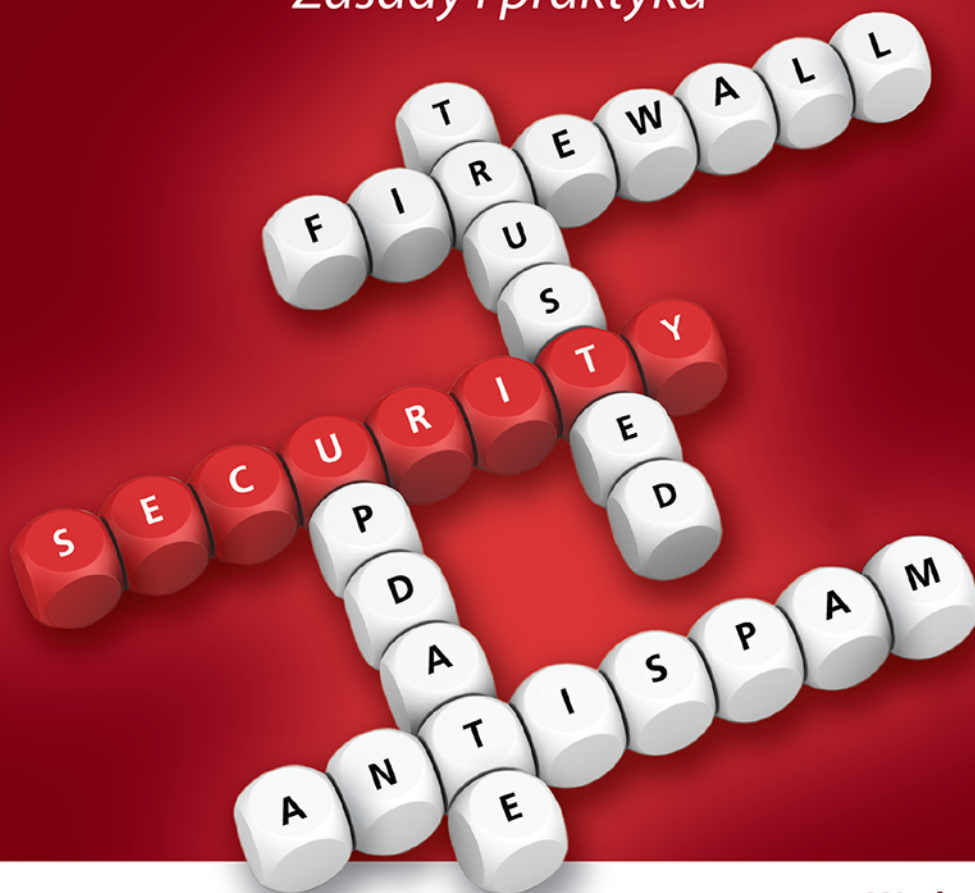


William Stallings • Lawrie Brown

BEZPIECZEŃSTWO SYSTEMÓW INFORMATYCZNYCH

Zasady i praktyka



TOM
1

Wydanie IV

Tytuł oryginału: Computer Security: Principles and Practice (4th Edition)

Tłumaczenie: Zdzisław Płoski, Radosław Meryk (słowniczek)

ISBN: 978-83-8322-550-0

Authorized translation from the English language edition, entitled: COMPUTER SECURITY: PRINCIPLES AND PRACTICE, Fourth Edition; ISBN 0134794109; by William Stallings, and by Lawrie Brown, published by Pearson Education, Inc. Copyright © 2018, 2015, 2012, 2008 by 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 © 2019, 2023.

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 wydawca 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 wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<https://helion.pl/user/opinie/bsi41v>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Helion S.A.

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 230 98 63

e-mail: helion@helion.pl

WWW: <https://helion.pl> (księgarnia internetowa, katalog książek)

Printed in Poland.

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

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

SPIS TREŚCI

Przedmowa	9
Notacja	21
O autorach	23
Rozdział 1. Przegląd	25
1.1. Koncepcje bezpieczeństwa komputerowego	26
1.2. Zagrożenia, ataki i aktywa	35
1.3. Funkcjonalne wymagania bezpieczeństwa	42
1.4. Podstawowe zasady projektowania bezpieczeństwa	44
1.5. Powierzchnie ataków i drzewa ataków	49
1.6. Strategia bezpieczeństwa komputerowego	53
1.7. Standardy	56
1.8. Podstawowe pojęcia, pytania sprawdzające i zadania	57
 CZĘŚĆ I TECHNIKI I ZASADY BEZPIECZEŃSTWA KOMPUTEROWEGO	
Rozdział 2. Narzędzia kryptograficzne	61
2.1. Osiąganie poufności za pomocą szyfrowania symetrycznego	62
2.2. Uwierzytelnianie komunikatów i funkcje haszowania	69
2.3. Szyfrowanie z kluczem publicznym	79
2.4. Podpisy cyfrowe i zarządzanie kluczami	85
2.5. Liczby losowe i pseudolosowe	90
2.6. Zastosowanie praktyczne: szyfrowanie przechowywanych danych	93
2.7. Podstawowe pojęcia, pytania sprawdzające i zadania	95
Rozdział 3. Uwierzytelnianie użytkownika	101
3.1. Zasady cyfrowego uwierzytelniania użytkownika	103
3.2. Uwierzytelnianie oparte na hasłach	109
3.3. Uwierzytelnianie oparte na żetonach	124
3.4. Uwierzytelnianie biometryczne	129
3.5. Zdalne uwierzytelnianie użytkownika	135
3.6. Zagadnienia bezpieczeństwa uwierzytelniania użytkownika	140
3.7. Zastosowanie praktyczne: tęczówkowy system biometryczny	142
3.8. Przykład konkretny: problemy bezpieczeństwa w systemach bankomatowych	144
3.9. Podstawowe pojęcia, pytania sprawdzające i zadania	147

Rozdział 4. Kontrolowanie dostępu	151
4.1. Zasady kontrolowania dostępu	154
4.2. Podmioty, obiekty i prawa dostępu	156
4.3. Uznaniove kontrolowanie dostępu	158
4.4. Przykład: kontrolowanie dostępu w uniksowym systemie plików	165
4.5. Kontrolowanie dostępu według ról	169
4.6. Kontrolowanie dostępu według atrybutów	176
4.7. Tożsamość, poświadczenia i zarządzanie dostępem	183
4.8. Ramy zaufania	187
4.9. Przykład konkretny: kontrolowanie ról w systemie bankowym	192
4.10. Podstawowe pojęcia, pytania sprawdzające i zadania	195
Rozdział 5. Bezpieczeństwo baz i centrów danych	201
5.1. Zapotrzebowanie na bezpieczeństwo baz danych	202
5.2. Systemy zarządzania bazami danych	204
5.3. Relacyjne bazy danych	206
5.4. Ataki wstrzykiwania w sql	210
5.5. Kontrolowanie dostępu do bazy danych	217
5.6. Wnioskowanie	223
5.7. Szyfrowanie baz danych	226
5.8. Bezpieczeństwo centrum danych	231
5.9. Podstawowe pojęcia, pytania sprawdzające i zadania	237
Rozdział 6. Malware — szkodliwe oprogramowanie	243
6.1. Rodzaje szkodliwego oprogramowania	245
6.2. Zaawansowane trwałe zagrożenie	249
6.3. Rozsiewanie — zainfekowana treść — wirusy	250
6.4. Rozsiewanie — wykorzystanie wrażliwych punktów — robaki	257
6.5. Rozsiewanie — socjotechnika — spam pocztowy, konie trojańskie	269
6.6. Ładunek — psucie systemu	272
6.7. Ładunek — agent ataku — zombie, boty	275
6.8. Ładunek — kradzież informacji — keylogery, phishing, spyware	277
6.9. Ładunek — działania ukradkowe — boczne drzwi, rootkity	280
6.10. Przeciwdziałania	285
6.11. Podstawowe pojęcia, pytania sprawdzające i zadania	293
Rozdział 7. Ataki polegające na odmowie świadczenia usług	297
7.1. Odmowa usług jako rodzaj ataku	298
7.2. Ataki zatapiające	307
7.3. Rozproszone ataki blokowania usług	310
7.4. Ataki na przepływność oparte na aplikacjach	312

