartości:

```
$ uptime | perl -anF'[\\s,]+' -e 'print $F[10]'2
1.20
```

Zastosowaliśmy tutaj język Perl do podziału wyjścia w każdym miejscu, gdzie występują spacje i przecinki, oraz do wyświetlenia zawartości dziesiątego pola (jest to średnia jednoczynutowego obciążenia). Voilà!

---

<sup>2</sup> To polecenie jest zależne od ustawień regionalnych, np. w polskiej wersji Linuksa nie zadziała, bo wartości dziesiętne są oddzielane przecinkiem, a nie kropką (1,20 zamiast 1.20) — *przyt. tłum.*



Chociaż w większości dziedzin Perl został przyćmiony przez nowocześniejsze języki, takie jak Python i Ruby, wciąż pozostaje królem szybkich manipulacji tekstem. Prawdopodobnie nie warto się uczyć Perla tylko w tym celu, ale umiejętność formułowania wyrafinowanych przekształceń tekstu jako jednowierszowych poleceń jest bardzo przydatna.

Możemy łatwo rozbudować to jednowierszowe polecenie do krótkiego skryptu, który określa wartość średniego obciążenia i przesyła ją do StatsD:

```
#!/usr/bin/env perl
use Net::Statsd;
use Sys::Hostname;
$Net::Statsd::HOST = 'statsd.admin.com';
$loadavg = (split /\s,/, `uptime`)[10];
Net::Statsd::gauge(hostname . '.loadAverage' => $loadavg);
```

Porównaj ten skrypt z naszym jednowierszowym poleceniem wysyłającym dane testowe do StatsD z poprzedniego punktu i naszym jednowierszowym przetwarzaniem wyjścia `uptime`. W tym przypadku Perl musi uruchomić polecenie `uptime` i przetworzyć dane wyjściowe jako łańcuch, dlatego ta część wygląda nieco inaczej niż jej jednowierszowy odpowiednik (w którym wykorzystaliśmy tryb automatycznego podziału).

Zamiast używać `nc` do obsługi sieciowej transmisji danych do StatsD, używamy prostego skryptu osłonowego dla StatsD, który pobraliśmy z archiwum CPAN<sup>3</sup>. Jest to na ogół zalecane podejście; biblioteki są stabilniejsze niż doraźne rozwiązania i wyjaśniają przeznaczenie kodu.

Wiele poleceń może wygenerować więcej niż jeden format wyjściowy. Sprawdź stronę podręcznika danego polecenia, aby przejrzeć dostępne opcje, zanim spróbujesz przetworzyć jego wyniki. Niektóre formaty są znacznie łatwiejsze w obsłudze niż inne.

Kilka poleceń obsługuje format wyjściowy, który szczególnie ułatwia przetwarzanie. Inne posiadają konfigurowalne systemy wyjściowe; możesz w nich zażądać podania wartości tylko tych pól, które naprawdę są Ci potrzebne. Często polecenia mają opcję, która zapobiega wyświetlaniu opisowych wierszy nagłówka w wyjściu.

## 28.5. MONITOROWANIE SIECI

Wiele ośrodków zaczynało od wykorzystywania systemów monitorujących i paneli do monitorowania stanu sieci, więc jest to pierwszy z kilku rodzajów monitoringu, któremu przyjrzymy się nieco dokładniej. W kolejnych punktach zajmiemy się również monitoringiem systemów operacyjnych, aplikacji i usług oraz bezpieczeństwa.

Podstawową jednostką monitoringu sieciowego jest ping sieci, znany również jako pakiet *Echo Request* (żądanie echa) protokołu ICMP. Szczegóły techniczne omawiamy dokładniej w podrozdziale 13.12; opisujemy tam także polecenia `ping` i `ping6`, które inicjują wysłanie sygnału `ping` z wiersza poleceń.

Koncepcja jest prosta: wysyłasz pakiet żądania echa do innego hosta w sieci, a jego implementacja IP w odpowiedzi zwraca pakiet do Ciebie. Jeśli otrzymasz odpowiedź na swój sygnał, wiesz, że wszystkie bramy sieciowe i urządzenia znajdujące się pomiędzy Tobą a docelowym hostem są sprawne. Wiesz również, że host docelowy jest włączony, a jego jądro jest uruchomione i działa. Ponieważ jednak sygnały `ping` obsługiwane są w stosie protokołu TCP/IP, nie dostarczają żadnych informacji na temat stanu oprogramowania wyższego poziomu, które może być uruchomione na docelowym komputerze.

<sup>3</sup> *Comprehensive Perl Archive Network*, [cpan.org](http://cpan.org).

Sygnaly ping nie narzucają zbyt dużego obciążenia sieci, więc można je wysyłać często — powiedzmy, że co dziesięć sekund. Zaprojektuj swoją strategię pingowania w sposób przemyślany, tak aby obejmowała wszystkie ważne bramy i sieci. Pamiętaj, że jeśli sygnał ping nie może przejść przez bramę, nie przejdą przez nią również dane monitoringu, które informują o awarii pingowania. Będziesz potrzebował co najmniej jednego zestawu pingów wychodzących z samego centralnego hosta monitorującego.

Bramy sieciowe nie muszą odpowiadać na pakiety ping, więc ich sygnał może zostać porzucony przez zajętą bramę. Nawet prawidłowo funkcjonująca sieć od czasu do czasu gubi pakiet, a więc nie uruchamiaj od razu alarmu przy pierwszej oznace błędów. Rozsądniejsze będzie zbieranie danych o sygnałach ping jako binarnych rekordów zdarzeń (przeszedł/nie przeszedł) i wprowadzanie ich do zagregowanych miar procentowych strat pakietów w dłuższym okresie.

Warto również dokonywać pomiarów przepustowości pomiędzy dwoma punktami w sieci. Można to zrobić za pomocą iPerf; szczegółowe informacje na ten temat można znaleźć w podrozdziale 13.13.

Większość urządzeń sieciowych obsługuje protokół SNMP(ang. *Simple Network Management Protocol*), będący branżowym standardem, jeśli chodzi o sposób nazywania i zbierania danych operacyjnych. Chociaż SNMP oddalił się znacznie od swoich sieciowych korzeni, uważamy, że jest on zbyt przestarzały, aby służyć do celów innych niż podstawowy monitoring sieci.

SNMP to dość obszerny temat, więc jego omówienie odłożymy na później (do podrozdziału 28.9, „Protokół SNMP”).

## 28.6. MONITOROWANIE SYSTEMÓW

Ponieważ jądro systemu steruje jego procesorem, pamięcią, systemami wejścia-wyjścia i urządzeniami, większość interesujących informacji o stanie systemu, które mógłbyś chcieć monitorować, znajduje się gdzieś wewnątrz jądra. Niezależnie od tego, czy analizujesz dany system ręcznie, czy też utworzyłeś zautomatyzowaną platformę monitorującą, potrzebujesz odpowiednich narzędzi do pozyskiwania i ujawniania informacji o stanie systemu. Większość jąder definiuje formalne kanały, którymi takie informacje są eksportowane.

Niestety, jądra są jak inne rodzaje oprogramowania; wyszukiwanie i usuwanie błędów oraz oprzyrządowanie to często coś, o czym myśli się po fakcie. Chociaż ostatnie lata przyniosły poprawę w zakresie przejrzystości, to zidetyfikowanie i zrozumienie dokładnego parametru, który chciałbyś monitorować, może być trudne, a czasami nawet niemożliwe.

Konkretną wartość można często uzyskać na kilka sposobów. Przykładowo w przypadku średnich obciążeń możesz odczytać wartości bezpośrednio z pliku `/proc/loadavg` w systemach linuksowych lub za pomocą polecenia `sysctl -n vm.loadavg` w systemie FreeBSD. Średnie obciążenia są również zawarte w wyjściach poleceń `uptime`, `w`, `sar` i `top` (choć `top` jest kiepskim wyborem do użytku nieinteraktywnego). Ogólnie rzecz biorąc, najłatwiejsze i najsukuczniejsze jest uzyskanie dostępu do wartości bezpośrednio z jądra (poprzez `sysctl` lub `/proc`), jeśli jest to możliwe.

➡ *Więcej informacji na temat systemu plików `/proc` można znaleźć w podrozdziale 11.4.*

Platformy monitorujące, takie jak Nagios i Icinga, zawierają bogaty zestaw wtyczek monitorujących rozwijanych przez społeczność, które można wykorzystać, aby uzyskać dostęp do powszechnie monitorowanych elementów. Często są to również zwykłe skrypty, które uruchamiają polecenia i przetwarzają dane wyjściowe, ale są już przetestowane i oczyszczone z błędów i działają na wielu platformach. Jeśli nie możesz znaleźć wtyczki, która zapewni Ci interesującą Cię wartość, możesz napisać własną.

## Polecenia dla systemów monitorowania

Tabela 28.2 zawiera listę poleceń, które są powszechnie używane w monitoringu. Wiele z nich daje bardzo różne wyniki w zależności od podanych opcji wiersza poleceń, więc sprawdź szczególnie na stronach podręcznika systemowego.

**Tabela 28.2.** Polecenia zwracające powszechnie monitorowane parametry

Polecenie	Dostępne informacje
df	wolne i zajęte miejsce na dysku oraz i-węzłach
du	rozmiary katalogów
free	ilość pamięci wolnej, zajętej i wirtualnej (wymiany)
iosstat	wydajność i przepustowość dysku
lsof	otwarte pliki i porty sieciowe
mpstat	wykorzystanie poszczególnych procesorów w systemach wieloprocessorowych
vmstat	statystyki dotyczące procesów, procesorów i pamięci

Polecenie `sar` (skrót od *system activity report*, raport o aktywności systemu) jest swobodnym szwajcarskim, jeśli chodzi o ekstrakcję danych z wiersza poleceń. Polecenie to ma skomplikowaną historię; pierwotnie zostało wprowadzone do uniksowego System V w latach osiemdziesiątych ubiegłego wieku<sup>4</sup>. Jego główną zaletą jest to, że zostało ono zaimplementowane na wielu różnych systemach, co zwiększa przenośność zarówno dla skryptów, jak i dla administratorów systemów. Niestety, port BSD nie jest już utrzymywany.

Poniższy przykład żąda raportowania co dwie sekundy przez okres jednej minuty (tj. 30 raportów). Argument `DEV` jest słowem kluczowym, a nie zmienną dla nazwy urządzenia lub interfejsu.

```
$ sar -n DEV 2 30
17:50:43 IFACE rxpck/s txpck/s rxbyt/s txbyt/s rxcmp/s txcmp/s rxmct/s
17:50:45 lo 3.61 3.61 263.40 263.40 0.00 0.00 0.00
17:50:45 eth0 18.56 11.86 1364.43 1494.33 0.00 0.00 0.52
17:50:45 eth1 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

Przykład ten pochodzi z komputera z systemem Linux z dwoma interfejsami sieciowymi. Wyjście zawiera zarówno chwilowe, jak i średnie odczyty wykorzystania interfejsu, przy czym każdy odczyt wyrażony jest zarówno w bajtach, jak i w pakietach. Drugi interfejs (`eth1`) najwyraźniej nie jest używany.

<sup>4</sup> Administratorów starej daty można rozpoznać po ich biegłości w posługiwaniu się poleceniem `sar`.

## collectd — pozyskiwanie ogólnych danych systemowych

W miarę jak administrowanie systemami ewoluowało od zmagania się z pojedynczymi systemami do zarządzania flotą wirtualnych instancji, użytkowanie prostych narzędzi wiersza poleceń zaczęło stwarzać wiele problemów w dziedzinie monitorowania. Choć pisanie skryptów do zbierania i analizowania parametrów jest utylitarnym i elastycznym podejściem, utrzymanie spójności tej bazy kodów w wielu systemach szybko staje się kłopotliwe. Nowoczesne narzędzia, takie jak collectd, sysdig i dtrace, oferują bardziej skalowalne podejście do zbierania tego typu danych.

Gromadzenie statystyk systemowych powinno być procesem ciągłym, a uniksowe rozwiązanie dla trwającego zadania polega na utworzeniu demona, który się tym zajmie. Jest nim collectd, demon zbierania statystyk systemowych.

To popularne i dojrzałe narzędzie dla systemów Linux i FreeBSD. Zazwyczaj collectd działa w systemie lokalnym, zbiera dane pomiarowe w określonych odstępach czasu i zapisuje wartości wynikowe. Możesz również skonfigurować collectd do pracy w trybie klient-serwer, w którym collectd (lub kilka jego instancji) agreguje dane z grupy innych serwerów.

Specyfikacja parametrów, które mają być zbierane, oraz miejsc docelowych, do których są one zapisywane, jest elastyczna; dostępnych jest ponad 100 wtyczek, które można dostosować do własnych potrzeb. Po uruchomieniu collectd może być odpytywany przez platformę monitorującą (taką jak Icinga lub Nagios) w celu monitorowania na bieżąco albo może przekazywać dane do takich platform jak Graphite bądź InfluxDB w celu analizy szeregów czasowych.

Oto przykładowy plik konfiguracyjny collectd:

```
## /etc/collectd/collectd.conf
Hostname client1.admin.com
FQDNLookup false
Interval 30
LoadPlugin syslog
<Plugin syslog>
    LogLevel info
</Plugin>
LoadPlugin cpu
LoadPlugin df
LoadPlugin disk
LoadPlugin interface
LoadPlugin load
LoadPlugin memory
LoadPlugin processes
LoadPlugin rrdtool
<Plugin rrdtool>
    DataDir "/var/lib/collectd/rrd"
</Plugin>
```

Ta podstawowa konfiguracja zbiera wiele interesujących statystyk systemowych co 30 sekund i zapisuje pliki danych zgodnie z RRD (ang. *Round Robin Database*, cykliczna baza danych) w katalogu `/var/lib/collectd/rrd`.

## sysdig i dtrace — śledzenie działań w systemie

sysdig (Linux) i dtrace (BSD) kompleksowo instrumentują aktywność zarówno jądra, jak i procesów użytkownika. Obejmują one komponenty, które są wstawiane do samego jądra, ujawniając nie tylko głęboko ukryte parametry jądra, ale także wywołania systemowe i inne statystyki wydajności. Narzędzia te są czasami określane jako „Wireshark dla jądra i procesów”.

➤ *Więcej informacji na temat Wiresharka można znaleźć w podrozdziale 13.12.*

Oba te narzędzia są skomplikowane. Warto jednak podjąć wyzwanie i zapoznać się z nimi. Weekend spędzony na nauce jednego z nich da Ci niesamowite nowe supermoce i zapewni Ci status gościa z listy VIP-ów na spotkaniu koktajlowym dla administratorów systemów.

sysdig potrafi obsługiwać kontenery, dzięki czemu zapewnia niezwykłą widoczność w środowiskach, w których używane są narzędzia takie jak Docker i LXC. sysdig jest dystrybuowany jako oprogramowanie open source i można go zintegrować z innymi narzędziami monitoringu, takimi jak Nagios lub Icinga. Deweloperzy oferują również komercyjną usługę monitoringu (Sysdig Cloud), która posiada pełne możliwości monitorowania i ostrzegania.

➤ *Więcej informacji na temat Dockera i kontenerów można znaleźć w rozdziale 25.*

## 28.7. MONITOROWANIE APLIKACJI

Na szczycie piramidy oprogramowania znajduje się Święty Graal: monitoring aplikacji. Ten rodzaj monitorowania jest dość niejasno zdefiniowany, ale ogólna koncepcja polega na tym, że próbuje się sprawdzać status i wydajność poszczególnych elementów oprogramowania, a nie systemów czy sieci jako całości. W wielu przypadkach monitorowanie aplikacji może sięgać do tych systemów i profilować ich wewnętrzne działania.

Aby mieć pewność, że monitorujesz właściwe rzeczy, musisz włączyć do pracy jednostki biznesowe i programistów, by opowiedzieli Ci więcej o swoich zainteresowaniach i obawach. Jeśli masz witrynę internetową działającą np. na zestawie LAMP, prawdopodobnie będziesz chciał monitorować czasy ładowania stron, oznaczać krytyczne błędy PHP, trzymać zakładki w bazie danych MySQL i monitorować konkretne problemy, takie jak nadmierne próby połączeń.

Monitorowanie tej warstwy może być złożone, ale jest to również obszar, w którym monitorowanie staje się atrakcyjne. Wyobraź sobie, że możesz monitorować (i prezentować na swoim pięknym panelu Grafany) liczbę widżetów, które sprzedałeś w ciągu minionej godziny, lub średni czas, przez jaki dany przedmiot pozostaje w koszyku. Jeśli pokażesz twórcom aplikacji i właścicielom procesów taki poziom funkcjonalności, zazwyczaj otrzymasz natychmiastowe wsparcie w celu dodania dodatkowego monitorowania, a może nawet pomoc w jego wdrożeniu. W końcu ta warstwa monitoringu stanie się nieoceniona dla biznesu, a Ty zaczniesz być postrzegany jako mistrz monitoringu, wskaźników i analizy danych.

Monitorowanie na poziomie aplikacji pozwala uzyskać dodatkowy wgląd w inne zdarzenia w środowisku. Przykładowo, jeśli sprzedaż widżetów szybko spada, może to być oznaką tego, że jedna z sieci reklamowych nie działa.

### Monitorowanie dzienników

W swojej najbardziej podstawowej formie monitorowanie dzienników polega na przeszukiwaniu plików dzienników w celu zdobycia interesujących danych, które chciałbyś monitorować, wyciąganiu tych danych i przetwarzaniu ich do postaci zdanej do analizy, wyświetlania i ostrzegania. Ponieważ komunikaty dziennika składają się z tekstu o dowolnej postaci, implementacja tego potoku może mieć bardzo różny poziom złożoności; może to być zarówno trywialne, jak i mocno skomplikowane.

Zazwyczaj dziennikami najlepiej zarządza się za pomocą kompleksowego systemu agregacji zaprojektowanego specjalnie w tym celu. Omawiamy takie systemy w podrozdziale 10.4,

„Zarządzanie dziennikami na dużą skalę”. Choć systemy te koncentrują się przede wszystkim na centralizacji danych dzienników oraz ułatwianiu ich wyszukiwania i przeglądania, większość systemów agregacji obsługuje również wartości progowe, alarmy i funkcje raportowania.

Jeśli potrzebujesz automatyzacji przeglądania dziennika dla kilku konkretnych celów i nie masz ochoty inwestować w bardziej ogólny system zarządzania dziennikami, zalecamy skorzystanie z pary narzędzi działających w mniejszej skali: `logwatch` i `OSSEC`.

`logwatch` jest elastycznym analizatorem dzienników, zorientowanym na przetwarzanie wsadowe. Jego podstawowym zastosowaniem jest tworzenie dziennych podsumowań zdarzeń raportowanych w dziennikach. Możesz uruchamiać `logwatch` częściej niż raz dziennie, ale nie został on zaprojektowany do monitorowania w czasie rzeczywistym. Do tego warto wykorzystać `OSSEC`, który omawiamy w podrozdziale 27.5. `OSSEC` jest promowany jako narzędzie bezpieczeństwa, ale jego architektura jest na tyle ogólna, że jest również przydatny dla innych rodzajów monitoringu.

## Supervisor + Munin — proste rozwiązanie dla ograniczonych zastosowań

Wszechstronna, kompleksowa platforma, taka jak `Icinga` czy `Prometheus`, może się okazać nieodpowiednia dla Twoich potrzeb lub Twojego środowiska. Co zrobić, jeśli jesteś zainteresowany tylko monitorowaniem jednego konkretnego procesu aplikacji i nie chcesz zwracać sobie głowy pełnoprawną platformą monitorującą? Rozważ połączenie ze sobą narzędzi `Munin` i `Supervisor`. Są one łatwe w instalacji, wymagają niewiele czynności konfiguracyjnych i dobrze ze sobą współpracują.

`Supervisor` i jego proces serwera `supervisord` pomagają monitorować procesy i generować zdarzenia lub powiadomienia, gdy procesy kończą działanie albo zgłaszają wyjątek. System ten przypomina w swojej istocie `Upstart` lub te części `systemd`, które odpowiadają za zarządzanie procesami.

Jak już wspominaliśmy, `Munin` jest platformą monitorującą ogólnego przeznaczenia, ze szczególnym uwzględnieniem monitorowania aplikacji. Jest napisany w języku `Perl` i wymaga, aby na każdym systemie, który chcesz monitorować, uruchomiony był agent zwany węzłem (ang. *Munin Node*). Konfiguracja nowego węzła jest prosta: zainstaluj pakiet `munin-node`, zmodyfikuj plik `munin-node.conf`, aby wskazać komputer nadrzędny, i gotowe.

Przedstawiając zgromadzone dane, `Munin` domyślnie tworzy wykresy za pomocą silnika opartego na `RRDtool`, więc jest to przyjemny i wymagający niewiele czynności konfiguracyjnych sposób na uzyskanie graficznej informacji zwrotnej. `Munin` posiada ponad 300 wtyczek oraz prawie 200 udostępnionych bibliotek. Prawdopodobnie znajdziesz już istniejącą wtyczkę, która zaspokoi Twoje potrzeby. Jeśli nie, łatwo jest napisać nowy skrypt do wykonania przez `munin-node`.

## Komercyjne narzędzia do monitorowania aplikacji

Jeśli wpiszesz w Google hasło „narzędzie do monitorowania aplikacji”, odkryjesz wiele stron przedstawiających komercyjne produkty do oceny. Aby dochować należytej staranności, powinienś również prześledzić ostatnie dyskusje na temat monitorowania wydajności aplikacji (APM, ang. *application performance monitoring*).

Znajdziesz tam wiele odniesień do DevOps, z uzasadnionych powodów: monitorowanie aplikacji i APM to kluczowe elementy DevOps. Dostarczają one danych ilościowych; zespoły mogą ich użyć, aby zdecydować, którym obszarom działania na rzecz poprawy wydajności i stabilności przyniosą największe korzyści.

➔ *Więcej informacji na temat DevOps można znaleźć w podrozdziale 31.1.*

Uważamy, że w tym zakresie wyróżniają się New Relic (*newrelic.com*) oraz AppDynamics (*appdynamics.com*). Możliwości tych systemów pod wieloma względami nakładają się na siebie, ale AppDynamics zazwyczaj koncentruje się na rozwiązaniach monitorujących wykorzystujących „pełny zestaw”, podczas gdy New Relic zajmuje się bardziej profilowaniem zachowania w obrębie samej warstwy aplikacji.

Niezależnie od tego, w jaki sposób monitorujesz swoje aplikacje, ważne jest, aby w ten proces byli zaangażowani programiści. Pomogą oni zapewnić monitorowanie wszystkich ważnych wskaźników. Ścisła współpraca w zakresie monitorowania sprzyja rozwijaniu wzajemnych relacji pomiędzy zespołami i ogranicza powielanie działań.

## 28.8. MONITOROWANIE BEZPIECZEŃSTWA

Monitoring bezpieczeństwa to osobny wszechświat. Ten obszar praktyki operacyjnej nazywany jest czasami operacjami bezpieczeństwa, w skrócie SecOps.

Można wymienić dziesiątki narzędzi i usług, zarówno open source, jak i komercyjnych, które pomagają w monitorowaniu bezpieczeństwa środowiska. Podmioty zewnętrzne, zwane dostawcami zarządzanych usług bezpieczeństwa (MSSP, ang. *managed security service providers*), świadczą usługi na zasadzie outsourcingu<sup>5</sup>. Mimo istnienia wszystkich tych możliwości naruszenia bezpieczeństwa są nadal powszechne i często pozostają niewykryte przez miesiące lub lata.

Prawdopodobnie najważniejszą rzeczą, jaką należy wiedzieć o monitorowaniu bezpieczeństwa, jest to, że nie wystarczy tu zautomatyzowane narzędzie lub zautomatyzowana usługa. Musisz wdrożyć kompleksowy program bezpieczeństwa, zawierający standardy dotyczące zachowań użytkowników, przechowywania danych i procedur reagowania na incydenty, by wymienić tylko kilka elementów. Podstawy tego zagadnienia omawiamy w rozdziale 27., „Bezpieczeństwo”.

Z automatyczną, ciągłą strategią monitorowania należy zintegrować dwie podstawowe funkcje bezpieczeństwa: weryfikację integralności systemu i wykrywanie włamań.

### Weryfikowanie integralności systemu

Weryfikacja integralności systemu, często nazywana monitoringiem integralności plików (FIM, *file integrity monitoring*), jest walidacją bieżącego stanu systemu w stosunku do znanego stanu wyjściowego. Najczęściej walidacja ta porównuje zawartość plików systemowych (jądro, polecenia wykonywalne, pliki konfiguracyjne) z kryptograficzną sumą kontrolną, taką

---

<sup>5</sup> Outsourcing operacji bezpieczeństwa zawsze wydaje się atrakcyjny; zapewnienie bezpieczeństwa Twojemu środowisku staje się wtedy problemem kogoś innego. Ale pomyśl o tym tak: czy czułbyś się komfortowo, płacąc komuś za pilnowanie Twojego wypełnionego gotówką portfela, leżącego wraz z 10 000 innych portfeli na stole na zatłoczonym dworcu kolejowym? Jeśli tak, to MSSP może być dla Ciebie dobrym rozwiązaniem!

jak SHA-512<sup>6</sup>. Jeżeli suma kontrolna pliku w uruchomionym systemie różni się od sumy kontrolnej wersji wyjściowej, administrator zostanie o tym powiadomiony. Oczywiście, należy brać pod uwagę regularne działania konserwacyjne, takie jak planowane zmiany, aktualizacje i poprawki; nie wszystkie zmiany są podejrzane.

Najczęściej stosowanymi platformami FIM są Tripwire i OSSEC, opisane bardziej szczegółowo w podrozdziale 27.5. Linuksowa wersja AIDE również zawiera monitorowanie integralności plików, ale wersja FreeBSD niestety nie ma tego komponentu.

Prostsze jest często lepsze. Wspaniałą standardową opcją FIM jest program `mtree`, który wywodzi się z FreeBSD, a niedawno został przeniesiony do Linuksa. `mtree` to prosty sposób na monitorowanie stanu plików i zmian ich zawartości, łatwy do zintegrowania ze skryptami monitorującymi. Oto przykład szybkiego skryptu wykorzystującego `mtree`:

```
#!/bin/bash
if [ $# -eq 0 ]; then
    echo "mtree-check.sh [-bv]"
    echo "-b = utworzenie stanu wyjściowego"
    echo "-v = weryfikacja względem stanu wyjściowego"
    exit
fi
## ziarno
KEY=93948764681464
## katalog bazowy
DIR=/usr/local/lib/mtree-check
if [ $1 = "-b" ]; then
    rm -rf $DIR/mtree_*
    cd $DIR
    mtree-c -K sha512 -s $KEY -p /sbin > mtree_sbin
fi
if [ $1 = "-v" ]; then
    cd $DIR
    mtree -s $KEY -p /sbin < mtree_sbin | \
        mail -s "mtree: test integralności `hostname`" dan@admin.com
fi
```

Po dodaniu opcji `-b` skrypt tworzy i zapisuje stan wyjściowy. Po ponownym uruchomieniu z opcją `-v` sprawdza bieżącą zawartość katalogu `/sbin` względem stanu wyjściowego.

Podobnie jak w przypadku wielu aspektów zarządzania systemem, stworzenie platformy FIM i jej obsługa w czasie to dwie zupełnie różne sprawy. Do przechowywania danych i odpowiadania na powiadomienia FIM potrzebny jest zdefiniowany proces. Sugerujemy wprowadzanie informacji z platformy FIM do infrastruktury monitorowania i ostrzegania, tak aby nie były one odsuwane na bok lub ignorowane.

## Monitorowanie wykrywania włamań

Powszechnie stosuje się dwa rodzaje systemów wykrywania włamań (IDS, ang. *intrusion detection systems*): hostowe (HIDS, ang. *host-based IDS*) i sieciowe (NIDS, ang. *network-based IDS*). Systemy NIDS badają ruch przechodzący przez sieć i próbują zidentyfikować niespodziewane lub podejrzane wzorce. Najpopularniejszy system NIDS jest oparty na Snort; omawiamy go szczegółowo w podrozdziale 27.5.

---

<sup>6</sup> Akceptowalne algorytmy skrótu zmieniają się w czasie. Na przykład algorytm MD5 nie jest już uważany za kryptograficznie bezpieczny i nie powinno się go używać.



Systemy HIDS działają jako zestaw procesów na każdym systemie. Zazwyczaj śledzą one różne rzeczy, w tym połączenia sieciowe, czasy modyfikacji plików i sumy kontrolne, dzienniki demonów i aplikacji, wykorzystanie podwyższonych uprawnień oraz inne wskazówki, które mogą sygnalizować działanie narzędzi zaprojektowanych w celu ułatwienia nieautoryzowanego dostępu (programów typu „root kit”). HIDS nie jest kompleksowym rozwiązaniem dla bezpieczeństwa, ale stanowi cenny element kompleksowego podejścia.

Dwie najpopularniejsze platformy HIDS typu open source to OSSEC (ang. *Open Source SECurity*) oraz AIDE (ang. *Advanced Intrusion Detection Environment*). Z naszego doświadczenia wynika, że OSSEC jest najlepszym wyborem. Chociaż AIDE jest świetną platformą FIM dla systemu Linux, OSSEC posiada bogatszy zestaw funkcji. Może być nawet używany w trybie klient-serwer, który obsługuje klientów innych niż Unix, takich jak Microsoft Windows, i wiele urządzeń infrastruktury sieciowej.

Podobnie jak alarmy FIM, dane HIDS są użyteczne tylko wtedy, gdy poświęcamy tej platformie należyłą uwagę. HIDS nie jest podsystemem typu „ustaw i zapomnij”; będziesz musiał zintegrować alarmy HIDS z Twoim ogólnym systemem monitorowania. Najskuteczniejszą strategią, jaką udało nam się znaleźć do rozwiązania tego problemu, jest automatyczne otwieranie zgłoszeń na ostrzeżenia HIDS w Twoim systemie zgłoszeniowym. Następnie można dodać kontrolę monitoringu, która ostrzeże o wszystkich nierozwiązanych zgłoszeniach HIDS.

## 28.9. PROTOKÓŁ SNMP

Przed laty branża sieciowa zdecydowała, że warto byłoby stworzyć standardowy protokół gromadzenia danych monitoringu. Tak powstał protokół SNMP (ang. *Simple Network Management Protocol*).

Wbrew swojej nazwie SNMP jest w rzeczywistości dość skomplikowany. Określa hierarchiczną przestrzeń nazw dla zarządzanych danych oraz metody odczytu i zapisu tych danych na każdym urządzeniu sieciowym. SNMP definiuje również sposób wysyłania komunikatów powiadamiających o zdarzeniach („pułapek”) przez zarządzane serwery i urządzenia („agenty”) do stacji zarządzających.

Zanim zagłębimy się w tajniki SNMP, chcielibyśmy podkreślić, że terminologia z nim związana jest jednym z najbardziej beznadziejnych przypadków bełkotu technologicznego, z jakim można się zetknąć na rynku rozwiązań sieciowych. Niejednokrotnie standardowe nazwy pojęć i obiektów związanych z SNMP utrudniają zrozumienie tego, jaka jest ich rzeczywista rola.

Sam protokół jest jednak prosty; większość złożoności SNMP leży ponad warstwą protokołu w konwencjach tworzenia przestrzeni nazw oraz w zbyt wyszukanim słownictwie otaczającym SNMP jak skorupa ochronna. Pomijając jego mechanikę wewnętrzną, SNMP jest łatwy w obsłudze.

SNMP został zaprojektowany w taki sposób, aby mógł być wdrażany przez dedykowany sprzęt sieciowy, taki jak routery; w tym kontekście jest to słuszne rozwiązanie. Później SNMP został rozbudowany o możliwość monitorowania serwerów i systemów desktopowych, ale jego przydatność w tym zakresie zawsze budziła wątpliwości. Obecnie dostępne są znacznie lepsze rozwiązania (np. `collected`; patrz podrozdział 28.6).

Sugerujemy, aby traktować SNMP jako niskopoziomowy protokół zbierania danych do użytku z urządzeniami o specjalnym przeznaczeniu, które nie obsługują niczego innego. Jak najszybciej wyprowadź dane ze świata SNMP i przekaz je platformie monitorującej ogólnego przeznaczenia w celu ich przechowywania i przetwarzania. SNMP może być ciekawą okolicą do zwiedzania, ale nie chciałbyś tam mieszkać!

## Organizacja SNMP

Dane SNMP są uporządkowane w ustandaryzowanej hierarchii. Hierarchia nazw oparta jest na formacie MIB (ang. *Management Information Bases*), czyli ustrukturyzowanych plikach tekstowych opisujących dane dostępne za pośrednictwem SNMP. Pliki MIB zawierają opisy zmiennych określających pewne dane, do których można się odwoływać przy użyciu tzw. identyfikatorów obiektów (OID)<sup>7</sup>. Wszystkie obecnie stosowane urządzenia korzystające z SNMP obsługują strukturę w formacie MIB-II, zdefiniowanym w dokumencie RFC 1213. Producenci mogą jednak rozbudowywać MIB (i robią to), aby dodać więcej danych i wskaźników.

Identyfikatory obiektów istnieją w hierarchicznej przestrzeni nazw, gdzie węzły są numerowane, a nie nazwane. Jednak aby ułatwić odwoływanie się do nich, węzły mają również konwencjonalne nazwy tekstowe. Separatorem ścieżek jest tu kropka. Przykładowo OID odwołujący się do czasu nieprzerwanej pracy urządzenia ma wartość 1.3.6.1.2.1.1.3. Ten OID ma także czytelną dla człowieka (choć niekoniecznie „zrozumiałą bez dodatkowej dokumentacji”) nazwę:

```
iso.org.dod.internet.mgmt.mib-2.system.sysUpTime
```

Tabela 28.3 przedstawia przykłady węzłów OID, które warto monitorować przy ocenie dostępności sieci.

**Tabela 28.3.** Wybrane OID z MIB-II

OID <sup>a</sup>	Typ	Zawartość
system.sysDescr	string	informacje systemowe: dostawca, model, system operacyjny itp.
interfaces.ifNumber	int	liczba obecnych interfejsów sieciowych
interfaces.ifTable	table	tabela charakterystyk każdego z interfejsów
ip.ipForwarding	int	1, jeśli system jest bramą, w przeciwnym wypadku 2
ip.ipAddrTable	table	tabela danych o adresach IP (maski itp.)
icmp.icmpInRedirects	int	liczba otrzymanych przekierowań ICMP
icmp.icmpInEchos	int	liczba otrzymanych sygnałów ping
tcp.tcpInErrs	int	liczba otrzymanych błędów TCP

<sup>a</sup> Ścieżka względna w stosunku do *iso.org.dod.internet.mgmt.mib-2*.

Oprócz powszechnie obsługiwanego formatu MIB-II istnieją też formaty MIB dla różnego rodzaju interfejsów i protokołów sprzętowych, poszczególnych dostawców, różnych implementacji serwerów snmpd, a także dla poszczególnych produktów sprzętowych.

MIB to tylko schemat nazewniczy dla zarządzanych danych. Aby urządzenie MIB mogło być użyteczne, musi być obsługiwane przez kod działający po stronie agenta, który odwzorowuje rzeczywisty stan urządzenia na przestrzeń nazw SNMP.

Agenty SNMP działające w systemach Unix, Linux lub Windows posiadają wbudowaną obsługę MIB-II. Większość z nich można rozbudować o możliwość obsługi dodatkowych MIB oraz interfejsów za pomocą skryptów, które wykonają właściwą pracę polegającą na zbieraniu i przechowywaniu danych związanych z tymi definicjami MIB. Zobaczysz wiele tego rodzaju programów, będących pozostałością minionej epoki, kiedy to SNMP był jeszcze nowością.

<sup>7</sup> OID to po prostu wyszukane określenie na odwołanie do zarządzanej informacji.

Ale to już tylko dużo dymu bez ognia; obecnie nie powinieneś w ogóle uruchamiać agenta SNMP w systemie Unix, chyba że musisz uzyskać odpowiedź na najbardziej podstawowe pytania dotyczące konfiguracji sieci.

## Operacje protokołu SNMP

Istnieją tylko cztery podstawowe operacje SNMP: `get`, `get-next`, `set` i `trap`.

`get` i `set` są głównymi operacjami odczytu i zapisu danych do węzła zidentyfikowanego przez określony OID. Operacja `get-next` służy do poruszania się w hierarchii MIB, ale może również odczytywać zawartość tabel.

Operacja `trap` (pułapka) to samodzielnie wysyłane, asynchroniczne powiadomienie od serwera (agenta) do klienta (zarządcy) o wystąpieniu interesującego zdarzenia lub stanu. Zdefiniowano kilka standardowych pułapek, w tym powiadomienia „właśnie się uruchomiłem”, raporty o awarii lub odzyskaniu połączenia sieciowego oraz komunikaty o różnych problemach z routowaniem i uwierzytelnianiem. Mechanizm, za pomocą którego określa się miejsca docelowe komunikatów `trap`, zależy od implementacji agenta.

Ponieważ komunikaty SNMP mogą potencjalnie modyfikować informacje konfiguracyjne, potrzebny jest jakiś mechanizm bezpieczeństwa. Najprostsza wersja zabezpieczeń SNMP używa pojęcia łańcucha wspólnotowego (ang. *community string*), które jest tak naprawdę strasznie zagmatwanym określeniem hasła. Zazwyczaj jeden łańcuch wspólnotowy umożliwia dostęp w trybie tylko do odczytu, a drugi pozwala na zapisywanie<sup>8</sup>. W dzisiejszych czasach znacznie bardziej sensowne jest utworzenie mechanizmu zarządzania na bazie protokołu SNMPv3, który zapewnia większe bezpieczeństwo, w tym uprawnienia i kontrolę dostępu dla poszczególnych użytkowników.

## Net-SNMP — narzędzia dla serwerów

W systemach Linux i FreeBSD najpopularniejszym pakietem implementującym SNMP jest Net-SNMP. Zawiera on agenta (`snmpd`), kilka narzędzi wiersza poleceń, serwer do odbierania komunikatów `trap`, a nawet bibliotekę do tworzenia aplikacji obsługujących SNMP.

Obecnie Net-SNMP stanowi przedmiot zainteresowania bardziej ze względu na swoje polecenia i biblioteki niż na to, że jest agentem. Został on przeniesiony do wielu różnych systemów uniksowych, dzięki czemu działa jako spójna platforma, na którą można pisać skrypty. Większość dystrybucji wydziela więc z agenta Net-SNMP własny pakiet, co sprawia, że łatwiej jest zainstalować same polecenia.



W systemach Debian i Ubuntu pakiety Net-SNMP noszą nazwy `snmp` i `snmpd`. Aby zainstalować tylko polecenia, wpisz `apt-get install snmp`.




W systemach Red Hat i CentOS analogiczne pakiety to `net-snmp` i `net-snmp-tools`. Zainstaluj polecenia przez `yum install net-snmp-tools`.



W Linuksie informacje konfiguracyjne trafiają do katalogu `/etc/snmp`; zwróć uwagę na znajdujący się tam plik `snmpd.conf` i katalog `snmp.d`. Aby uruchomić demona, wpisz polecenie `systemctl start snmpd`.

<sup>8</sup> W wielu systemach łańcuch wspólnotowy ma domyślne ustawienie „public”. Nigdy tego tak nie zostawiaj; ustaw własne hasła dla obu łańcuchów, zarówno dla tego, który pozwala na odczytywanie, jak i tego, który umożliwia odczyt i zapis.

 W systemie FreeBSD wszystko znajduje się w jednym pakiecie: `pkg install net-snmp`. Informacje konfiguracyjne są wprowadzane do katalogu `/usr/local/etc/snmp`, który musisz samemu utworzyć. Możesz uruchomić agenta ręcznie poleceniem `service snmpd start` lub umieścić wpis:

```
snmpd_enable="YES"
```

w pliku `/etc/rc.conf`, aby był uruchamiany wraz z systemem.

We wszystkich systemach, w których musisz uruchomić agenta SNMP, należy się upewnić, że port UDP 162 nie jest zablokowany przez zaporę sieciową.

Polecenia dostarczane przez Net-SNMP pomogą Ci się zapoznać z SNMP, a także świetnie się nadają do jednorazowych kontroli poszczególnych OID. Tabela 28.4 zawiera listę najczęściej używanych narzędzi.

**Tabela 28.4.** Narzędzia wiersza poleceń w pakiecie Net-SNMP

Polecenie	Działanie
<code>snmpdelta</code>	monitoruje zmiany wartości zmiennych SNMP w czasie
<code>snmpdf</code>	monitoruje przestrzeń dyskową na zdalnym hoście przez SNMP
<code>snmpget</code>	odczytuje wartość zmiennej SNMP podaną przez agenta
<code>snmpgetnext</code>	pobiera kolejną zmienną z sekwencji
<code>snmpset</code>	ustawia wartość zmiennej SNMP na agencie
<code>snmptable</code>	odczytuje tabelę zmiennych SNMP
<code>snmptranslate</code>	wyszukuje i wyświetla informacje o OID w hierarchii MIB
<code>snmptrap</code>	generuje komunikat trap
<code>snmpwalk</code>	przełąca kolejne wartości drzewa MIB, rozpoczynając od wskazanego OID

Podstawowe kontrole SNMP zazwyczaj wykorzystują połączenie `snmpget` i `snmpdelta`. Inne programy są pomocne, gdy chcesz zidentyfikować nowe OID do monitorowania za pomocą swojego wyszukanego narzędzia zarządzającego. Przykładowo `snmpwalk` rozpoczyna od określonego OID (domyślnie jest nim początek drzewa MIB) i wielokrotnie wywołuje na agencie operację `get-next`. Proces ten umożliwia uzyskanie kompletnej listy dostępnych OID oraz przypisanych im wartości.

W poniższym listingu przedstawiamy przycięty przykład działania polecenia `snmpwalk` na hoście *tuva* z systemem Linux. `secret813community` to łańcuch wspólnotowy, a opcja `-v1` wymusza proste uwierzytelnianie.

```
$ snmpwalk -c secret813community -v1 tuva
SNMPv2-MIB::sysDescr.0 = STRING: Linux tuva.atrust.com 2.6.9-11.ELsmp #1
SNMPv2-MIB::sysUpTime.0 = Timeticks: (1442) 0:00:14.42
SNMPv2-MIB::sysName.0 = STRING: tuva.atrust.com
IF-MIB::ifDescr.1 = STRING: lo
IF-MIB::ifDescr.2 = STRING: eth0
IF-MIB::ifDescr.3 = STRING: eth1
IF-MIB::ifType.1 = INTEGER: softwareLoopback(24)
IF-MIB::ifType.2 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifType.3 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifPhysAddress.1 = STRING:
IF-MIB::ifPhysAddress.2 = STRING: 0:11:43:d9:1e:f5
IF-MIB::ifPhysAddress.3 = STRING: 0:11:43:d9:1e:f6
```

```

IF-MIB::ifInOctets.1 = Counter32: 2605613514
IF-MIB::ifInOctets.2 = Counter32: 1543105654
IF-MIB::ifInOctets.3 = Counter32: 46312345
IF-MIB::ifInUcastPkts.1 = Counter32: 389536156
IF-MIB::ifInUcastPkts.2 = Counter32: 892959265
IF-MIB::ifInUcastPkts.3 = Counter32: 7712325
...

```

W tym przykładzie widzimy kilka ogólnych informacji o systemie, po których następują statystyki dotyczące interfejsów sieciowych: *lo0*, *eth0* i *eth1*. W zależności od tego, jakie drzewa MIB są obsługiwane przez agenta, którym zarządzasz, kompletny zrzut może się składać z setek wierszy. Prawdę mówiąc, pełna instalacja systemu Ubuntu skonfigurowanego do obsługi każdego MIB wyrzuca ponad 12 000 wierszy!

Jeśli przejrzysz drzewa MIB<sup>9</sup> w najnowszej wersji Net-SNMP w systemie Ubuntu, zobaczysz, że OID określający średnie obciążenie z ostatnich 5 minut to 1.3.6.1.4.1.2021.10.1.3.2. Jeśli chciałbyś zobaczyć to obciążenie dla lokalnego hosta (skonfigurowanego z łańcuchem wspólnotowym „public”), wpisz polecenie:

```

$ snmpget -v 2c -c public localhost .1.3.6.1.4.1.2021.10.1.3.2
iso.3.6.1.4.1.2021.10.1.3.2 = STRING: "0.08"

```

Wiele użytecznych modułów Perla, Ruby’ego i Pythona związanych z SNMP jest dostępnych w odpowiednich repozytoriach dla tych języków. Co prawda możesz pisać skrypty w formie poleceń Net-SNMP, ale zwykle łatwiej i prościej jest skorzystać z modułów dostosowanych do wybranego języka.

## 28.10. KRUCZKI I SZTUCZKI

Na przestrzeni lat zebraliśmy kilka wskazówek dotyczących maksymalizowania skuteczności monitoringu. Oto najważniejsze z nich:

- Unikaj zmęczenia spowodowanego monitoringiem. Dopilnuj, aby administratorzy systemów otrzymujący powiadomienia poza swoimi normalnymi godzinami pracy mieli regularne przerwy. Cel ten najlepiej osiągnąć w systemie rotacyjnym, gdzie w danym dniu albo tygodniu wzywane są dwu- lub kilkuosobowe zespoły, po czym zadanie przejmuje kolejny zespół. Nieprzestrzeganie tej zasady sprawi, że administratorzy staną się zgorzkniali i zniechęcą swoją pracę.
- Zdefiniuj okoliczności, które naprawę wymagają uwagi 24 godziny na dobę przez 7 dni w tygodniu, i upewnij się, że informacje te zostały jasno przekazane Twojemu zespołowi monitorującemu, zespołom dyżurującym oraz obsługiwanym klientom lub jednostkom biznesowym. Sam fakt, że coś monitorujesz, nie oznacza, że administratorzy powinni być wzywani o 3:30 nad ranem, gdy jakaś wartość przekroczy normę. Wieloma kwestiami można się zająć w normalnych godzinach pracy.
- Eliminuj szumy monitorowania. Jeśli generowane są fałszywe alarmy lub powiadomienia o usługach niekrytycznych, poświęć czas na ich zatrzymanie i naprawienie. W przeciwnym razie, podobnie jak w bajce o chłopcu, który ostrzegał przed wilkiem, w końcu wszystkie zgłoszenia przestaną być traktowane z należytą powagą.

<sup>9</sup> Sprawdź na [mibdepot.com](http://mibdepot.com) lub zainstaluj pakiet `snmp-mibs-downloader`.

- Twórz procedury dla wszystkiego. Każde ponowne uruchomienie, każdy reset lub każdą procedurę naprawczą należy udokumentować w formie umożliwiającej podjęcie odpowiednich działań respondentowi, który nie zna bezpośrednio danego systemu. Skutki nieposiadania takiej dokumentacji są takie, że problemy nie będą szybko naprawiane, popełniane będą błędy, a w nagłych przypadkach trzeba będzie wzywać dodatkowy personel. Do przechowywania tego typu dokumentacji idealnie nadają się strony wiki.
- Monitoruj platformę monitorującą. Ta porada wyda Ci się oczywista, gdy przegapisz krytyczny przestój, ponieważ platforma monitorująca również przestanie działać. Ucz się na naszych błędach i upewnij się, że coś będzie obserwować Twoje czujne oczy.
- Przegapiłeś przestój z powodu czegoś, co nie było monitorowane? Upewnij się, że zostało to dodane, aby następnym razem wyłapać problem.
- I wreszcie ostatnia, choć być może najistotniejsza zasada: żaden serwer ani żadna usługa nie mogą wejść do środowiska produkcyjnego bez uprzedniego dodania ich do systemu monitorowania. I nie ma tu żadnych wyjątków.

## 28.11. ZALECANA LITERATURA

Hecht James, *Rethinking Monitoring for Container Operations*, <http://thenewstack.io/monitoring-reset-containers>. Wiele interesujących szczegółów dotyczących strategii i filozofii monitorowania kontenerów.

Turnbull James, *The Art of Monitoring*, Amazon Digital Services, Seattle 2016.

Dixon Jason, *Monitoring with Graphite: Tracking Dynamic Host and Application Metrics at Scale*, O'Reilly Media, Sebastopol 2017.

# SKOROWIDZ

Pliki zostały uporządkowane alfabetycznie pod względem ostatniego elementu i w większości przypadków wymieniany jest tylko ten element. Przykładowo, aby znaleźć wpisy indeksu odnoszące się do pliku `/etc/mail/aliases`, szukaj pod hasłem „aliases”. Zostaliśmy zmuszeni do takiego postępowania przez producentów, którzy w każdym systemie ukrywają standardowe pliki w nowych i nietypowych katalogach.

## A

- ACL, access control lists, 183, 686
  - implementacje, 185
  - interakcje z trybami, 194
  - w stylu NFSv4, 190
  - w stylu POSIX, 186
  - w systemie FreeBSD, 186
- Active Directory, 307, 617, 624, 861
- administrator
  - systemu, 1165, 1170
    - główne zadania, 46
    - podstawowe narzędzia, 50
  - baz danych, 70
  - sieci, 69
- adresowanie
  - IP, 424
  - IPv6, 433
  - pakietów, 423
  - sprzętowe, 423
  - za pomocą nazw, 425
- adresy
  - anycast, 426
  - broadcast, 426
  - IP, 426, 432
  - MAC, 423
  - multicast, 426
  - prywatne, 431
  - przydzielanie, 431
  - sieciowe, 423
  - unicast, 426
- akceptacja systemów zgłoszeniowych, 1141
- akcje, 900
- aktualizacje
  - oprogramowania, 1019
  - plików strefowych, 584
- aliasy, 225
  - hostów, 1052
  - pocztowe, 646
- Amazon Web Services, 315, 324

- analiza
    - obciążenia wejścia-wyjścia, 1111
    - statycznego kodu, 992
    - użycia pamięci, 1109
    - użycia procesora, 1106
  - anatomia wiadomości pocztowej, 637
  - Ansible, 889, 926
    - akcje, 900
    - bazy konfiguracyjne, 903
    - bezpieczeństwo, 927
    - generowanie szablonów, 899
    - grupy dynamiczne i obliczane, 895
    - grupy klientów, 894
    - interakcja z Jinja, 899
    - iteracja, 898
    - listy zadań, 896
    - opcje dostępu, 904
    - parametry state, 898
    - przypisywanie zmiennych, 895
    - role, 902
    - scenariusze, 900
    - ustawienia klienta, 892
  - Apache, 736
  - API, 731
  - aplikacje
    - Adobe, 226
    - bezserwerowe, 735
    - pakietu Microsoft Office, 226
  - AppArmor, 129
  - AppleScript, 226
  - APT, Advanced Package Tool, 210
    - automatyzacja, 214
  - architekci, 70
  - architektura
    - Dockera, 954
    - planisty kontenerów, 976
    - sieci, 513
    - systemu rsyslog, 345
    - VPC, 487
    - ZFS, 811
  - argumenty wiersza poleceń, 243
  - ARP, 418, 440
  - artefakty, 991
    - budowania, 1011
  - atak, 1066
    - typu brute-force, 1024, 1036
    - typu DDoS, 446, 1017
  - ataki blokady usług, 675
  - atrybuty plików, 176
  - audyt bezpieczeństwa, 1031
  - autofs, 846
  - automatyczna
    - instalacja, 201
    - konfiguracja drukarki, 407
    - konfiguracja plików urządzeń, 379
  - automatyczne numerowanie hostów, 435
  - automatyzacja, 226, 323, 1137
    - APT, 214
    - instalacji FreeBSD, 203
  - automount, 853, 854
  - autoryzacja, 322
  - autouzgadnianie, 503
  - awarie dysków twardych, 761, 794
  - AWS, Amazon Web Services, 315, 324
    - dziennik konsoli, 327
    - instancja EC2, 325
    - kontrola podsystemów, 324
    - kończenie instancji, 327
    - wirtualna chmura prywatna, 484
- B**
- balanser obciążenia, 721, 723
  - bastion, 321
  - baza
    - aliasów, 649
    - danych DNS, 549
  - bazy konfiguracyjne, 903
  - bezklasowe trasowanie międzydomenowe, 430
  - bezpieczeństwo
    - aktualizacje oprogramowania, 1019
    - Ansible, 927
    - centrów danych, 1129
    - certyfikacja, 1061
    - CIS, 1064
    - Common Criteria, 1064
    - czujność, 1023
    - DNS, 587
    - filtrowanie pakietów, 1022
    - hasła, 1023, 1024
    - incydenty, 1150
    - komunikatów, 357
    - konta współużytkowane, 1027
    - kopie zapasowe, 1021
    - Metasploit, 1031
    - monitorowanie, 1089
    - narzędzia kryptograficzne, 1036
    - Nessus, 1030
    - NIST 800, 1063
    - OWASP, 1064
    - PCI DSS, 1063
    - PGP, 1045



podręczny audyt, 1031  
 Postfix, 699  
 reakcja na atak, 1066  
 rootkity, 1022  
 Samby, 865  
 serwera sendmail, 670  
 sieci, 444  
 sieci bezprzewodowych, 509  
 skaner portów, 1028  
 SSHFP, 1056  
 standardy, 1062  
 systemów komputerowych, 1013  
 testy penetracyjne, 1024  
 TLS, 1040  
 tunelowanie połączeń TCP, 1054  
 uwierzytelnianie wieloskładnikowe, 1023  
 VPN, 1060  
 wirusy i robaki, 1021  
 wykrywania włamań sieciowych, 1032  
 zapory sieciowe, 1058, 1059  
 zbędne usługi, 1020  
 zdalna powłoka SSH, 1046  
 zdalne logowanie zdarzeń, 1021  
 zunifikowane podejście, 1046  
 źródła informacji, 1064  
 bezpieczne  
   logowanie, 1057  
   przesyłanie wiadomości, 672  
 bezpieczny  
   transfer plików, 631  
   tryb obliczeniowy, 951  
 BGP, Border Gateway Protocol, 526  
 bhyve, 943  
 BIND, Berkeley Internet Name Domain, 562  
   kanały, 605  
   kategorie, 606  
   kategorie rejestrowania, 607  
   komponenty, 563  
   komunikaty dziennika, 606  
   konfiguracja, 580  
   konfiguracja rejestrowania, 609  
   lista kontroli dostępu, 588  
   opcje bezpieczeństwa, 588  
   poziomy szczegółowości, 609  
   rejestrowanie, 604  
 BIOS, 76  
 bit  
   lepkości, 177  
   setgid, 177  
   setuid, 177  
 bity uprawnień, 176

blokowanie  
   kont użytkowników, 305  
   plików, 834  
 blokowe urządzenia pamięci masowej, 322  
 błędy  
   jądra, 396  
   konfiguracji  
     aplikacji, 1018  
     sieci, 1018  
     systemu, 1018  
 Bro, 1032  
 Btfs, 819  
 budowanie  
   bazy aliasów, 649  
   obrazu, 1003  
 bufor  
   odwrotnego serwera pośredniczącego, 727  
   serwera pośredniczącego, 727  
 buforowanie, 727

## C

Cacti, 475  
 całkowite obciążenie cieplne, 1126  
 camcontrol, 775  
 CDN, content delivery network, 728  
 centra danych  
   poziomy niezawodności, 1129  
 centralny komputer rejestrujący, 357  
 centrum danych, 1120  
 certyfikacja, 1061  
 certyfikat, 1043  
 Chef, 886  
 chgrp, 181  
 chłodzenie, 1124  
   centrum danych, 1126  
 chmod, 179  
 chmura, 1173  
   Amazon Web Services, 315  
   automatyzacja, 323  
   autoryzacja, 322  
   DigitalOcean, 316  
   hosting WWW, 733  
   funkcje bezserwerowe, 323  
   Google Cloud Platform, 316  
   kontrola kosztów, 331  
   obliczeniowa, 311, 315  
   tożsamość, 322  
   uruchamianie niestandardowych jąder, 395  
   VPC, 487

chmury  
   hybrydowe, 314  
   prywatne, 314  
   publiczne, 314  
 chown, 181  
 chroot, 590  
 CI/CD, 985, 998, 1010  
   potoki, 990  
   serwer Jenkins, 995  
   środowiska, 988  
 ciągła integracja, 983, 985  
 ciągle  
   dostarczanie, 983, 985  
   wdrażanie, 985  
 CIDR, Classless Inter-Domain Routing, 430  
 CIP, 1157  
 CIS, 1064  
 Cobbler, 203  
   rozruch sieciowy, 203  
 collectd, 1086  
 Common Criteria, 1064  
 COPPA, 1156  
 cron, 152, 156  
 CUPS, 402  
 curl, 718  
 cut, 235  
 cykl życia procesu, 137  
 czarne listy, 668

## D

dane LDAP, 618  
 DDoS, 1017  
 dedykowane serwery plików, 848  
 definiowanie  
   podpowiedzi, 576  
   poziomu usług, 1153  
 delegowania, 543  
 demon, 83, 521  
   cron, 152  
   cupsd, 406  
   Dockera, 973  
   nfsd, 842  
   nginx, 744  
   sssd, 626  
   trasujący, 528  
   wydruku, 409  
 detekcja włamań, 1032  
 devd, 379  
 devfs, 379

DevOps, 69, 71, 1135  
   administracja systemem, 1138  
   zasady, 1135  
 DHCP, 441  
   działanie, 442  
   oprogramowanie, 441, 443  
 diagnostyka  
   konfiguracji, 359  
   serwera sendmail, 676  
   serwerów TLS, 1044  
   sieci, 510  
   systemu BIND, 604  
 DigitalOcean, 316, 329, 492  
 DKIM, DomainKeys Identified Mail, 644  
 DNS, 535  
   aktualizowanie plików strefowych, 584  
   analizator, 549  
   architektura, 536  
   automatyczne aktualizacje, 585  
   baza danych, 549  
   bezpieczeństwo, 587  
   buforowanie, 545  
   delegowania, 543  
   diagnostyka, 546  
   dokumenty RFC, 613  
   dostawcy usług, 537  
   efektywność, 545  
   konfiguracja, 452  
   obsługa zapytania, 544  
   odpowiedzi wielokrotne, 545  
   przestrzeń nazw, 539  
   przesyłanie informacji strefowych, 584  
   rejestracja nazwy domeny, 540  
   rekordy zasobów, 543, 550  
   rndc, 610  
   rozdzielony, 578  
   rozszerzenia DNSSEC, 593  
   równoważenie obciążenia, 545  
   serwery autorytatywne i buforujące, 542  
   serwery nazw, 541  
   serwery rekurencyjne, 542  
   system BIND, 562  
   typy rekordów, 552  
   wyszukiwania, 538  
   wyszukiwanie niepoprawnych delegowań, 611  
 DNSSEC, 593, 594  
   generowanie par kluczy, 596  
   łańcuch zaufania, 600  
   narzędzia, 601, 602  
   podpisywanie stref, 598

- rekordy zasobów, 594
  - usuwanie błędów, 603
  - włączanie, 596
  - wymiana kluczy, 600
  - doc, domain obscenity control, 612
  - Docker, 953
    - architektura, 954
    - bezpieczeństwo, 972
    - budowanie obrazów, 966
    - dostęp do demona, 973
    - konfigurowanie klienta, 955
    - nakładki sieciowe, 963
    - przestrzenie nazw, 961
    - rejestratory, 972
    - repozytoria, 969
    - sieci, 961
    - silnik kontenerowy, 953