7.5.	Ataki odbijające i ataki ze wzmocnieniem	315
7.6.	Obrona przed odmową świadczenia usług	321
7.7.	Reagowanie na atak typu odmowa świadczenia usług	325
7.8.	Podstawowe pojęcia, pytania sprawdzające i zadania	327
Rozdział 8.	Wykrywanie włamań	331
8.1.	Intruzi	332
8.2.	Wykrywanie włamań	336
8.3.	Podejścia analityczne	341
8.4.	Wykrywanie włamań oparte na hoście	344
8.5.	Wykrywanie włamań oparte na sieci	351
8.6.	Rozproszone lub hybrydowe wykrywanie włamań	358
8.7.	Format wymiany wykrywania włamań	361
8.8.	Miodownice (honeypoty)	364
8.9.	Przykład systemu: snort	367
8.10.	Podstawowe pojęcia, pytania sprawdzające i zadania	371
Rozdział 9.	Zapory sieciowe i systemy zapobiegania włamaniom	377
9.1.	Zapotrzebowanie na zapory sieciowe	378
9.2.	Charakterystyka zapór sieciowych i polityka dostępu	379
9.3.	Rodzaje zapór sieciowych	381
9.4.	Posadowienie zapór sieciowych	389
9.5.	Umiejscowienie i konfiguracja zapór sieciowych	392
9.6.	Systemy zapobiegania włamaniom	398
9.7.	Przykład: ujednolicone środki opanowywania zagrożeń	404
9.8.	Podstawowe pojęcia, pytania sprawdzające i zadania	409
CZĘŚĆ II	BEZPIECZEŃSTWO OPROGRAMOWANIA I SYSTEMÓW	
Rozdział 10.	Przepełnienie bufora	415
10.1.	Przepełnienia stosu	417
10.2.	Obrona przed przepełnieniami bufora	442
10.3.	Inne formy ataków przepełniających	450
10.4.	Podstawowe pojęcia, pytania sprawdzające i zadania	457
Rozdział 11.	Bezpieczeństwo oprogramowania	461
11.1.	Zagadnienia bezpieczeństwa oprogramowania	463
11.2.	Obsługa wejścia programu	468
11.3.	Pisanie bezpiecznego kodu	482
11.4.	Współpraca z systemem operacyjnym i innymi programami	488
11.5.	Obsługa wyjścia programu	504
11.6.	Podstawowe pojęcia, pytania sprawdzające i zadania	507

Rozdział 12. Bezpieczeństwo systemów operacyjnych	511
12.1. Wprowadzenie do bezpieczeństwa systemów operacyjnych	514
12.2. Planowanie bezpieczeństwa systemu operacyjnego	514
12.3. Hartowanie systemów operacyjnych	515
12.4. Bezpieczeństwo aplikacji	521
12.5. Dbłość o bezpieczeństwo	522
12.6. Bezpieczeństwo w systemach Linux i UNIX	524
12.7. Bezpieczeństwo w systemie Windows	529
12.8. Bezpieczeństwo wirtualizacji	532
12.9. Podstawowe pojęcia, pytania sprawdzające i zadania	542
Rozdział 13. Bezpieczeństwo chmur i internetu rzeczy	545
13.1. Obliczenia w chmurze	546
13.2. Koncepcje bezpieczeństwa chmury	556
13.3. Podejścia do bezpieczeństwa chmury	561
13.4. Internet rzeczy (ir)	570
13.5. Bezpieczeństwo internetu rzeczy	576
13.6. Podstawowe pojęcia i pytania sprawdzające	587
Spis treści tomu 2.	589
Słowniczek	595
Akronimy	605
Literatura	607
Skorowidz	621

KONTROLOWANIE DOSTĘPU

4.1. Zasady kontrolowania dostępu

Kontekst kontrolowania dostępu

Zasady kontrolowania dostępu

4.2. Podmioty, obiekty i prawa dostępu

4.3. Uznaniowe kontrolowanie dostępu

Model kontrolowania dostępu

Domeny ochrony

4.4. Przykład: kontrolowanie dostępu w uniksowym systemie plików

Tradycyjne kontrolowanie dostępu do plików w UNIX-ie

Lista kontroli dostępu w UNIX-ie

4.5. Kontrolowanie dostępu według ról

Modele wzorcowe RBAC

4.6. Kontrolowanie dostępu według atrybutów

Atrybuty

Architektura logiczna ABAC

Polityka ABAC

4.7. Tożsamość, poświadczenia i zarządzanie dostępem

Zarządzanie tożsamością

Zarządzanie poświadczeniami

Zarządzanie dostępem

Federacja tożsamości

4.8. Ramy zaufania

Tradycyjne podejście do wymiany tożsamości

Rama zaufania otwartej tożsamości

4.9. Przykład konkretny: kontrolowanie ról w systemie bankowym

4.10. Podstawowe pojęcia, pytania sprawdzające i zadania

W TYM ROZDZIALE POZNASZ I ZROZUMIESZ:

- ◆ miejsce kontrolowania dostępu w szerszym kontekście uwierzytelniania, upoważniania i doglądania;
- ◆ trzy główne rodzaje polityki kontrolowania dostępu;
- ◆ różnice między podmiotami, obiektami i prawami dostępu;
- ◆ model kontrolowania dostępu do plików w systemie UNIX;
- ◆ podstawowe koncepcje kontrolowania dostępu według ról;
- ◆ podsumowanie modelu RBAC;
- ◆ podstawowe zasady kontrolowania dostępu opartego na atrybutach;
- ◆ model tożsamości, poświadczeń i zarządzania dostępem;
- ◆ pojęcie federacji tożsamości i jego związek z ramą zaufania.

Zrozumienie zakresu pojęcia kontrolowania dostępu ułatwiają dwie definicje.

1. NISTIR 7298 (*Glossary of Key Information Security Terms*, z ang. „Słownik podstawowych terminów z zakresu bezpieczeństwa informacji”) z maja 2013 roku definiuje kontrolowanie dostępu jako proces spełniania lub odmowy spełnienia określonych żądań: (1) uzyskania i użytkowania informacji i usług przetwarzania związanych z informacjami oraz (2) dostępu do określonych urządzeń (rozwiązań) fizycznych.
2. RFC 4949 (*Internet Security Glossary*) definiuje kontrolowanie dostępu jako postępowanie regulujące użycie zasobów systemu zgodnie z zasadami bezpieczeństwa, dozwolone tylko upoważnionym jednostkom (użytkownikom, programom, procesom lub innym systemom) stosownie do przyjętych zasad.

Kontrolowanie dostępu możemy uważać za centralny element bezpieczeństwa komputerowego. Podstawowym celem bezpieczeństwa komputerowego jest zapobieganie uzyskiwaniu przez nielegalnych użytkowników dostępu do zasobów, zapobieganie dostępowi do zasobów przez legalnych użytkowników w sposób nieupoważniony i umożliwianie legalnym użytkownikom dostępu do zasobów zgodnie z ich uprawnieniami. Tabela 4.1, zaczerpnięta z dokumentu NIST SP 800-171 (*Protecting Controlled Unclassified Information in Nonfederal Information Systems and Organizations*) z sierpnia 2016 roku, zawiera pożyteczną listę wymagań bezpieczeństwa dotyczących usług kontrolowania dostępu.