    - sterowniki pamięci masowej, 964
    - usuwanie błędów, 975
    - woluminy, 959
  - Docker Swarm, 979
  - dockerd, 964
  - dodawanie
    - dysków, 812
    - sterownika urządzenia, 385
    - użytkowników, 304
  - dokumentacja, 57, 1143, 1171
    - autorytatywna, 59
    - pakietów, 59
    - Salt, 925
  - dokumenty
    - HOWTO, 61
    - internetowe, 60
    - RFC, 60, 417, 533, 613
  - domeny wirtualnych skrzynek pocztowych, 705
  - domyślne zachowania sieciowe, 461
  - dopasowywanie, 250
    - ślug, 910
  - dostawca chmury, 314, 319
  - dostęp
    - do chmury, 318
    - do demona, 973
    - do pliku, 189, 830
    - w NFSv4, 192
  - dostępność, 1015
  - dostrajanie wydajności, 1098
  - dowiązania
    - symboliczne, 171, 175
    - twarde, 173
  - Dropbear, 1057
  - drukarki sieciowe, 404, 407
  - drukowanie, 401
    - bezppośrednie, 411
    - filtry, 405
    - interfejsy, 403
    - kolejka drukowania, 403
    - konfiguracja drukarki, 407
    - konfiguracja serwera, 407
    - problemy, 411
    - sieciowe, 404, 411
    - wyłączenie usługi, 408
    - zadania konfiguracyjne, 409
  - drzewo plików, 168
  - DS, Designated Signer, 594
  - dtrace, 1086
  - dynamiczna konfiguracja hostów, 441
  - dyski
    - hybrydowe, 766
    - korporacyjne, 762
    - NAS, 762
    - PATA, 774
    - SATA, 774
    - SSD, 755, 763–765
    - twarde, 760
  - dystrybucje systemu, 51, 53
    - Arch, 52
    - CentOS, 52
    - CoreOS, 52
    - Debian, 52
    - Fedora, 52
    - Kali, 52
    - Linux Mint, 52
    - openSUSE, 52
    - openWRT, 52
    - Oracle Linux, 52
    - RancherOS, 52
    - Red Hat Enterprise, 52
    - Slackware, 52
    - SUSE Linux Enterprise, 52
    - Ubuntu, 52
  - działania arytmetyczne, 249
  - dziennik konsoli, 327
  - dzienniki, 338
- ## E
- EC2, 325
  - ECS, Elastic Container Service, 980
  - edycja
    - plików, 296
    - poleczeń, 231

efemeryczna pamięć masowa, 322  
 EFI, Extensible Firmware Interface, 783  
 EGID, 136  
 EIGRP, Enhanced Interior Gateway Routing  
   Protocol, 526  
 e-mail, 160, 633, 637  
 enkapsulacja, 421  
 etapy budowania, 1005  
 Ethernet, 496  
   przesyłanie sygnałów, 496  
   ramki, 504  
   rozszerzanie sieci, 501  
   topologia, 498  
 EUID, 135  
 ewidencjonowanie, 878  
 Exim, 678  
   diagnostyka, 696  
   filtrowanie, 693  
   instalacja serwera, 679  
   język konfiguracji, 682  
   konfiguracja ponownie, 694  
   konfiguracja przepisywania, 695  
   listy, 685  
   lokalna funkcja skanująca, 695  
   makra, 685  
   mechanizm ACL, 686  
   mechanizmy uwierzytelniające, 689  
   narzędzia serwera, 681  
   opcje globalne, 684  
   plik konfiguracyjny, 683  
   skanowanie treści, 689  
   transporty, 693  
   uruchamianie serwera, 681  
   zapisywanie dzienników, 695

## F

Fail2Ban, 1036  
 fałszowanie adresów IP, 446  
 FERPA, 1156  
 filtrowanie, 693  
   iptables, 478  
   pakietów, 480, 1022  
   usług, 1058  
 filtry, 405  
   pakietów, 472  
 fio, 1112  
 FISMA, 1156  
 Fluke LanMeter, 510  
 format  
   crontab, 153  
   LDIF, 619

formatowanie, 773  
   systemu plików, 802  
 formuły Salt, 921  
 FreeBSD, 81  
   Active Directory, 624  
   automatyzacja instalacji, 203  
   bhyve, 943  
   budowanie jądra, 386  
   domyślne parametry sieciowe, 464  
   dostrajanie parametrów jądra, 385  
   Kerberos, 624  
   kolekcja portów, 218  
   komunikaty rozruchowe, 394  
   konfiguracja  
     TCP/IP, 463  
     sprzętu sieciowego, 462  
   ładowalne moduły jądra, 389  
   obsługa ACL, 186  
   panika jądra, 399  
   partycjonowanie, 784  
   pkg, 217  
   sieci, 461  
   skrypty startowe, 99  
   system bazowy, 216  
   ścieżka  
     BIOS-u, 81  
     UEFI, 82  
   tryb pojedynczego użytkownika, 104  
   wersje jądra, 370  
   zarządzanie  
     dziennikami, 362  
     oprogramowaniem, 215  
     urządzeniami, 378  
     woluminami logicznymi, 790

fsck, 802  
 FTC Red Flag Rules, 1157  
 FTC Safe Harbor, 1157  
 funkcja, 243  
   access\_db, 661  
   always\_add\_domain, 661  
   ldap\_routing, 662  
   redirect, 661  
   use\_cw\_file, 660  
   virtusertable, 662  
 funkcje  
   bezserwerowe, 323  
   skrót, 1040  
   stanowe, 916  
   systemów zgłoszeniowych, 1140  
   wykonawcze, 916

**G**

gcloud, 328  
 GCP, Google Cloud Platform, 316, 328, 490  
   konfigurowanie gcloud, 328  
   uruchamianie instancji, 328  
 generowanie  
   liczb losowych, 1042  
   szablonów, 899  
 GID, 136, 289  
 Git, 274  
   społecznościowe tworzenie kodu, 278  
   zastrzeżenia, 278  
 GLBA, 1157  
 gniazda, 341  
   lokalne, 171, 174  
 Go, 731  
 Google Cloud Platform, 316, 328  
 GPT, 783  
 Graphite, 1076  
 Graylog, 363  
 grep, 238  
 GRUB, 78  
   konfiguracja programu, 79  
   opcje konfiguracyjne, 79  
   polecenia, 81  
   wiersz poleceń, 80  
 grupowa koordynacja protokołów, 527  
 grupowanie kontenerów, 976  
 grupowe adresy protokołów, 527  
 grupy  
   kontrolne, 148, 951  
   woluminów, 778  
   zabezpieczeń, 486

**H**

HAProxy, 748  
   kontrolowanie stanu serwera, 749  
   lepkie sesje, 750  
   statystyki serwera, 750  
   terminacja TLS, 751  
 harmonogram poleceń, 152  
 hasła, 117, 287, 1023, 1024  
   menedżery, 1025  
   okres ważności, 1027  
   słabe, 1031  
   zmiany, 1025  
 hasło, 297  
 hasło zaszyfrowane, 286  
 hdparm, 775

head, 237  
 HIPAA, 1157  
 hipernadzorca, 934  
 hipernadzorca parawirtualizacyjny, 939  
 host wirtualny, 739  
 hosting, 68  
   treści statycznych, 735  
   WWW, 713  
   WWW w chmurze, 733  
 hostowane platformy monitorujące, 1079  
 hosty wirtualne, 720  
 HTTP, 714  
   kategorie odpowiedzi, 717  
   metody żądań, 716  
   oprogramowanie pośredniczące, 722  
   podstawowe uwierzytelnianie, 740  
   przez TLS, 720  
   struktura transakcji, 716  
   z wiersza poleceń, 718  
 httpd, 736  
   rejestrowanie zdarzeń, 742  
   ustawienia konfiguracyjne, 737  
 hub, 501

**I**

IaaS, 318  
 IaaS, Infrastructure-as-a-Service, 317  
 IAM, identity and access management, 308  
 Icinga, 1075  
 ICMP, 418, 439  
   przekierowania, 445  
 identyfikator  
   grupy, 136  
   procesu, 134  
   procesu macierzystego, 135  
   stanów, 914  
   użytkownika, 135  
   zależności, 914  
 ifconfig, 462  
 implementacja serwera LDAP, 620  
 incydenty bezpieczeństwa, 1150  
 InfluxDB, 1077  
 informacje  
   o bezpieczeństwie, 1064  
   o IPv6, 437  
   o pakietach, 208  
   o procesach, 148  
   o stanie procesu, 148  
   o użyciu pamięci, 148  
   strefowe, 584

- infrastruktura
    - bezprzewodowa, 506
    - jako kod, 1144
    - jako usługa, IaaS, 317
    - klucza publicznego, 1038
  - init, 86, 92, 99
  - instalacja
    - automatyczna, 201
    - Dockera, 955
    - dysku, 771
    - oprogramowania, 62
    - oświetleniowa, 1126
    - pakietu OSSEC, 1034
    - przez sieć, 196
    - serwera Exim, 679
    - serwera Samba, 859
    - systemów operacyjnych, 196
    - UTP, 499
    - ze skryptu WWW, 67
    - ze źródeł, 66
  - instalator, 201
  - instalatory pakietów, 205
  - instalowanie
    - oprogramowania, 64
    - plików startowych, 298
  - instancja EC2, 325
  - instancje drukarek, 404
  - instrukcja
    - acl, 571
    - controls, 577
    - include, 565
    - key, 571
    - logging, 573
    - masters, 573
    - options, 565
    - server, 572
    - statistics-channels, 573
    - view, 578
    - zone, 574
  - instrukcje
    - Dockerfile, 968
    - instalacji, 196
    - w named.conf, 564
  - instrumentacja, 1071
  - integralność, 1015
    - systemu, 1089
  - interfejs
    - API, 323
    - PCI Express, 768
    - podsystemu drukowania, 403
    - programowania aplikacji, API, 731
    - SAS, 769
    - SATA, 768
    - sieciowy, 449
    - urządzeń pamięci masowej, 768
  - internet, 416
  - inwentaryzacja sprzętu, 1103
  - inżynierowie
    - ds. bezpieczeństwa, 69
    - ds. niezawodności, 69
    - sieciowych centrów operacyjnych, 70
  - IP, 418, 444
    - falszowanie adresów, 446
    - konfiguracja, 449
    - śledzenie pakietów, 467
  - iPerf, 474
  - IPFilter, 481
  - iptables, 476
    - cele reguł, 477
    - konfiguracja zapory, 478
    - opcje filtrowania, 478
  - IPv4, 419
  - IPv6, 419
    - adresowanie, 433
    - prefiksy, 435
    - tunelowanie, 437
    - zapis adresów, 434
  - ISC, 443
  - ITGC, 1158
  - ITIL, 1158
- J**
- Java, 730
  - jądro, 359, 367, 951
    - błędy, 396
    - budowanie, 381, 383
    - dostrajanie parametrów, 380
    - konfiguracja, 380, 385
    - konfigurowanie opcji, 383
    - ładowalne moduły, 387, 389
    - panika, 399
    - uruchamianie w chmurze, 395
    - wersje, 369, 370
  - jednostki, 56, 87
    - celu, 91
    - kolejność wykonywania, 94
    - mocy, 1123
    - NFSv4, 191
    - statusy, 89
    - zależności, 93

Jenkins, 995  
 potok, 1001  
 język  
   Forth, 83  
   Go, 731  
   Java, 730  
   JavaScript, 228  
   Perl, 228  
   PHP, 228, 731  
   powłoka Bourne'a, 228  
   powłoka C, 228  
   Python, 228, 255, 730  
   Ruby, 228, 262, 729  
 języki  
   sieci WWW, 729  
   skryptowe, 227  
 Jinja, 899, 913  
 JSON, 731

## K

kable UTP, 499  
 kanał oczekiwania, 143  
 katalog, 170, 172  
   /proc, 148  
   /sys, 374, 382  
   domowy, 290  
   network-scripts, 98  
 kategorie rejestrowania, 607  
 Kerberos, 124, 623, 1046  
   integracja z Active Directory, 624  
 Kickstart, 198  
   plik konfiguracyjny, 198, 200  
   tworzenie serwera, 200  
 kill, 140  
 klastry stron, 765  
 klasy adresów IPv4, 427  
 klauzula match-destinations, 579  
 klient  
   bezprzewodowy, 506  
   poczty, 635  
   rejestrujący, 356  
 klucz  
   TLS, 741  
   ZSK, 596  
 klucze  
   dopasowań udev, 379  
   dopasowań udevd, 376  
   publiczne, 1037, 1038  
   symetryczne, 1037  
 kolejka drukowania, 403  
 kolekcja portów, 218  
 komponenty BIND, 563  
 komunikaty  
   dziennika, 336  
   jądra, 359  
   rozruchowe, 390  
   rozruchowe systemu FreeBSD, 394  
   systemu syslog, 344  
 koncentratory, 501  
 konferencje, 62  
 konfigurowanie  
   BIND, 580  
   DNS, 452  
   drukarki, 407  
   drukarki sieciowej, 407  
   gcloud, 328  
   HAProxy, 748  
   hostów wirtualnych, 739  
   httpd, 737  
   interfejsów sieciowych, 462  
   IP, 449  
   iptables, 476  
   jądra, 380, 385  
   klienta, 664, 955  
   macierzy, 797  
   mechanizmu PAM, 628  
   opcji jądra, 383  
   oprogramowania, 219  
   pakietu OSSEC, 1035  
   PAM, 629  
   pamięci masowej, 820  
   plików urządzeń, 379  
   ponowień, 694  
   programu  
     GRUB, 79  
     loader, 83  
 PXE, 198  
 rejestratora systemd, 341  
 rejestrowania  
   w BIND, 609  
   zdarzeń, 866  
 repozytorium, 212  
 resolvera klienta, 538  
 ról, 300  
 serwera  
   Apache, 737  
   poczty, 649  
   NGINX, 744  
   Postfix, 700  
   Samba, 859  
   sendmail, 655, 656  
   strefy, 574, 575

konfigurowanie  
   sieci, 448, 453, 456, 463  
   sieciowego serwera wydruków, 407  
   sprzętu sieciowego, 462  
   strefy przekazującej, 577  
   sudo, 119  
   systemu rsyslog, 346  
   TCP/IP, 463  
   TLS, 740, 747  
   tras, 451  
   TSIG, 591  
   udziałów, 861  
   zapory iptables, 478  
 konta  
   systemowe, 121  
   współużytkowane, 1027  
 kontenery, 949, 971, 1010  
   danych, 960  
   EC2, 980  
   grupowanie, 976  
   jako środowisko budowania, 1011  
   obrazy, 951, 1011  
   sieć, 952  
   silnik kontenerowy, 953  
   zarządzanie, 977  
 konteneryzacja, 938  
 konto  
   nobody, 837  
   użytkownika root, 110  
 kontrola  
   dostępu, 107, 108  
     AppArmor, 129  
     do programu cron, 156  
     do systemu plików, 124  
     listy, 124  
   nowoczesne mechanizmy, 125  
   obowiązkowa, 126  
   oddzielne ekosystemy, 126  
   oparta na rolach, 127, 300  
   rozszerzenia, 122  
   Security-enhanced Linux, 128  
   w systemie plików, 109  
   wady modelu standardowego, 122  
   zaawansowana, 116  
   kosztów, 331  
   podsystemów aws, 324  
   procesów, 133  
   rozruchu, 96  
   wersji, 274  
   wydajności systemu, 1103  
 konwertowanie pamięci masowej, 820

kopie  
   migawkowe, 815  
   migawkowe woluminów, 787  
   zapasowe, 161, 1021  
 kopiowanie przy zapisie, 808  
 krople, 105  
 kryptografia, 1036  
   klucza publicznego, 1037  
   klucza symetrycznego, 1037  
 kryptograficzne funkcje skrótu, 1040  
 Kubernetes, 977  
 KVM, Kernel-based Virtual Machine, 942

## L

LAN, Local Area Network, 496  
 LDA, 636  
 LDAP, 307, 617  
   atrybuty, 619  
   implementacja serwera, 620  
   konwertowanie plików, 622  
   specyfikacje RFC, 632  
   struktura danych, 618  
   zapytania, 621  
   zastosowania, 618  
 LDIF, LDAP Data Interchange Format, 619  
 ldns, 602  
 lenistwo, 254  
 licencje na oprogramowanie, 1161  
 LINUX, 1171, 1174  
 listy kontroli dostępu, ACL, 124, 183  
 LKM, loadable kernel modules, 387  
 loader, 83  
   konfiguracja programu, 83  
   polecenia, 83  
 localhost, 580  
 logowanie, 111, 623  
 logrotate, 360  
   opcje programu, 361  
   zarządzanie dziennikami, 360  
 lokalizowanie oprogramowania, 219  
 lokalne wytwarzanie energii, 1121  
 ls, 177  
 Lynis, 1031

## Ł

ładowne moduły jądra, 387  
 łańcuchy, 476  
 łącze symboliczne, 148



**M**

MAC, Mandatory Access Control, 126, 423

macierz

nadmiarowa, 790

RAID, 778

magazyny obiektów, 321

makro

DOMAIN, 660

FEATURE, 660

MAIL\_HUB, 664

MAILER, 660

MASQUERADE\_AS, 663

OSTYPE, 660

SMART\_HOST, 664

maksymalna jednostka transmisji, 422

maper urzędzeń, 779

Marathon, 978

maskarada, 664

maszyna wirtualna, 937, 939

MBR, 81, 782

mdadm, 795

mechanika konta, 284

mechanizm

ACL, 686

automount, 854

Kerberos, 624

kontroli dostępu, 125

PAM, 284, 627

TSIG, 590

uwierzytelniania, 627, 689

menedżer

pakietów FreeBSD, 217

haseł, 1025

Mesos, 978

Metasploit, 1031

metodologia, 1133

MFA, multi-factor authentication, 324

miary kosztu, 523

migawki woluminów, 823

migracja w locie, 937

mikroskrypty, 224

model

biznesowy, 880

standardowy, 122

warstw TCP/IP, 419

moduły uwierzytelniania, 306

monitoring, 1069

aplikacji, 1087

bezpieczeństwa, 1089

dysku twardego, 776

dzienników, 1087

hostowane platformy, 1079

integralności plików, 1089

interaktywny procesów, 144

interfejsy użytkownika, 1073

komercyjne platformy, 1079

maksymalizowanie skuteczności, 1095

panele, 1073

platformy, 1074

czasu rzeczywistego, 1075

pobieranie i przetwarzanie, 1072

polecenia, 1085

powiadomienia, 1072

pozyskiwanie danych, 1082

systemowych, 1086

procesów, 142

sieci, 473, 1083

systemów, 1084

śledzenie działań, 1086

środowiska, 1128

tworzenia wykresów, 1078

wykrywania włamań, 1090

zbieranie danych, 1080

montowanie

napedów USB, 807

plików, 863

systemu plików, 804, 849

zdalnych systemów plików, 846

możliwości, 951

systemu, 124

MSA, 634, 635

MTA, 634, 636

MTU, 422

MUA, 634

multipleksacja połączeń, 1053

Munin, 1078, 1088

MX, mail exchanger, 558

**N**

NACL, Network Access Control Lists, 484, 486

nadużycia wewnętrzne, 1018

Nagios, 1075

nagłówki, 717

nakładki sieciowe, 963

naruszenie bezpieczeństwa, 1015

narzędzia, 50, 71

administratora systemu, 1131

bezpieczeństwa, 1028

dla serwerów, 1093

DNSSEC-Tools, 602

- narzędzia
    - do monitorowania aplikacji, 1088
    - do odpytywania, 546
    - do wyliczania podsieci, 429
    - kryptograficzne, 1036
    - serwera Exim, 681
  - narzędzie, *Patrz* program
  - NAS, Network Attached Storage, 762
  - NAT, 431, 476, 480
  - nazwa
    - jądra, 369
    - komputera, 423, 448
    - pliku, 243
    - urządzenia, 773
    - użytkownika, 285
  - nazwane potoki, 175
  - ND, 440
  - Nessus, 1030
  - Net-SNMP, 1093
  - NetworkManager, 454
  - newsyslog, 362
    - zarządzanie dziennikami, 362
  - newusers, 304
  - NFS, 827, 830
    - bezpieczeństwo, 834, 836
    - blokowanie plików, 834
    - dedykowane serwery plików, 848
    - dostęp z uprawnieniami root, 837
    - eksporty, 833
    - kontrola stanu, 829
    - montowanie automatyczne, 849
    - odwzorowania
      - bezpośrednie, 851
      - główne, 851
      - pośrednie, 851
      - tożsamości, 847
      - wykonywalne, 852
    - po stronie klienta, 844
    - statystyki połączeń, 848
    - wersje protokołu, 831
    - wydajność, 829
    - zdalne wywoływanie procedur, 832
  - nfsd, 842
  - nfsstat, 848
  - NFSv4, 190
    - dziedziczenie wpisów ACL, 192
    - określanie dostępu, 192
    - podgląd list ACL, 193
    - ustawienia list ACL, 194
  - NGINX, 743
    - instalacja i uruchamianie serwera, 743
    - konfigurowanie serwera, 744
    - konfigurowanie TLS, 747
    - równoważenie obciążenia, 747
  - nieodwracalność, 1041
  - NIS, Network Information Service, 630
  - NIST, 1063
  - nmap, 1028
  - Node.js, 730
  - NS, name server, 555
  - NSS, Name Service Switch, 627
  - null client, 700
  - numer
    - GID, 289
    - UID, 288
  - numerowanie wersji jądra, 369
  - numery urządzeń, 371
- O**
- obrazy
    - kontenerów, 1011
    - maszyn wirtualnych, 937
  - obsługa
    - dysków, 771
    - kontenerów EC2, 980
    - poczty elektronicznej, 634
    - zmian, 876
  - obszar wymiany, 807
  - ochrona prywatności, 1159
  - odbicia lustrzane, 161
  - odpowiedzi HTTP, 717
  - odzyskiwanie systemów chmurowych, 104
  - okablowanie, 511
  - opcje
    - architekuralne, 880
    - bezpieczeństwa w BIND, 588
    - dockerd, 964
    - dodatkowe, 182
    - dostępu Ansible, 904
    - filtrowania iptables, 478
    - jądra systemu, 80
    - językowe, 883
    - konfiguracyjne klienta SSH, 1049
    - konfiguracyjne m4, 665
    - polecenia dnssec-signzone, 599
    - sprzętu sieciowego, 458
    - TCP/IP, 459
    - TLS, 973
    - zarządzania zależnościami, 884
  - OpenDNSSEC, 603
  - OpenLDAP, 620

OpenSSH, 1047  
 openssl, 1043  
 operacje wejścia-wyjścia, 1113  
 operatorzy systemów, 1166  
 oprogramowanie  
   BIND, 562  
   buforujące, 727  
   DHCP, 441, 443  
   jako usługa, SaaS, 317  
   kryptograficzne, 1043  
   sprzętowe systemu, 75  
   wirtualizacyjne, 933  
 organizacja procesu lokalizowania, 219  
 organizacje, 1162  
   ICANN, 416  
   IGF, 416  
   ISOC, 416  
 OSPF, Open Shortest Path First, 526  
 OSSEC, 1033, 1034  
   instalacja pakietu, 1034  
   konfiguracja pakietu, 1035  
 otwarty resolver, 589  
 OWASP, 1064  
 oznakowanie, 233

## P

PaaS, 318  
 PaaS, Platform-as-a-Service, 317, 734  
 Packer, 944  
 paczki, 877  
 pakiet, 421, 518  
   ethernetowy, 520  
   gem, 991  
   IP, 444  
   Net-SNMP, 1094  
   OSSEC, 1034  
   PGP, 1045  
   ping, 445  
   pip, 991  
   RPM, 206  
   virtualenv, 271  
 pakiety  
   instalowanie, 269  
   wyszukiwanie, 269  
 PAM, Pluggable Authentication Modules, 123,  
   284, 306, 627  
   konfiguracja, 628, 629  
   znaczniki kontrolne, 629  
 pamięć  
 flash, 764  
   masowa, 321, 755  
   podręczna, 725  
   podręczna przeglądarki, 726  
 panika jądra, 399  
 parametry, 917  
   sieciowe, 464  
 parawirtualizacja, 935  
 partycja, 778  
 partycje MBR, 782  
 partycjonowanie  
   dysków, 780  
   tradycyjne, 781  
   w systemie FreeBSD, 784  
   w systemie Linux, 784  
 PCI DSS, 1063, 1157  
 pętle, 247, 261  
 PGP, Pretty Good Privacy, 1045  
 PHP, 731  
 PID, 134  
 ping  
   gromadzenie statystyk, 473  
 pisanie skryptów, 228  
 pkg, 217  
 plan naprawy, 1148  
 planista operacji wejścia-wyjścia, 1113  
 platforma  
   AWS, 324  
   GCP, 68  
   jako usługa, PaaS, 317, 734  
 platformy  
   chmur obliczeniowych, 314  
   czasu rzeczywistego, 1075  
   do tworzenia wykresów, 1078  
   monitorujące, 1074  
   szeregów czasowych, 1076  
 plik  
   apache2, 339  
   apt, 339  
   auth.log, 339  
   autoclose, 382  
   boot.log, 339  
   cloud-init.log, 339  
   config, 1052  
   cron, 339  
   ctrl-alt-del, 382  
   daemon.log, 339  
   debug, 339  
   dmesg, 339  
   Dockerfile, 966, 967  
   dpkg.log, 339  
   exports, 839, 841

## plik

- faillog, 339
- file-max, 382
- group, 294, 296
- icmp\_echo\_ignore\_all, 382
- ip\_forward, 382
- iptables-config, 98
- kern.log, 339
- kmsg, 341
- lastlog, 339, 340
- local\_port\_range, 382
- login.conf, 293
- mail, 339
- main.cf, 700
- master.passwd, 123, 292
- mdadm.conf, 797
- named.conf, 563, 564, 581
- nsswitch.conf, 538, 627
- overcommit\_memory, 382
- overcommit\_ratio, 382
- panic, 382
- panic\_on\_oops, 382
- passwd, 285, 296
- printk\_ratelimit, 382
- printk\_ratelimit\_burst, 382
- resolv.conf, 538
- rp\_filter, 382
- samba, 339
- secure, 339
- shadow, 123, 290
- shmmax, 382
- smb.conf, 860
- socket, 341
- sources.list, 212
- sudoers, 114, 118
- switch, 652
- syslog, 339
- tcp\_fin\_timeout, 382
- tcp\_synccookies, 382
- udev.conf, 375
- wtmp, 339
- xen, 339
- yum.log, 339

## pliki

- .deb, 991
- .exe, 991
- .forward, 693
- .jar, 991
- .mc, 657
- .rpm, 991
- .sls, 916

- .war, 991
- atrybuty, 176
- binarne jądra, 384
- cookie, 750
- crontab, 153, 155, 156
- danych o strefie, 584
- dodatkowe atrybuty, 182
- dzienników, 161, 338, 360, 410
- jednostek, 87, 91, 94
- konfiguracyjne, 198
  - sieci, 456
  - serwera Exim, 683
- reguł, 375
- spacje w nazwach, 243
- startowe, 299
- strefowe, 549
- typy, 171
- urządzeń, 371, 372
  - automatyczna konfiguracja, 379
  - blokowych, 171, 173
  - dyskowych, 772
  - znakowych, 171, 173
- z dziennikami, 338
- zwykle, 172

plytkie kopie, 823

poczta elektroniczna, 633

- aliasy pocztowe, 646
- anatomia wiadomości, 637
- czarne listy, 668
- DKIM, 644
- dokumenty RFC, 711
- klienty, 635
- kody błędów SMTP, 641
- kolejki pocztowe, 654
- konfiguracja serwera poczty, 649
- kontrola
  - dostępu, 706
  - przekazywania, 667
- mechanizmy uwierzytelniające, 689
- odbijanie wiadomości, 710
- odczyt aliasów, 648
- oszustwa, 643
- plik switch, 652
- preprocesor m4, 655
- protokół SMTP, 640
- prywatność, 645
- routery, 690
- Sender ID, 644

serwer

- Exim, 678
- Postfix, 697
- sendmail, 651, 652, 656

- skrzynki pocztowe, 637
- spam, 643
- SPF, 644
- system
  - dostarczania lokalnego, 636
  - dostępowy, 637
  - obsługi, 634
  - przyjmujący, 635
  - transportowy, 636
- szyfrowanie, 645, 708
- uwierzytelnianie
  - klientów, 708
  - SMTP, 642
  - wysyłanie wiadomości, 648
- podatności oprogramowania, 1016
- podłączenie dysku, 756, 758
- podpisywanie strefy, 596, 598
- podpowiedzi, 576
- podręcznik systemowy, 57
- podsieci, 428, 485
- podsluchiwanie pakietów, 470, 471
- podstawianie tożsamości, 111
- podwoluminy, 822
- podział wierszy, 235
- pojedyncze logowanie, 615
- pole GECOS, 289
- polecenia
  - Dockera, 954
  - filtrujące, 235
  - pakietu IPFilter, 483
  - plików strefowych, 549
  - podsystemu CUPS, 410
  - SMTP, 641
- polecenie
  - apachectl, 737
  - camcontrol, 775
  - chgrp, 181
  - chown, 181
  - chmod, 179
  - cmd, 148
  - cut, 235
  - dnssec-keygen, 597
  - dnssec-signzone, 599
  - fsck, 802
  - grep, 238
  - halt, 101
  - hdparm, 775
  - head, 237
  - ifconfig, 462, 506
  - ifdown, 454, 456
  - ifup, 456
  - ip, 455
  - ipf, 482
  - iwconfig, 506
  - iwlist, 506
  - kill, 140
  - logrotate, 118
  - lpadmin, 408
  - ls, 177
  - lsblk, 757
  - newusers, 304
  - nfsstat, 848
  - nice, 146
  - openssl, 1043
  - ping, 465
  - pkg, 217
  - postconf, 701
  - ps, 142, 143
  - renice, 146
  - rndc, 611
  - rpm, 206
  - smbstatus, 865
  - smokeping, 473
  - sort, 235
  - strace, 148
  - su, 112
  - sudo, 113, 115
  - tail, 237
  - tee, 237
  - top, 144
  - traceroute, 467
  - truss, 148
  - udevadm, 374
  - umask, 181
  - uniq, 236
  - useradd, 301, 302, 303
  - wc, 237
  - wpa\_supplicant, 506
- polimorfizm systemu plików, 802
- połączenia przewodowe, 511
- połączenie TCP, 719
- pomiary, 1124, 1138
  - zgodności ze standardami, 1155
- porty, 218, 425
- porządkowanie systemu plików, 161
- POSIX
  - dziedziczenie wpisów ACL, 189
  - określanie dostępu, 189
- Postfix, 697
  - architektura serwera, 697
  - bezpieczeństwo, 699
  - diagnostyka, 709

- Postfix
  - domeny wirtualne, 704
  - dostarczanie lokalne, 703
  - kolejki, 709
  - konfiguracja serwera, 700
  - kontrola dostępu, 706
  - null client, 700
  - odbieranie poczty, 698
  - odbijanie wiadomości, 710
  - tablice dostępu, 707
  - tablice przeglądowe, 702
  - wysyłanie poczty, 699
  - zarządzanie kolejkami wiadomości, 698
- potok, 231
  - CI/CD, 990
  - demonstracyjny, 999, 1005, 1010
  - jako kod, 997
  - Jenkinsa, 1001
  - nazwany, 175
  - poleceń, 237
- poufność, 1015
- Power over Ethernet, 504
- PowerShell, 226
- powiadomienia, 1072
- powiązania, 876, 900
  - stanów ze sługami, 919
- powłoka, 223, 230, 279
  - logowania, 290
  - sh, 238
- poziomy
  - niezawodności centrów danych, 1129
  - RAID, 791
  - uruchomieniowe, 92
- PPID, 135
- prawa własności, 109, 300
- prefiksy IPv6, 435
- preprocesor m4, 655