Rozpocniemy ten rozdział od przeglądu kilku ważnych pojęć. Potem przyjrzymy się trzem powszechnie stosowanym technikom implementowania zasad kontrolowania dostępu. Następnie poszerzymy nasz ogłód o ogólne zarządzanie kontrolowaniem dostępu z użyciem oznak tożsamości, poświadczeń i atrybutów. Na koniec wprowadzimy pojęcie ramy zaufania.

Tabela 4.1. Wymagania bezpieczeństwa dotyczące kontrolowania dostępu (SP 800-171)

Podstawowe wymagania bezpieczeństwa
1. Ograniczaj dostęp do systemu informacyjnego do upoważnionych użytkowników, procesów działających w imieniu upoważnionych użytkowników lub urzędzeń (w tym innych systemów informacyjnych)
2. W systemie informacyjnym ograniczaj dostęp do rodzajów transakcji i funkcji, które wolno wykonywać upoważnionym użytkownikom
Pochodne wymagania bezpieczeństwa
3. Kontroluj przepływ CUI zgodnie z przyjętymi upoważnieniami
4. Oddzielaj obowiązki poszczególnych osób, aby zmniejszyć ryzyko nieukartowanych wrogich działań
5. Stosuj zasadę najmniejszych przywilejów, odnoszoną w szczególności do określonych funkcji i uprzywilejowanych kont
6. Używaj nieuprzywilejowanych kont lub ról w dostępie do funkcji nieodnoszących się do bezpieczeństwa
7. Nie dopuszczaj do wykonywania przez nieuprzywilejowanych użytkowników uprzywilejowanych funkcji i doglądaj wykonywania takich funkcji
8. Ograniczaj nieudane próby logowania (rozpoczęcia sesji w systemie)
9. Zadbaj o zgodność adnotacji dotyczących prywatności i bezpieczeństwa ze stosowanymi regułami CUI
10. Stosuj blokowanie sesji z ukrywaniem wyświetlanych wzorców w celu zapobiegania dostępowi i oglądania danych po upływie okresu bezczynności
11. Kończ (automatycznie) sesję użytkownika po wystąpieniu określonego warunku
12. Monitoruj i kontroluj sesje zdalnego dostępu
13. Korzystaj z mechanizmów kryptograficznych do ochrony poufności sesji zdalnego dostępu
14. Prowadź zdalny dostęp przez odpowiednio przygotowane punkty kontrolowania dostępu
15. Upoważniaj zdalne wykonywanie uprzywilejowanych poleceń i zdalny dostęp do informacji wymagających bezpieczeństwa
16. Dopilnuj upoważnienia dostępu bezprzewodowego przed zezwoleniem na takie połączenia
17. Chronić dostęp bezprzewodowy za pomocą uwierzytelniania i szyfrowania
18. Kontroluj połączenia urzędzeń mobilnych
19. Szyfruj CUI w urzędzeniach mobilnych
20. Weryfikuj i kontroluj lub ograniczaj połączenia i korzystanie z zewnętrznych systemów informacyjnych
21. Ograniczaj użycie w zewnętrznych systemach informacyjnych przenośnych urzędzeń pamięciowych należących do danej organizacji (firmy)
22. Kontroluj CUI wysyłane pocztą lub przetwarzane w dostępnych publicznie systemach informacyjnych

CUI (ang. *controlled unclassified information*) = kontrolowana informacja nieklasyfikowana

Źródło: na podstawie raportu NIST SP 800-172 *Protecting Controlled Unclassified Information in Nonfederal Information Systems and Organizations* z grudnia 2016. National Institute of Standards and Technology (NIST), United States Department of Commerce.

4.1. ZASADY KONTROLOWANIA DOSTĘPU

W szerokim rozumieniu całość bezpieczeństwa komputerowego skupia się na kontrolowaniu dostępu. I rzeczywiście, dokument RFC 4949 definiuje bezpieczeństwo komputerowe następująco: środki realizacji i gwarantowania usług zabezpieczeń w systemie komputerowym ze szczególnym uwzględnieniem tych zapewniających usługi kontrolowania dostępu. Ten rozdział jest poświęcony węższej, bardziej specyficznej koncepcji kontrolowania dostępu — kontrolowaniu dostępu realizującego zasady bezpieczeństwa definiujące, kto lub co (np. proces) może mieć dostęp do określonego zasobu systemowego oraz dozwolony rodzaj dostępu w każdym takim przypadku.

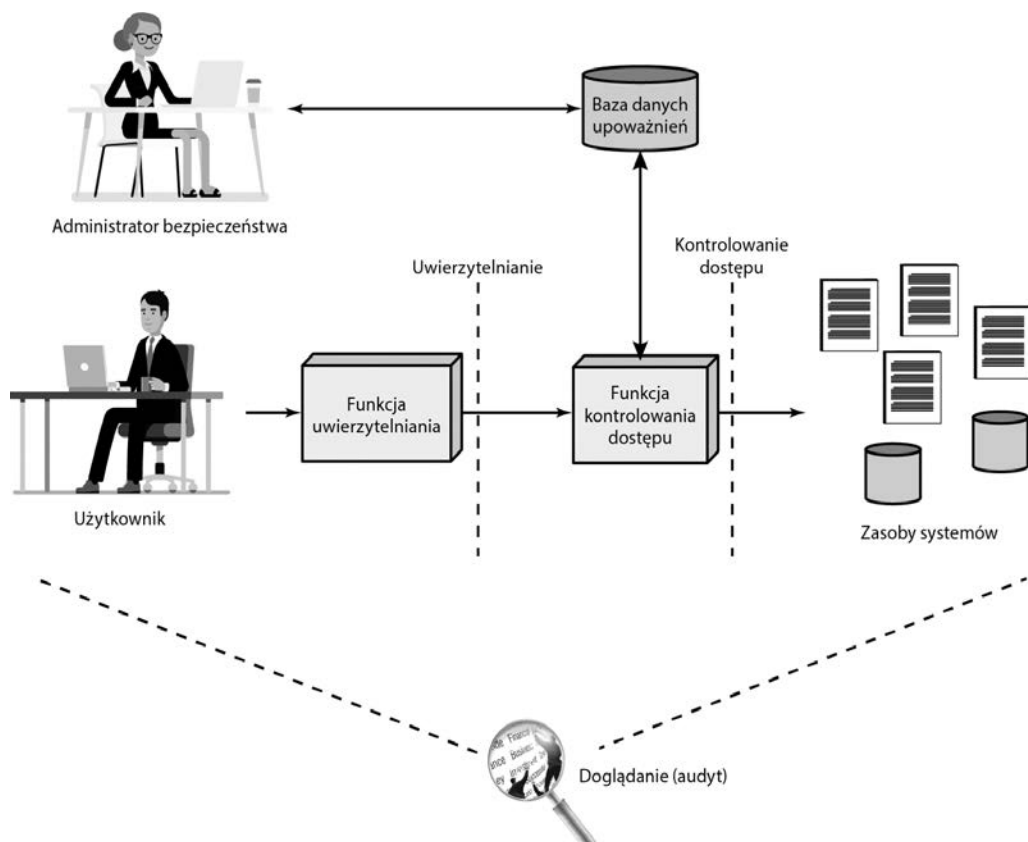
Kontekst kontrolowania dostępu

Na rysunku 4.1 pokazano kontrolowanie dostępu w poszerzonym kontekście. Oprócz kontrolowania dostępu ten kontekst obejmuje następujące jednostki i funkcje:

- **Uwierzytelnianie** (ang. *authentication*). Weryfikowanie, że poświadczenia użytkownika lub innej jednostki systemowej są ważne.
- **Upoważnianie** (ang. *authorization*). Udzielanie jednostce systemu prawa lub pozwolenia na dostęp do zasobu systemowego. Ta funkcja określa, kto jest zaufany w określonym celu.
- **Audyt** (wgląd, ang. *audit*). Niezależny przegląd i badanie zapisów i aktywności systemu w celu sprawdzenia adekwatności kontroli systemowej, przestrzegania zgodności z przyjętymi zasadami i procedurami operacyjnymi oraz wykrywania naruszeń bezpieczeństwa, jak również zalecanie wszelkich zmian w kontroli, polityce i procedurach, których konieczność wprowadzenia zostanie wykazana.