- priorytety zadań, 1155
- problemy z siecią, 464
- procedury, 1152
  - ponownego uruchamiania, 101
  - uruchomieniowe, 74
- proces, 133
  - budowania, 990, 997
  - cykl życia, 137
  - dopasowywania, 250
  - identyfikator procesu macierzystego, 135
  - init, 84, 85
    - implementacje, 85
    - tryby pracy systemu, 85
    - zadania, 84
  - interaktywne monitorowanie, 144
  - lokalizowania, 219
  - monitorowanie, 142
  - niekontrolowany, 150
  - numer identyfikacyjny, 134
  - okresowy, 152
  - priorytet przełączania, 136
  - rozruchowy, 74
  - stany, 141
  - terminal sterujący, 136
  - uprzejmość, 146
  - żądania przerwania, 137
- profilowanie systemu, 1114
- program
  - Cacti, 475
  - Cobbler, 203
  - cron, 156
  - doc, 612
  - DNSSEC, 601
  - ELK, 362
  - fio, 1112
  - gcloud, 328
  - GRUB, 79
  - iPerf, 474
  - IPFilter, 481
  - iptables, 477
  - Kickstart, 198
  - ldns, 602
  - logrotate, 360
  - newsyslog, 362
  - OpenDNSSEC, 603
  - qmgr, 699
  - RIPE, 603
  - rsync, 631
  - sar, 1113
  - syslog, 343, 344
  - tcpdump, 470–472
  - Terraform, 487
  - TShark, 472
  - Wireshark, 470, 472
  - yum, 215
- programowa strona pamięci masowej, 777
- programowalna sieć komputerowa, 509
- programowanie
  - w języku Python, 255
  - w języku Ruby, 262
- programy
  - powłoki, 1028
  - rozruchowe, 78
- projektowanie sieci, 512
- Prometheus, 1077

- protokoły
  - stanu łączy, 523
  - transportowe, 832
  - wektora odległości, 522
  - wewnętrzne i zewnętrzne, 524
  - wyznaczania tras, 522
- protokół
  - ARP, 418, 440
  - BGP, 526
  - bramy brzegowej, 526
  - DHCP, 441
  - EIGRP, 526
  - HTTP, 714
  - ICMP, 418
  - informowania o trasach, 524
  - IP, 418
  - IPv4, 419
  - IPv6, 419
  - ND, 440
  - NFS, 830
  - OSPF, 526
  - przesyłania danych, 1080
  - RIP, 524
  - SMTP, 640
  - SNMP, 1091
  - TCP, 356, 418
  - trasowania bramy wewnętrznej, 526
  - UDP, 418
- prywatna przestrzeń adresowa, 484
- przechwytywanie, 253
  - pakietów IP, 444
- przekierowania, 231
  - ICMP, 439, 445
- przekierowywanie portów, 1054
- przełączniki, 502
  - funkcji, 989
  - z obsługą VLAN, 503
- przepływ sterowania, 245
- przestrzenie nazw, 125, 951
  - DNS, 539
- przesyłanie
  - pakietów, 518
  - plików, 1057
  - sygnałów, 496
- przewodniki, 59
- przydzielanie
  - adresów, 431
  - zgłoszeń, 1140, 1143
- przypisywanie
  - adresu IP, 448
  - nazwy komputera, 448
- przywracanie
  - dysku, 794
  - systemu, 1147
    - ocena ryzyka, 1147
    - plan naprawy, 1148
    - zwalczanie skutków katastrof, 1149
- ps, 142
- pseudolosowość, 1041
- PTR, pointer, wskaźnik, 557
- pula pamięci masowej, 817
- punkty dostępu, 506
- Puppet, 886
- PXE, 198
- Python, 255, 256, 280, 730
  - instalowanie pakietów, 269
  - krotki, 258
  - liczby, 258
  - listy, 258
  - łańcuchy, 258
  - obiekty, 258
  - pętle, 261
  - pliki, 258
  - słowniki, 258
  - środowiska odtwarzalne, 270
  - wirtualne środowiska, 271
- Python 2, 256
- Python 3, 255

**Q**

- qmgr, 699
- Quagga, 529

**R**

- RAID, 790
  - 5, 794
  - poziomy, 791
  - programowy, 790, 795
  - sprzętowy, 790
- RainerScript, 353
- ramki Jumbo, 504
- ramkowanie, 422
- RBAC, 300
- Red Hat
  - katalogi, 97
  - pliki, 97
- Red Hat Network, 210
- regulacje prawne, 1159

- reguly, 375, 476, 1133, 1151
  - COBIT, 1156
  - grupy zabezpieczeń, 487
  - iptables, 477
  - ustalania priorytetów zadań, 1155
- rejestracja
  - klientów, 878
  - nazwy domeny, 540
  - zgłoszeń, 1139
- rejestrator systemd, 340
- rejestratory Dockera, 972
- rejestrowanie
  - zapytań, 607
  - zdarzeń, 98, 335, 363, 742, 972
- rejestry internetowe, 431
- rekordy
  - A, 556
  - AAAA, 556
  - CNAME, 559
  - DKIM, 562
  - DMARC, 562
  - DNSSEC, 562
  - DS, 594
  - infrastruktury strefy, 551
  - MX, 558
  - NS, 555
  - opcjonalne, 552
  - PTR, 557
  - SOA, 553
  - SPF, 562
  - SRV, 560
  - trasowania, 551
  - TXT, 561
  - zabezpieczeń, 552
  - zasobów, 550
  - zasobów DNSSEC, 594
  - zwykłe, 551
- replikowane systemy plików, 853
- repozytoria
  - lokalne, 213
  - paczek, 877
  - z pakietami, 209
- repozytorium, 212
  - Git, 276
  - LDAP, 307
  - Subversion, 216
- resolver, 589
- ręczne konfigurowanie sieci, 455
- RFC, Requests for Comments, 417
- RIP, Routing Information Protocol, 522, 524, 529
- RIPE, 603
- RIPng, 524
- rncd, 577, 610
- robaki, 1021
- rodzaje
  - adresów, 426
  - danych, 1071
  - dysków, 762
- role, 300, 902
- root, 110, 837, 1028
  - logowanie, 111
  - wyłączanie konta, 120
  - zarządzanie kontem użytkownika, 111
- rootkity, 1022
- rotacja plików dziennika, 161
- routed, 529
- router, 503
  - accept, 691
  - dnslookup, 691
  - manualroute, 692
  - redirect, 692
- routery CISCO, 530
- routing, *Patrz* trasowanie
- rozgłaszanie, 445
- rozproszona odmowa usługi, 1017
- rozruch, 389
  - sieciowy, 203
  - systemu, 73
    - problemy, 102
- rozszerzalność, 927
- RPM, 206
- rsync, 631
- rsyslog
  - architektura systemu, 345
  - konfiguracja systemu, 346, 355
  - moduły, 347
  - opcje konfiguracyjne, 353
  - plik konfiguracyjny, 355
  - składnia, 349
  - wersje systemu, 346
  - właściwości komunikatów, 354
- Ruby, 262, 263, 281, 729
  - bloki, 265
  - hasze opcji, 266
  - instalacja, 263
  - instalowanie pakietów, 269
  - jako filtr, 268
  - symbole, 266
  - środowiska odtwarzalne, 270
  - środowiska wirtualne, 272
  - wrażenia regularne, 267
- RVM, Ruby enVironment Manager, 272



**S**

- SaaS, Software-as-a-Service, 317
- Salt, 913, 926
  - dokumentacja, 925
  - dopasowywanie sług, 910
  - formuły, 921
  - powiązania wartości zmiennych, 909
  - stany, 912
  - środowiska, 921
  - typy dopasowań sług, 911
  - ustawianie usługi, 908
  - wysokie stany, 920
- Samba, 858
  - instalacja i konfigurowanie serwera, 859
- sar, 1113
- scenariusze, 900
- SDN, Software Defined Network, 495, 509
- Security-enhanced Linux, 128
- Sender ID, 644
- sendmail, 651, 652, 653
  - bazy danych, 658
  - bezpieczeństwo, 670
  - bezpieczniejsze przesyłanie wiadomości, 672
  - funkcje ogólnego zastosowania, 659
  - konfiguracja serwera, 655
  - makra, 659
  - mechanizmy antyspamowe, 667
  - monitorowanie kolejki, 677
  - ograniczenia, 669
  - opcje konfiguracyjne, 666
  - opcje prywatności, 674
  - pliki dziennika, 677
  - tabele, 658
  - testowanie i diagnostyka, 676
  - uprawnienia katalogów, 672
  - uruchamianie serwera, 675
  - własność plików, 671
- Sensu, 1076
- serwer
  - aplikacji, 721
  - automatyzacji typu open source, 995
  - autorytatywny, 542, 570
  - buforujący, 542
  - CUPS, 402, 406
  - Exim, 678
  - główny strefy, 574
  - Jenkins, 995
  - Kickstart, 200
  - lustrzany, 213
  - nazw, 545
  - NFS, 838
  - NGINX, 743
  - nierekurencyjny, 542
  - OpenSSH, 1055
  - pamięci podręcznej, 721
  - podległy strefy, 575
  - Postfix, 697
  - rejestrujący, 357
  - rekurencyjny, 542, 570
  - sendmail, 651, 652, 653
  - skoku, 321
  - SMB, 858
  - transportowy, 678
  - WWW, 721, 736
  - wirtualny, 320, 324, 933
  - WWW, 722
  - wydruków, 407
- setgid, 111
- setuid, 111
- sieci
  - bezwzrostowe, 498, 505
  - bezpieczeństwo, 509
  - drogie, 508
  - Infrastruktura bezwzrostowa, 506
  - klient bezwzrostowy, 506
  - punkty dostępu, 506
  - tanie, 508
  - topologia, 507
- CDN, 728
- diagnostyka, 510
- dokumentacja, 514
- dostarczania treści, 728
- Ethernet, 422, 496
- grupy zabezpieczeń, 486
- konfiguracja, 453, 456, 463
- konserwacja, 514
- listy NACL, 486
- mobilne, 454
- monitoring, 473
- mostowe, 961
- podstawowa konfiguracja, 448
- programowalne, 509
- projektowanie, 512
- przeciążenie, 513
- rozbudowa, 513
- rozwiązywanie problemów, 464
- SAN, 472
- testowanie, 510
- TCP/IP, 415
- tworzenie wykresów, 475
- w chmurze, 484

## sieci

- w DigitalOcean, 492
- w GCP, 490
- w systemach Debian i Ubuntu, 456
- w systemach Red Hat i CentOS, 456
- w systemie FreeBSD, 461, 463
- w systemie LINUX, 454
- wirtualne, 321
- wydajność, 474
- zapory sieciowe, 476
- zarządzanie, 514

## sieciowe

- listy kontroli dostępu, 484
- systemy plików, 828
- uwierzelnianie kryptograficzne, 124

## sieciowy klient rejestrujący, 356

## silnik kontenerowy, 953

## skalowalność procesu wdrażania, 926

## skaner portów sieciowych, 1028, 1030

## skrętka, 511

- nieekranowana, 498

## skrypty, 83, 223, 228, 240

- do dodawania użytkowników, 301

- startowe, 99, 359

- w powłoce SH, 238

## skrzynki pocztowe, 637

## SLA, service level agreements, 1153

## SLAAC, 436

## SLC, single-level cells, 764

## sługi, 908

## SMART, 776

## SMB, Server Message Block, 857

- bezpieczeństwo, 865
- konfigurowanie rejestrowania zdarzeń, 866
- montowanie plików, 863
- sprawdzanie stanu, 865

## smbstatus, 865

## smokeping, 473

## SMTP, 640

- kody błędów, 641
- uwierzelnianie, 642

## SMTP AUTH, 689

## snapshot, 787

## SNMP, 1091

- Net-SNMP, 1093
- operacje protokołu, 1093
- organizacja, 1092

## Snort, 1033

## SOA, Start of Authority, 553

## socjotechnika, 1015

## sort, 235

## sortowanie wierszy, 235

## spam, 643

## SPF, 644

## splątanie, 1041

- sprawdzanie plików, 177
- poprawności wejścia, 260

## sprzęt

- elektroniczny, 1125
- sieciowy, 458, 462, 495

## SSD, Solid State Drives, 755

## sshd, 1055

## SSHFP, 1056

- SSO, single sign-on, 307, 615
- komponenty, 616, 617

## sssd, 626

## standard

- CJIS, 1156
- EIA-606, 512
- ISO/IEC 27001, 1063
- TIA/EIA-568A, 500

## standardy

- bezpieczeństwa, 1062
- bezzprzewodowe, 505
- okablowania, 511
- sieciowe, 417

## standaryzacja dokumentacji, 1144

## stany

- procesów, 141
- wątków, 141
- wysokie, 920

## StatsD, 1080

## statusy jednostek, 89

## statystyki

- polecenia ping, 473
- wydajności, 1113

## sterowanie serwerem nazw, 610

## sterowniki, 367

- pamięci masowej, 964
- parawirtualizowane, 936
- urządzeń, 370

## strefa localhost, 580

## struktura

- transakcji HTTP, 716
- zegarów systemd, 157

## strukturyzacja aktualizacji, 220

## su, 112

## sudo, 113

- hasła, 117
- konfiguracja, 119
- terminal sterujący, 119

- ustawienia, 116
    - zaawansowana kontrola dostępu, 116
    - zalety i wady, 115
  - superużytkownik, 110
  - Supervisor, 1088
  - surowe woluminy, 816
  - switch, 502
  - sygnał, 137, 139
    - HUP, 140
    - INT, 139
    - KILL, 139
    - QUIT, 140
    - TERM, 139
  - sysdig, 1086
  - sysfs, 374
  - syslog, 343, 344
    - bezpieczeństwo komunikatów, 357
    - diagnostyka konfiguracji, 359
    - komunikaty systemu, 344
    - popularne działania, 351
    - poziomy ważności komunikatów, 350
    - usługi, 350
  - system
    - Ansible, 887, 889, 926
    - BIND, 604
    - Chef, 886
    - chmurowy, 68, 104
    - CI/CD, 985, 988
    - Docker, 953
    - dostarczania lokalnego, 636
    - dostępowy, 637
    - drukowania, 401
    - jednozadaniowy, 1166
    - Kerberos, 1046
    - kontroli wersji, 274
    - nazw domenowych, 535
    - plików, 163, 779, 788, 799, 812
      - /PROC, 147
      - Btrfs, 808, 819
      - dla każdego użytkownika, 815
      - dziedziczenie właściwości, 814
      - EXT4, 800
      - formatowanie, 802
      - katalogi, 170
      - montowanie, 166, 804
      - montowanie automatyczne, 804
      - naprawa, 802
      - NFS, 141, 827
      - odmontowywanie, 166
      - podwoluminy, 822
      - polimorfizm, 802
      - porządkowanie, 161
      - replikowany, 853
      - sieciowy, 828
      - ścieżki dostępu, 165
      - UFS, 800
      - warstwy, 959
      - właściwości, 812
      - woluminy, 822
      - wydajność, 809
      - wykrywanie błędów, 809
      - XFS, 800
      - ZFS, 808, 810
    - pojedynczego logowania, 307, 615
    - przyjmujący, 635
    - Puppet, 886
    - reagowania na ataki brute-force, 1036
    - rozruchowy
      - FreeBSD, 81
      - PXE, 198
    - Salt, 887, 906
    - transportowy, 636
    - uwierzytelniania, 623
    - wykrywania włamań, 1033
    - zarządzania
      - konfiguracją, 879
      - pakietami, 206
      - tożsamością, 308
    - zgłoszeniowy, 1139, 1142
  - systemctl, 88, 97
    - podporadki, 89
  - systemd, 86, 92, 96
    - konfiguracja rejestratora, 341
    - opcje filtrujące, 342
    - program syslog, 343
    - przeglądanie dzienników, 340
    - rejestrowanie zdarzeń, 98
    - struktura zegarów, 157
    - wyrażenia czasowe, 159
  - szafy, 1120
  - szczupłe zarządzanie, 1136
  - szperacze sieciowe, 510
  - szyfrowanie, 676, 708
- ## Ś
- ścieżka