Mechanizm kontrolowania dostępu pośredniczy między użytkownikiem (lub procesem działającym na zlecenie użytkownika) a zasobami systemowymi, takimi jak aplikacje, systemy operacyjne, zapory sieciowe, routery, pliki i bazy danych. System musi najpierw uwierzytelnić użytkownika ubiegającego się o dostęp. Funkcja uwierzytelnienia zazwyczaj ustala, czy użytkownikowi wolno w ogóle mieć prawo wstępu do systemu. W następnej kolejności funkcja kontroli dostępu określa, czy danemu użytkownikowi przysługuje zamawiany rodzaj dostępu. Administrator bezpieczeństwa utrzymuje bazę danych z upoważnieniami określającymi, jaki rodzaj dostępu i do których zasobów jest dozwolony danemu użytkownikowi. Funkcja kontroli dostępu sprawdza tę bazę danych, aby rozstrzygnąć, czy udzielić dostępu. Funkcja doglądania (audytu) nadzoruje i utrzymuje rekord odnotowujący dostępy użytkownika do zasobów systemowych.

W prostym modelu przedstawionym na rysunku 4.1 funkcja kontrolowania dostępu jest ukazana w postaci pojedynczego modułu logicznego. W praktyce w realizacji funkcji kontrolowania dostępu może brać udział wiele współpracujących ze sobą komponentów. Wszystkie systemy operacyjne mają co najmniej elementarną, a w wielu przypadkach



Rysunek 4.1. Związek między kontrolowaniem dostępu i innymi funkcjami bezpieczeństwa.
Źródło: na podstawie [SAND94]

całkiem odporną składową kontrolowania dostępu. Dodatkowe pakiety bezpieczeństwa mogą uzupełniać rdzenne możliwości kontrolowania dostępu systemu operacyjnego. Poszczególne aplikacje lub narzędzia, takie jak system zarządzania bazą danych, również zawierają funkcje kontrolowania dostępu. Urządzenia zewnętrzne w rodzaju zapór sieciowych także mogą świadczyć usługi kontrolowania dostępu.

Zasady kontrolowania dostępu

Polityka kontrolowania dostępu, którą może ucieleśniać baza danych upoważnień, określa dozwolone rodzaje dostępu, warunki jego występowania oraz uprawnione podmioty. Zasady kontrolowania dostępu dzielą się ogólnie na następujące kategorie:

- **Uznaniowe kontrolowanie dostępu** (ang. *discretionary access control* — DAC). Kontrolowanie dostępu na podstawie tożsamości ubiegającej się osoby (podmiotu) oraz reguł dostępu (upoważnień) określających, co jej wolno (lub czego nie wolno) robić.

Politykę tę nazywa się *uznaniową*, ponieważ jednostka może mieć takie prawa dostępu, które umożliwią jej udzielenie dostępu do pewnego zasobu innej jednostce wedle własnej woli.

- **Obligatoryjne kontrolowanie dostępu** (ang. *mandatory access control* — MAC). Kontrolowanie dostępu oparte na porównywaniu etykiet bezpieczeństwa (wskazujących stopień wrażliwości lub krytycznego znaczenia zasobów systemu) z certyfikatem bezpieczeństwa (pokazującym, które jednostki systemu kwalifikują się do dostępu do pewnych zasobów). Tę politykę zwie się *obligatoryjną* (obowiązkową), gdyż jednostka posiadająca pozwolenie na dostęp do zasobu nie może ot tak, wedle swojego uznania, udostępnić go innej jednostce.
- **Kontrolowanie dostępu według ról** (ang. *role-based access control* — RBAC). Kontrolowanie dostępu oparte na rolach odgrywanych przez użytkowników w systemie i regułach określających, jakie dostępy są dozwolone użytkownikom występującym w danych rolach.
- **Kontrolowanie dostępu według atrybutów** (ang. *attribute-based access control* — ABAC). Kontrolowanie dostępu oparte na atrybutach użytkownika, zasobach do udostępnienia i bieżących warunkach środowiskowych.

DAC jest tradycyjną metodą urzeczywistniania kontroli dostępu. Opisano ją w podrozdziałach 4.3 i 4.4. MAC jest koncepcją, która wyewoluowała z wymagań dotyczących bezpieczeństwa informacji wojskowej i najlepiej wpisuje się w kontekst systemów zaufanych, którymi zajmujemy się w rozdziale 27. Zarówno RBAC, jak i ABAC zyskują wciąż na popularności; są one omówione, odpowiednio, w podrozdziałach 4.5 i 4.6.

Te cztery rodzaje polityki nie wykluczają się nawzajem. Mechanizm kontrolowania dostępu może wykorzystywać dwie lub nawet trzy spośród nich, aby obejmować różne klasy zasobów systemu.

4.2. PODMIOTY, OBIEKTY I PRAWA DOSTĘPU

Podstawowymi elementami kontrolowania dostępu są podmiot, obiekt i prawo dostępu.

Podmiot (ang. *subject*) jest jednostką zdolną do dostępu do obiektów. Na ogół pojęcie podmiotu zrównuje się z procesem¹. Dowolny użytkownik lub aplikacja uzyskuje w rzeczywistości dostęp do obiektu za pomocą procesu reprezentującego użytkownika lub aplikację. Proces przejmuje atrybuty użytkownika, takie jak prawa dostępu.

Podmiot zazwyczaj jest obciążany odpowiedzialnością za podejmowane działania i w ramach doglądania może być zapisywany ślad związków podmiotu z działaniami istotnymi dla bezpieczeństwa wykonywanymi przez podmiot na obiekcie.

¹ Inni autorzy podmiot (w opisywanym tu znaczeniu) nazywają też domeną ochrony (ang. *protection domain*) — *przyp. tłum.*

Podstawowe systemy kontrolowania dostępu zwykle definiują trzy klasy podmiotów z różnymi prawami dostępu w każdej klasie:

- **Właściciel** (ang. *owner*). Może to być twórca zasobu, na przykład pliku. W przypadku zasobów systemowych prawa własności mogą należeć do administratora systemu. Jeśli chodzi o zasoby dotyczące projektów, ich własność może być przypisana administratorowi lub kierownikowi projektu.
- **Grupa** (ang. *group*). Oprócz przywilejów przypisanych właścicielowi nazwanej grupie użytkowników mogą również być przyznane prawa dostępu, tak że przynależność do grupy będzie wystarczała do korzystania z tych praw. W większości schematów użytkownik może należeć do wielu grup.
- **Świat** (ang. *world*). Najmniejsze przywileje są udzielane użytkownikom, którzy mogą mieć dostęp do systemu, lecz nie są zaliczeni do kategorii właścicieli lub grupy w związku z danym zasobem.

Obiekt (ang. *object*) jest zasobem, do którego dostęp podlega kontroli. Przykładami są rekordy, bloki, strony, segmenty, pliki, fragmenty plików, katalogi, drzewa katalogów, skrzynki pocztowe, komunikaty i programy. Niektóre systemy kontrolowania dostępu obejmują również bity, bajty, słowa, procesory, porty komunikacyjne, zegary i węzły sieci.

Liczba i typy obiektów chronionych przez system kontrolowania dostępu zależą od środowiska, w którym dokonuje się kontrolowania dostępu, oraz od pożądaných kompromisów między bezpieczeństwem, jego złożonością i związanymi z tym uciążliwościami przetwarzania a wygodą użytkowania.

Prawo dostępu (ang. *access right*) opisuje, w jaki sposób podmiot może sięgać po obiekt. Do praw dostępu należą:

- **Czytanie** (ang. *read*). Użytkownik może oglądać informacje zgromadzone w zasobie systemu (np. w pliku, wybranych rekordach pliku, wybranych polach w rekordzie lub w pewnych ich kombinacjach). Dostęp do czytania obejmuje zdolność do kopiowania lub drukowania.
- **Pisanie** (ang. *write*). Użytkownik może dodawać, modyfikować lub usuwać dane w zasobie systemu (np. w pliku, rekordach, programach). Dostęp do pisania obejmuje dostęp do czytania.
- **Wykonywanie** (ang. *execute*). Użytkownik może wykonywać określone programy.
- **Usuwanie** (ang. *delete*). Użytkownik może usuwać pewne zasoby systemu, na przykład pliki lub rekordy.
- **Tworzenie** (ang. *create*). Użytkownik może tworzyć nowe pliki, rekordy lub pola.
- **Wyszukiwanie** (ang. *search*). Użytkownik może wyprowadzać wykaz plików z katalogu lub w inny sposób przeszukiwać katalog.

4.3. UZNANIOWE KONTROLOWANIE DOSTĘPU

Jak powiedziano poprzednio, schemat uznaniowego kontrolowania dostępu polega na tym, że jakaś jednostka z własnej woli może udzielić praw dostępu pozwalających innej jednostce sięgać po pewien zasób. Ogólne podejście do DAC² stosowane przez system operacyjny lub system zarządzania bazą danych zasadza się na **macierzy dostępu** (ang. *access matrix*). Pomysł macierzy dostępu został sformułowany przez Lampsona [LAMP69, LAMP71], a później uściślony przez Grahama i Denninga [GRAH72, DENN71] oraz Harrisona i in. [HARR76].

Jeden z wymiarów macierzy tworzą zidentyfikowane podmioty, które mogą pretendować do dostępu do danych. Lista ta będzie na ogół zawierać poszczególnych użytkowników lub grupy użytkowników, choć oprócz użytkowników lub zamiast nich kontrolowaniem dostępu mogą być również objęte terminale, wyposażenie sieciowe, komputery w sieci lub aplikacje. W drugim wymiarze mieszczą się obiekty, do których może następować dostęp. Na największym poziomie szczegółowości obiektami mogą być indywidualne pola danych. Jednostki wykazujące większe zgrupowanie, jak rekordy, pliki lub nawet cała baza danych, też mogą być obiektami tej macierzy. Każdy wpis w macierzy zawiera prawa dostępu danego podmiotu do danego obiektu.

Rysunek 4.2a, oparty na rysunku z [SAND94], ukazuje prosty przykład macierzy dostępu. Mamy tu użytkownika A posiadającego pliki 1 i 3 oraz prawa ich czytania i zapisywania. Użytkownik B ma prawo czytania pliku 1 itd.

W praktyce macierz dostępu jest rzadka i jest implementowana przez zdekomponowanie jej na dwa sposoby. Macierz może być zdekomponowana według kolumn, w rezultacie czego otrzymujemy **listy kontroli dostępu** (ang. *access control lists* — ACLs), widoczne na rysunku 4.2b. Dla każdego obiektu lista kontroli dostępu wymienia użytkowników i przysługujące im prawa. Lista kontroli dostępu może zawierać wpis domyślny, czyli publiczny. Umożliwia to użytkownikom, którym jawnie nie przyznano określonych praw, posiadanie domyślnego zbioru praw. W domyślnym zbiorze praw należy zawsze przestrzegać reguły najmniejszych przywilejów lub udzielać dostęp tylko do czytania — zależnie od tego, która z tych dwu opcji jest osiągalna. Elementy listy mogą zawierać poszczególnych użytkowników lub grupy użytkowników.

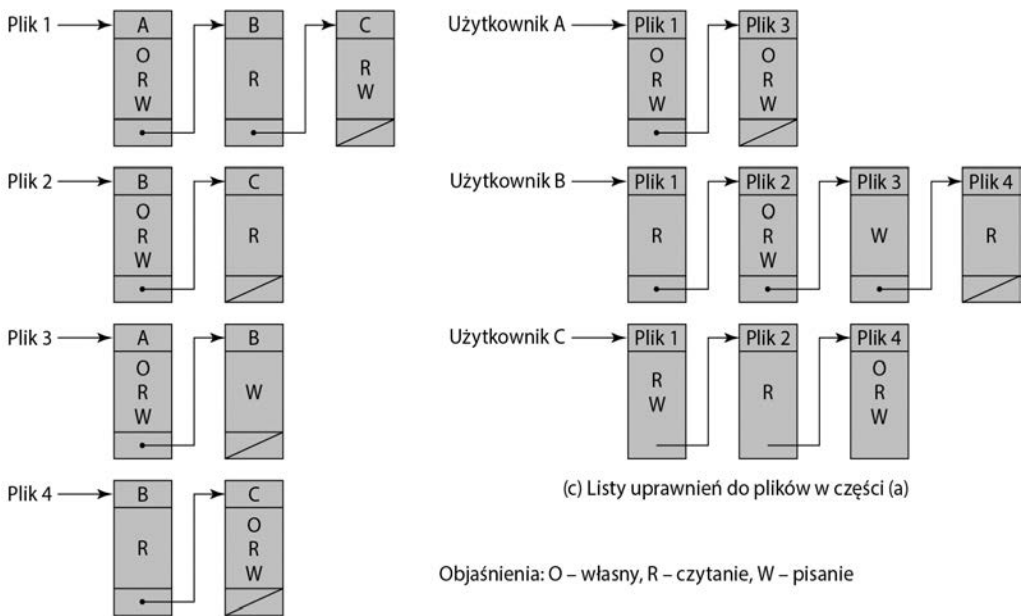
Jeśli jest potrzebne określenie, które podmioty mają prawa dostępu — i jakie — do konkretnego zasobu, listy ACL są wygodne, ponieważ każda lista ACL zawiera informacje dotyczące danego zasobu. Jednak ta struktura danych nie jest wygodna w wypadku ustalenia praw dostępu przysługujących konkretnemu użytkownikowi.

Podział według wierszy daje **bilety uprawnień** (mandaty uprawnień, ang. *capability tickets*), pokazane na rysunku 4.2c. Bilet uprawnień określa upoważnienia użytkownika do wykonywania operacji na obiektach. Każdy użytkownik ma pewną liczbę biletów i może być upoważniony do wypożyczania lub przekazywania ich innym. Ponieważ bilety mogą

² Ang. *discretionary access control*, czyli właśnie uznaniowe kontrolowanie dostępu — *przyp. tłum.*

		OBIEKTY			
		Plik 1	Plik 2	Plik 3	Plik 4
PODMIOTY	Użytkownik A	Własny Czytanie Pisanie		Własny Czytanie Pisanie	
	Użytkownik B	Czytanie	Własny Czytanie Pisanie	Pisanie	Czytanie
	Użytkownik C	Czytanie Pisanie	Czytanie		Własny Czytanie Pisanie

(a) Macierz dostępu



Rysunek 4.2. Przykłady struktur kontrolowania dostępu

być rozsiane w systemie, stanowią większy problem bezpieczeństwa niż listy kontroli dostępu. Należy chronić i zapewnić (co zwykle należy do obowiązków systemu operacyjnego) nienaruszalność biletu. W szczególności bilety muszą być nie do podrobienia. Jednym ze sposobów, aby to osiągnąć, jest przechowywanie przez system wszystkich biletów w imieniu użytkowników. Te bilety powinny być przechowywane w rejonie pamięci niedostępnym dla użytkowników. Inna możliwość polega na dołączeniu niefałszowalnego żetonu do uprawnień zapisanych w bilecie. Może nim być wielkie losowe

hasło lub kryptograficzny kod uwierzytelniający komunikatu. Ta wartość jest weryfikowana według odpowiedniego zasobu przy każdym żądaniu dostępu. Taka postać biletu uprawnień nadaje się do użytku w środowisku rozproszonym, gdy nie można zagwarantować bezpieczeństwa jego zawartości.