    - BIOS-u, 81
    - dostępu, 165
    - UEFI, 82

śledzenie  
 funkcji systemowych, 148  
 pakietów IP, 467  
 sygnałów, 148  
 wydajności sieci, 474  
 środowisko, 877  
 chroot, 590, 675  
 produkcyjne, 988  
 przedprodukcyjne, 988  
 rozwojowe, 988  
 światłowody jednomodowe, 500

## T

tablica, 476  
 partycji GUID, 783  
 przeglądowa, 702  
 tras, 438, 485  
 tras VPC, 486  
 tail, 237  
 TCP, 418, 719  
 TCP/IP, 415, 459  
 tcpdump, 470, 471, 472  
 technicy centrów danych, 70  
 techniki wdrażania, 994  
 technologia Advanced Format, 767  
 tee, 237  
 telefonia VoIP, 504  
 teoria wielkiej unifikacji, 1134  
 terminal sterujący, 136  
 Terraform, 487  
 test penetracyjny, 1031  
 testowanie, 221, 991  
 kropli, 1008  
 serwera sendmail, 676  
 sieci, 510  
 urzędzeń, 374  
 wydajności, 1112  
 testy  
 akceptacyjne, 992  
 infrastruktury, 992  
 integracyjne, 992  
 jednostkowe, 992, 1000  
 penetracyjne aplikacji, 1024  
 wydajnościowe, 992  
 TKEY, 590  
 TLS, Transport Layer Security, 676, 740,  
 973, 1040  
 diagnostyka serwerów, 1044  
 top, 145  
 topologia  
 Ethernetu, 498  
 sieci bezprzewodowych, 507  
 tożsamość, 322  
 traceroute, 467  
 transfer plików, 631  
 transport, 693  
 appendfile, 694  
 smtp, 694  
 trasowanie, 517  
 BGP, 526  
 demony trasujące, 528  
 dokumenty RFC, 533  
 EIGRP, 526  
 grupowe adresy protokołów, 527  
 kryteria wyboru strategii, 527  
 miary kosztu, 523  
 OSPF, 526  
 protokoły  
 stanu łączy, 523  
 wektora odległości, 522  
 wewnętrzne i zewnętrzne, 524  
 przekierowania ICMP, 439  
 przez nadawcę, 445  
 RIP, 524  
 tablice tras, 438  
 trendy historyczne, 1071  
 treść wiadomości, 717  
 tryb pojedynczego użytkownika, 102, 104  
 z programem GRUB, 104  
 tryby pracy systemu, 85  
 TShark, 472  
 TSIG, 590  
 tunelowanie  
 IPsec, 1060  
 IPv6, 437  
 tworzenie  
 chmury VPC, 487  
 instancji EC2, 325  
 jądra, 383, 386  
 katalogu domowego, 298  
 kopii zapasowych, 161, 824  
 macierzy, 796  
 odtwarzalnych środowisk, 270  
 plików binarnych jądra, 384  
 plików urzędzeń, 372  
 reguł, 1152  
 skryptów, 279  
 własnych poddomen, 540  
 wykresów, 1078  
 typy  
 artefaktów, 991  
 dopasowań sług, 911  
 hipernadzorców, 936  
 plików, 171

**U**

udev  
 klucze dopasowań, 379  
 udevadm, 374  
 udevd  
 klucze dopasowań, 376  
 udostępnianie katalogów  
 dla projektów, 862  
 domowych, 862  
 UDP, 418  
 UEFI, 76  
 UID, 135, 284, 288  
 umask, 181, 182  
 uniq, 236  
 UNIX, 1167, 1171, 1174  
 kontrola dostępu, 108  
 uprawnienia, 179, 191, 300  
 administracyjne, 107, 300  
 domyślne, 181  
 uprzejmość, 146  
 URL, Uniform Resource Locator, 715  
 uruchamianie  
 aplikacji WWW, 741  
 zadań wsadowych, 161  
 urządzenia, 370  
 pamięci masowej, 759, 778  
 trwale nazwy, 375  
 USB, 770  
 montowanie napędu, 807  
 useradd, 301, 302, 303  
 usługi  
 chmurowe, 312, 317  
 katalogowe, 617, 623  
 lokalne, 95  
 opisy, 1154  
 systemowe, 87  
 zakresy, 1154  
 ustawienia pliku konfiguracyjnego, 198  
 usunięcie  
 drukarki, 408  
 konta, 304  
 UTP, unshielded twisted pair, 498  
 utrzymanie  
 lokalnej dokumentacji, 1143  
 niezależnych środowisk, 1146  
 uwierzytelnianie, 119, 306, 623, 627, 676,  
 689, 1050  
 HTTP, 740  
 klientów, 708  
 kryptograficzne, 124

lokalne, 860  
 SMTP, 642  
 wieloskładnikowe, 1023  
 użytkownik, 284  
 blokowanie konta, 305  
 dodawanie, 301, 304  
 finalizacja konta, 300  
 identyfikator UID, 288  
 katalog domowy, 290, 298  
 nazwa, 285  
 nowe konto, 296  
 pole GECOS, 289  
 powłoka logowania, 290  
 prawa własności, 300  
 root, 1028  
 scentralizowane zarządzanie kontem, 307  
 uprawnienia, 300  
 ustawianie hasła, 297

**V**

Vagrant, 946  
 VirtualBox, 944  
 virtualenv, 271  
 VLAN, Virtual Local Area Networks, 503  
 VMware, 943  
 VoIP, 504  
 VPC  
 VPC, virtual private cloud, 484–487  
 VPN, Virtual Private Network, 447, 1060

**W**

wady modelu standardowego, 122  
 walidacja stanu systemu, 1089  
 WAP, wireless access points, 505  
 wątki, 141  
 wbudowane moduły, 927  
 wc, 237  
 wdrażanie, 993, 1008  
 bez przestojów, 994  
 wejście, 242  
 wektor odległości, 522  
 WEP, Wired Equivalent Privacy, 509  
 wersje jądra, 369  
 wersjonowanie, 274  
 weryfikacja  
 integralności systemu, 1089  
 klucza hosta, 1056  
 wiadomości e-mail, 160, 633, 637  
 widoczność zasobów, 852

- widoki, 582
  - wiersz poleceń, 243
    - procesu, 148
  - wilgotność, 1128
  - Windows, 1172
  - Wireshark, 470, 472
  - wirtualizacja, 933
    - hipernadzorczy, 934
    - konteneryzacja, 938
    - KVM, 942
    - maszyny wirtualne, 937
    - Packer, 944
    - sprzętowa, 935
    - sterowniki parawirtualizowane, 936
    - typy hipernadzorców, 936
    - Vagrant, 946
    - VirtualBox, 944
    - VMware, 943
    - w Linuksie, 939
    - Xen, 939
  - wirtualne
    - chmury prywatna, 484
    - domeny aliasów, 704
    - hosty, 720
    - serwery prywatne, 320, 324
    - sieci, 321
    - sieci prywatne, 447
    - środowiska, 271
  - wirusy, 1021
  - witryny informacyjne, 61
  - włamanie, 1032, 1033, 1090
  - włókna światłowodowe, 500
  - woluminy, 822, 959
    - logiczne, 778, 784
  - wskaźniki
    - awaryjności, 761
    - czasu rzeczywistego, 1071
  - współdzielenie plików, 860, 861
  - wybór dostawcy chmury, 314
  - wycofania, 254
  - wydajność, 1097
    - analiza
      - obciążenia wejścia-wyjścia, 1111
      - problemów, 1102
      - użycia pamięci, 1109
      - użycia procesora, 1106
    - czynniki, 1101
    - dostrajanie, 1098
    - energetyczna, 1123
    - gromadzenie danych, 1105
    - kontrola, 1103
    - metody poprawy, 1099
    - operacje wejścia-wyjścia, 1113
    - podsystemu dyskowego, 1112
    - profilowanie systemu, 1114
    - statystyki, 1113
    - zabieranie cykli procesora, 1102
    - zarządzanie pamięcią, 1108
  - wyjście, 242
  - wykonywanie poleceń, 239
  - wykrywanie włamań, 1033, 1090
    - sieciowych, 1032
  - wyliczanie podsieci, 429
  - wyłączanie
    - fizycznych systemów, 101
    - konta użytkownika root, 120
    - systemów chmurowych, 101
  - wymazywanie wstępne, 765
  - wymuszanie stosowania reguł, 1160
  - wyrażenia
    - czasowe systemd, 159
    - regularne, 249, 252, 280
      - w języku Ruby, 267
  - wyrażenie \$, 1007
  - wysyłanie
    - sygnałów, 140
    - wiadomości
      - do plików, 648
      - do programów, 649
      - e-mail, 160
  - wyszukiwanie
    - pakietów, 269
    - tekstu, 238
  - wyświetlanie listy, 177
  - wyznaczanie tras, *Patrz* trasowanie
- X**
- Xen, 939
  - XML, 731
  - XORP, 530
- Y**
- YAML, 887
  - yum, 215
- Z**
- zachłanność, 254
  - zadania wsadowe, 161
  - zamykanie systemu, 101

- zapis adresów IPv6, 434
- zapisywanie dzienników, 695
- zaplanowane zadania, 160
- zapora
  - aplikacji, 721
  - IPFilter, 481
  - iptables, 478
- zapory
  - filtrujące pakiety, 1058
  - sieciowe, 446, 476, 1058
    - dla instancji EC2, 484
    - z kontrolą stanu, 1059
- zapytania LDAP, 621
- zarządzania
  - zgłoszeniami, 1139
  - bibliotekami, 269
  - dziennikami, 360, 362, 363
    - na dużą skalę, 362
  - konfiguracją, 873
    - ewidencjonowanie, 878
    - obsługa zmian, 876
    - opcje architekuralne, 880
    - opcje językowe, 883
    - opcje zarządzania zależnościami, 884
    - operacje i parametry, 874
    - paczki, 877
    - powiązania, 876
    - rejestracja klientów, 878
    - repozytoria paczek, 877
    - system Ansible, 887
    - system Chef, 886
    - system Puppet, 886
    - system Salt, 887
    - systemy, 879
    - środowiska, 877
      - wzorce postępowania, 929
    - zmiennie, 875
  - kontami, 307
  - kontenerami, 976
  - oprogramowaniem, 215
  - pakietami, 204, 208
    - .deb, 207
    - RPM, 206
  - pamięcią, 1108
  - plikami
    - crontab, 155
    - urządzeń, 372, 373
  - pulą pamięci masowej, 817
  - systemd, 88
  - systemem, 83
  - środowiskiem, 117
  - tożsamością, 308
  - urządzeniami, 373, 378
    - wysokopoziomowe, 379
  - uszkodzonymi blokami, 773
  - użytkownikami, 283
  - woluminami logicznymi, 784, 790
  - zestawami znaków, 867
  - zasady DevOps, 1135
  - zasilacze awaryjne, 1121
  - zasilanie
    - nadmiarowe, 1121
    - szaf, 1122
  - zbędne usługi, 1020
  - zdalna powłoka SSH, 1046
  - zdalne
    - logowanie zdarzeń, 1021
    - sterowanie, 1124
    - systemy plików, 846
    - wywoływanie procedur, 832
  - zdarzenia, 335, 363, 1071
    - rejestrowane w dziennikach, 364
  - zegary
    - przejściowe, 160
    - systemd, 156
  - ZFS, 810
  - zgłoszenia, 1139
  - zgodność
    - regulacje i standardy, 1156
    - ze standardami, 1155
  - zliczanie wierszy, 237
  - zmiana
    - hasła, 1025
    - systemu plików, 788
    - tożsamości użytkownika, 112
    - uprawnień, 179
    - właściciela i grupy, 181
  - zmiennie, 233, 875
    - jądra, 461
    - środowiskowe, 234, 1007
  - znaczniki
    - SEP, 596
    - kontrolne PAM, 629
  - znaki
    - dosłowne, 250
    - specjalne, 250
  - zrzut pamięci, 138

## Ż, Ż

- źródła informacji, 60
- żądania HTTP, 716

## NOTATKI



# PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



- 1. ZAREJESTRUJ SIĘ**
- 2. PREZENTUJ KSIĄŻKI**
- 3. ZBIERAJ PROWIZJĘ**

Zmień swoją stronę WWW  
w działający bankomat!

**Dowiedz się więcej i dołącz już dzisiaj!**

<http://program-partnerski.helion.pl>

## Unix i Linux: niezawodność, wydajność i elastyczność na najwyższym poziomie!

Mijają lata, a systemy Unix i Linux ciągle są gwarancją najwyższej niezawodności, wydajności i elastyczności. Ich potencjał jest niekwestionowany, a odporność na niestandardowe warunki zapewnia bezawaryjną pracę w różnych sytuacjach. To wszystko sprawia, że systemy te są wykorzystywane w dużych i złożonych organizacjach. Ich sieci wymagają profesjonalnego administrowania, co jest trudną i odpowiedzialną pracą. Łatwo skonfigurować pojedynczy system, znacznie trudniej jednak zapewnić stabilne działanie rozproszonej, opartej na chmurze platformy, narażonej na skokowe wzrosty popularności, rozbicia sieci i celowe ataki.

Ta książka jest kolejnym, uzupełnionym i zaktualizowanym wydaniem kultowego podręcznika dla profesjonalnych administratorów zarządzających środowiskami produkcyjnymi w korporacjach czy urzędach. Znajdziesz tu obszernie wyjaśnienie takich zagadnień jak instalacja systemu, skrypty powłoki, kontrolowanie procesów czy konfiguracja uprawnień. Dowiesz się, jak zarządzać użytkownikami, przestrzenią dyskową, zadaniami okresowymi oraz backupami. Przystudiujesz zagadnienia sieciowe, a zwłaszcza kwestie bezpieczeństwa i reakcji na incydenty. W tym wydaniu znajdziesz m.in. omówienie demonów zarządzania systemem, zasady zarządzania kontem użytkownika root, techniki kontroli wersji za pomocą narzędzia Git, kwestie związane z zarządzaniem sieciami w chmurze, tworzeniem i utrzymywaniem centrów danych, opis metodologii DevOps i wiele innych!

### Najciekawsze zagadnienia:

- podstawy administrowania i obowiązki administratora
- system plików i zarządzanie oprogramowaniem
- rejestrowanie zdarzeń
- zarządzanie jądrem systemu i sterownikami
- sieci i sprzęt sieciowy
- zapewnianie wydajności i niezawodności sieci
- metodologie i reguły w IT

**Evi Nemeth** była matematykiem, kryptografem i wiodącą współautorką książek uznawanych za biblie administratorów systemów. Była również autorytetem w dziedzinie sieci. W 2013 roku zaginęła bez wieści, gdy żeglowała po Morzu Tasmana.

**Garth Snyder** jest inżynierem. Pracował w firmach NeXT i Sun. **Trent R. Hein** jest pasjonatem bezpieczeństwa informatycznego i automatyzacji. Lubi piesze wędrówki, narty, muzykę bluegrass, psy i gramatykę.

**Ben Whaley** jest założycielem WhaleTech, niezależnej firmy doradczej. Został uhonorowany przez firmę Amazon jako jeden z pierwszych bohaterów społeczności AWS.

**Dan Mackin** jest zadeklarowanym użytkownikiem Linuksa i innych technologii open source. Uwielbia jeździć na nartach, żeglować i spędzać czas z żoną i psem.

**Helion**

helion.pl

HELION SA  
ul. Kosciuszki 1c  
44-100 Gliwice  
tel.: 32 230 98 63  
helion@helion.pl

Sprawdź nasze szkolenia!

SZKOLENIA



AKADEMIA IT & BUSINESS

WWW.SZKOLENIA.HELION.PL

KOD KORZYŚCI  
Sięgnij po więcej! ▶



ISBN 978-83-283-4175-3



9 788328 341753



Pearson  
Addison-Wesley

INFORMATYKA W NAJLEPSZYM WYDANIU

Cena: 179,00 zł