Wygodne i niewygodne cechy biletów uprawnień mają się odwrotnie do cech list ACL. Łatwo jest ustalić zbiór praw dostępu przysługujących użytkownikowi, lecz znacznie trudniej jest ustalić listę użytkowników mających określone prawa dostępu do określonego zasobu.

[SAND94] proponuje strukturę danych, która nie jest rzadka jak macierz dostępu, lecz jest wygodniejsza zarówno od list ACL, jak i od list uprawnień (tabela 4.2). **Tabela upoważnień** (ang. *authorization table*) zawiera po jednym wierszu na jedno prawo dostępu jednego podmiotu do jednego zasobu. Sortowanie lub sięganie do tabeli według podmiotów jest równoważne liście uprawnień. Sortowanie lub dostęp do tabeli według obiektów jest równoważne ACL. W relacyjnej bazie danych można łatwo zrealizować tabelę upoważnień tego typu.

Tabela 4.2. Tabela upoważnień do plików z rysunku 4.2

Podmiot	Tryb dostępu	Obiekt
A	Własny	Plik 1
A	Czytanie	Plik 1
A	Pisanie	Plik 1
A	Własny	Plik 3
A	Czytanie	Plik 3
A	Pisanie	Plik 3
B	Czytanie	Plik 1
B	Własny	Plik 2
B	Czytanie	Plik 2
B	Pisanie	Plik 2
B	Pisanie	Plik 3
B	Czytanie	Plik 4
C	Czytanie	Plik 1
C	Pisanie	Plik 1
C	Czytanie	Plik 2
C	Własny	Plik 4
C	Czytanie	Plik 4
C	Pisanie	Plik 4

Model kontrolowania dostępu

W tym punkcie wprowadzamy ogólny model polityki DAC, opracowany przez Lampsona, Grahama i Denninga [LAMP71, GRAH72, DENN71]. Model ten zakłada istnienie zbioru podmiotów, zbioru obiektów i zbioru reguł rządzących dostępem podmiotów do obiektów. Zdefiniujemy stan ochrony systemu jako zbiór informacji w danej chwili określających prawa dostępu każdego podmiotu do każdego obiektu. Możemy wyróżnić trzy wymagania: reprezentowanie stanu ochrony, wymuszanie praw dostępu i zezwalanie podmiotom na zmienianie stanu ochrony w pewien sposób. Model uwzględnia wszystkie trzy wymagania, dając ogólny opis logiczny systemu DAC.

W celu reprezentowania stanu ochrony rozszerzamy przestrzeń obiektów w macierzy dostępu o następujące elementy:

- **Procesy.** Prawa dostępu obejmują zdolność usuwania procesu, jego wstrzymywania (blokowania) i budzenia.
- **Urządzenia.** Prawa dostępu obejmują zdolność czytania lub zapisywania urządzenia, kontrolowanie jego działania (np. wyszukiwania na dysku) oraz blokowania lub odblokowywania urządzenia w związku z jego użyciem.
- **Komórki lub obszary pamięci.** Prawa dostępu obejmują zdolność czytania lub zapisywania pewnych chronionych komórek lub regionów pamięci, do których dostęp jest domyślnie zabroniony.
- **Podmioty.** Prawa dostępu dotyczące podmiotu odnoszą się do możliwości udzielania lub odbierania praw dostępu danego podmiotu innym obiektom (podmiotom), co wyjaśniono dalej.

Na rysunku 4.3 przedstawiono przykład. Każdy wpis $A[S, X]$ macierzy dostępu A zawiera napisy, zwane atrybutami dostępu, które określają prawa dostępu podmiotu S do obiektu X . Na rysunku 4.3 podmiot S_1 może na przykład czytać plik F_1 , ponieważ w pozycji $A[S_1, X_1]$ widnieje atrybut *czytaj*.

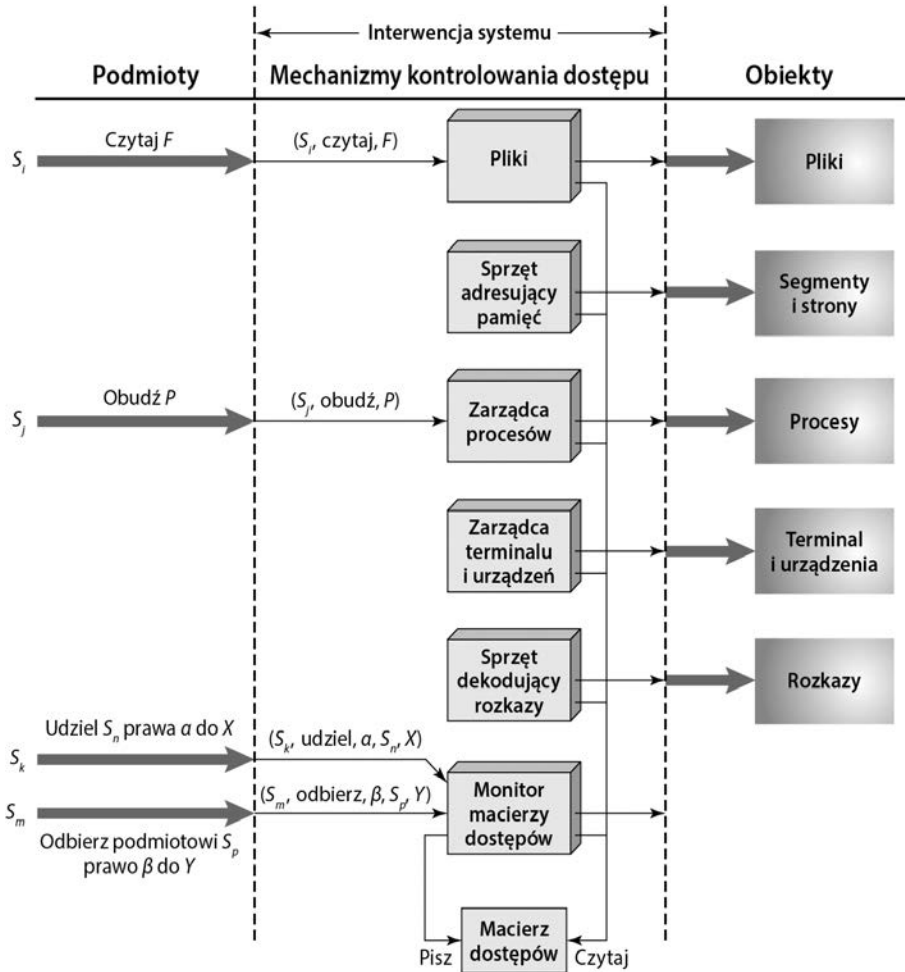
Z logicznego lub funkcjonalnego punktu widzenia z każdym typem obiektu jest skojarzony osobny moduł kontrolowania dostępu (zob. rysunek 4.4). Moduł taki ocenia zamówiony przez podmiot dostęp do obiektu, sprawdzając, czy dane prawo dostępu istnieje. Próba dostępu powoduje wykonanie następujących kroków:

1. Podmiot S_0 wydaje żądanie typu α pod adresem obiektu X .
2. Żądanie to powoduje wygenerowanie przez system (system operacyjny lub jakiś moduł interfejsu kontroli dostępu) komunikatu postaci (S_0, α, X) dla kontrolera X .
3. Kontroler zagląda do macierzy A , żeby sprawdzić, czy α występuje w $A[S_0, X]$. Jeżeli tak, dostęp zostaje udzielony, w przeciwnym razie następuje odmowa dostępu i naruszenie ochrony. Owo naruszenie powinno spowodować ostrzeżenie i odpowiednie działanie.

		OBIEKTY								
		Podmioty			Pliki		Procesy		Napędy dysków	
		S_1	S_2	S_3	F_1	F_2	P_1	P_2	D_1	D_2
PODMIOTY	S_1	Kontrola	Właściciel	Kontrola właścicielska	Czytaj *	Czytaj jako właściciel	Obudź	Obudź	Wyszukaj	Właściciel
	S_2		Kontrola		Pisz *	Wykonaj			Właściciel	Wyszukaj*
	S_3			Kontrola		Pisz	Zatrzymaj			

* = Znacznik kopiowania ustawiony

Rysunek 4.3. Poszerzona macierz kontroli dostępu



Rysunek 4.4. Organizacja funkcji kontroli dostępu

Na rysunku 4.4 zasugerowano, że każdy dostęp podmiotu do obiektu jest negocjowany z kontrolerem tego obiektu, a decyzja kontrolera opiera się na aktualnej zawartości macierzy dostępów. Ponadto niektóre podmioty są uprawnione do wykonywania określonych zmian w macierzy dostępów. Żądanie modyfikacji macierzy dostępów jest traktowane jako dostęp do tej macierzy, a poszczególne jej wpisy są traktowane jak obiekty. Dostępy tego rodzaju są negocjowane z kontrolerem macierzy dostępów, który sprawuje nadzór nad jej uaktualnieniami.

Model zawiera również zbiór reguł rządzących modyfikowaniem macierzy dostępów, co pokazano w tabeli 4.3. Na tę okoliczność wprowadzamy prawa dostępu *właściciel* i *kontrola* oraz pojęcie znacznika kopiowania, wyjaśnione w następnych akapitach.

Tabela 4.3. Polecenia systemu kontrolowania dostępu

Reguła	Polecenie (wydawane przez S_0)	Upoważnienie	Operacja
R1	przekaż S prawa $\left\{ \begin{matrix} \alpha^* \\ \alpha \end{matrix} \right\}$ do X	„ α^* ” w $A[S_0, X]$	zapamiętaj $\left\{ \begin{matrix} \alpha^* \\ \alpha \end{matrix} \right\}$ w $A[S, X]$
R2	udziel S prawa $\left\{ \begin{matrix} \alpha^* \\ \alpha \end{matrix} \right\}$ do X	„właściciel” w $A[S_0, X]$	zapamiętaj $\left\{ \begin{matrix} \alpha^* \\ \alpha \end{matrix} \right\}$ w $A[S, X]$
R3	usuń α z S, X	„kontrola” w $A[S_0, S]$ lub „właściciel” w $A[S_0, X]$	usuń α z $A[S, X]$
R4	$w \leftarrow$ czytaj S, X	„kontrola” w $A[S_0, S]$ lub „właściciel” w $A[S_0, X]$	kopiuje $A[S, X]$ do w
R5	utwórz obiekt X	Nic	dodaj kolumnę obiektu X do A ; zapamiętaj prawo „właściciel” w $A[S_0, X]$
R6	zlikwiduj obiekt X	„właściciel” w $A[S_0, X]$	usuń kolumnę X z A
R7	utwórz podmiot S	Nic	dodaj wiersz S do A ; wykonaj utwórz obiekt S ; zapamiętaj prawo „kontrola” w $A[S, S]$
R8	zlikwiduj podmiot S	„właściciel” w $A[S_0, S]$	usuń wiersz S z A ; wykonaj usuń obiekt S

Pierwsze trzy reguły dotyczą przekazywania, udzielania i usuwania praw dostępu. Przypuśćmy, że w $A[S_0, X]$ istnieje wpis α^* . Oznacza to, że S_0 ma prawo dostępu α do obiektu X i — z uwagi na obecność znacznika kopiowania — może przekazać to prawo wraz ze znacznikiem kopiowania, lub bez niego, innemu podmiotowi. Zdolność tę wyraża reguła R1. Podmiot mógłby przekazać prawo dostępu bez znacznika kopiowania na wypadek, gdyby zachodziła obawa, że nowy podmiot może złośliwie przekazać dane prawo innemu podmiotowi, który nie powinien go posiadać. Na przykład S_1 może umieścić prawo *czytanie* lub *czytanie** w dowolnej pozycji kolumny F_1 macierzy. Reguła R2 głosi, że jeśli S_0 jest mianowany właścicielem obiektu X , to S_0 może udzielić prawa dostępu do tego obiektu dowolnemu innemu podmiotowi. Reguła R2 głosi, że S_0 może dodać

dowolne prawo dostępu do $A[S, X]$ dowolnemu S , jeśli S_0 ma dostęp *właściciela* do X . Reguła R3 pozwala podmiotowi S_0 usunąć dowolne prawo dostępu z dowolnej pozycji macierzy w wierszu, w którym S_0 kontroluje dany podmiot, i z dowolnej pozycji macierzy w kolumnie, w której S_0 jest właścicielem obiektu. Reguła R4 pozwala podmiotowi czytać tę część macierzy, której jest właścicielem lub nad którą sprawuje kontrolę.

Pozostałe reguły w tabeli 4.3 rządzą tworzeniem i usuwaniem podmiotów i obiektów. Reguła R5 głosi, że dowolny podmiot może utworzyć nowy obiekt, stając się jego właścicielem, a potem może udzielać do niego dostępu oraz go cofać. Na mocy reguły R6 właściciel obiektu może zlikwidować obiekt, co spowoduje usunięcie odpowiedniej kolumny macierzy dostępu. Reguła R7 umożliwia dowolnemu podmiotowi utworzenie nowego podmiotu; twórca staje się właścicielem nowego podmiotu, a nowy podmiot ma kontrolę dostępu do samego siebie. Reguła R8 pozwala właścicielowi podmiotu na usunięcie z macierzy dostępu wiersza i kolumny (jeśli istnieją kolumny podmiotu) wskazanych przez ten podmiot.

Zbiór reguł w tabeli 4.3 jest przykładem reguł, które można zdefiniować w systemie kontrolowania dostępu. Poniżej podano przykłady dodatkowych lub alternatywnych reguł, które można by dołączyć. Można zdefiniować **prawo przeniesienia** (ang. *transfer-only*), które powodowałoby przekazanie prawa docelowemu podmiotowi z jednoczesnym odebraniem go podmiotowi przekazującemu. Liczbę właścicieli obiektu lub podmiotu można ograniczyć do jednego przez niedopuszczenie, aby znacznik kopiowania towarzyszył prawu właściciela.

Zdolności podmiotu do tworzenia innego podmiotu i posiadania prawa *właściciela* do tego podmiotu można użyć do zdefiniowania hierarchii podmiotów. Na przykład na rysunku 4.3 S_1 jest właścicielem S_2 i S_3 , więc S_2 i S_3 podlegają S_1 . Na mocy reguł z tabeli 15.1 S_1 może przyznawać i cofać podmiotowi S_2 prawa dostępu, które sam już posiada. Podmiot może więc utworzyć inny podmiot z podzbiorem własnych praw dostępu. Może to być użyteczne, gdy na przykład podmiot wywołuje aplikację, której nie w pełni ufa, i nie chce, aby ta aplikacja mogła przekazać prawa dostępu innym podmiotom.

Domeny ochrony

Omówiony przez nas model macierzy kontroli dostępu kojarzy jak na razie zbiór zdolności z użytkownikiem. Ogólniejszym i bardziej elastycznym podejściem zaproponowanym w [LAMP71] jest skojarzenie zdolności z domenami ochrony. **Domena ochrony** (ang. *protection domain*) jest zbiorem obiektów wraz z przysługującymi do nich prawami dostępu. W kategoriach macierzy dostępu domenę ochrony definiuje wiersz macierzy. Do tej pory zrównywaliśmy każdy wiersz z konkretnym użytkownikiem. Wobec tego w tak ograniczonym modelu każdy użytkownik ma domenę ochrony, a wszystkie utworzone przez niego procesy mają prawa dostępu zdefiniowane przez tę samą domenę ochrony.

Ogólniejsze pojmowanie domeny ochrony daje więcej elastyczności. Na przykład użytkownik może tworzyć procesy z podzbiorem swoich praw dostępu zdefiniowanym jako nowa domena ochrony. Ogranicza to zdolności procesu. Taki schemat można by zasto-

sować w procesie serwera do tworzenia procesów dla różnych klas użytkowników. Użytkownik mógłby również zdefiniować domenę ochrony w odniesieniu do programu, do którego nie ma pełnego zaufania, więc jego dostępności zostałyby ograniczone do bezpiecznego podzbioru praw dostępu użytkownika.

Związek między procesem a domeną może być statyczny lub dynamiczny. Na przykład proces może wykonywać ciąg procedur i wymagać w każdej procedurze różnych praw dostępu, takich jak czytanie lub zapisywanie pliku. Ogólnie biorąc, chcielibyśmy minimalizować prawa dostępu posiadane przez dowolnego użytkownika lub proces w danym czasie. Zastosowanie domen ochrony stanowi prosty środek spełnienia tego postulatu.

Jedna z form domeny ochrony wiąże się z dokonywanym w wielu systemach operacyjnych, na przykład w systemie UNIX, rozróżnianiem między użytkownikiem a jądrem. Program użytkownika pracuje w **trybie użytkownika** (ang. *user mode*), w którym pewne obszary pamięci są chronione przed wykorzystaniem przez użytkownika, a pewne instrukcje nie mogą być wykonane. Gdy proces użytkownika wywołuje procedurę systemową, jest ona wykonywana w trybie systemu, lub — jak to się zwykle nazywać — w **trybie jądra** (ang. *kernel mode*), w którym instrukcje uprzywilejowane mogą być wykonywane, a dostęp do chronionych obszarów pamięci jest dozwolony.

4.4. PRZYKŁAD: KONTROLOWANIE DOSTĘPU W UNIKSOWYM SYSTEMIE PLIKÓW

Nasze omówienie kontrolowania dostępu do plików w systemie UNIX rozpoczniemy od wprowadzenia kilku podstawowych pojęć związanych z uniksowymi plikami i katalogami.

Wszystkie typy plików w UNIX-ie są administrowane przez system operacyjny za pomocą i-węzłów. **I-węzeł** (węzeł indeksowy, ang. *inode*)³ jest strukturą sterowania zawierającą kluczowe informacje potrzebne systemowi operacyjnemu w związku z danym plikiem. Z jednym i-węzłem można związać kilka nazw plików, lecz aktywny i-węzeł jest skojarzony tylko z jednym plikiem i każdy plik jest kontrolowany z użyciem tylko jednego i-węzła. W i-węźle są przechowywane atrybuty pliku oraz związane z nim prawa dostępu i inne informacje sterujące. Na dysku znajduje się tablica i-węzłów (lub lista i-węzłów), która zawiera i-węzły wszystkich plików w systemie plików. Gdy plik jest otwierany, jego i-węzeł jest sprowadzany do pamięci głównej i zapamiętywany w tablicy i-węzłów rezydującej w pamięci.

Katalogi tworzą strukturę hierarchicznego drzewa⁴. Każdy **katalog** (ang. *directory*) może zawierać pliki i (lub) inne katalogi. Katalog zawarty w innym katalogu nazywa się **podkatalogiem** (ang. *subdirectory*). Katalog jest po prostu plikiem, który zawiera wykaz nazw plików oraz wskaźniki do skojarzonych z nimi i-węzłów. Zatem z każdym katalogiem jest skojarzony jego własny i-węzeł.

³ Wyraz pisany też z łącznikiem: *i-node*; inna nazwa polska: węzeł indeksujący — *przyp. tłum.*

⁴ W oryginale: *hierarchical tree* — *przyp. tłum.*

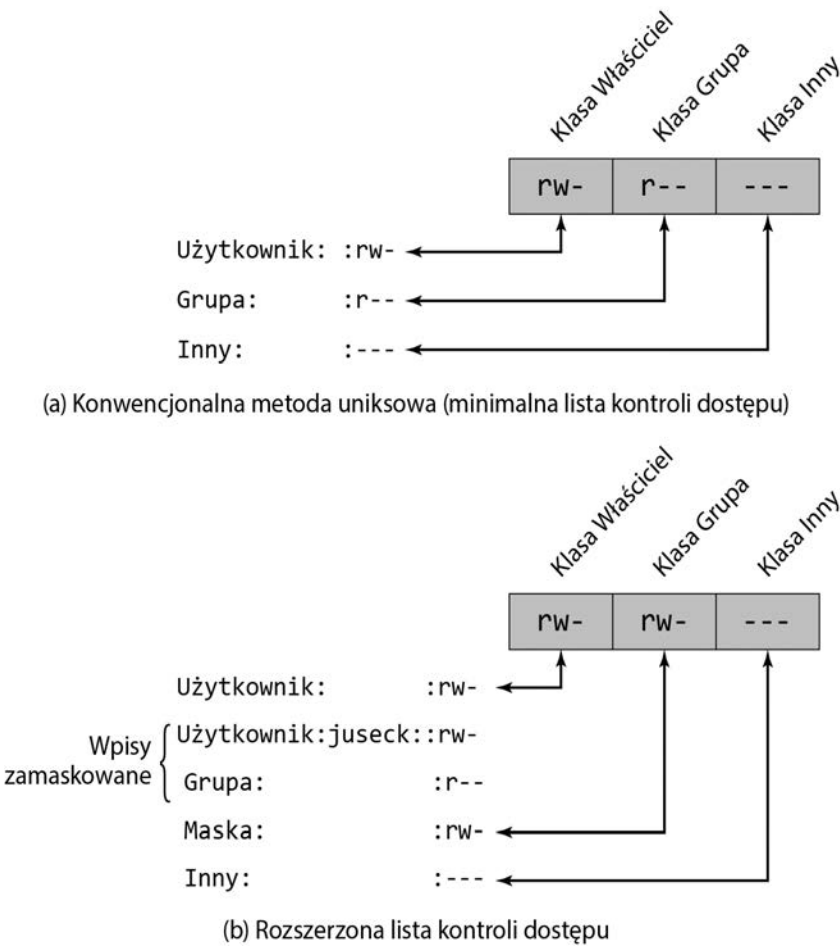
Tradycyjne kontrolowanie dostępu do plików w UNIX-ie

Większość systemów UNIX zależy od schematu kontrolowania dostępu wprowadzonego we wczesnych wersjach UNIX-a lub przynajmniej się na nim opiera. Każdy użytkownik UNIX-a ma przypisany jednoznaczny numer identyfikacyjny użytkownika (ID użytkownika). Użytkownik jest również członkiem podstawowej grupy i być może kilku innych grup, z których każda ma ID grupy. Tworzony plik staje się własnością konkretnego użytkownika i zostaje oznaczony jego identyfikatorem. Należy on także do specyficznej grupy, która początkowo jest podstawową grupą jego twórcy lub grupą katalogu jego przodka, jeśli ten katalog ma ustawione pozwolenie SetGID. Z każdym plikiem jest skojarzony zbiór 12 bitów ochrony. ID użytkownika, ID grupy oraz bity ochrony są częścią i-węzła pliku.

Dziewięć bitów ochrony określa prawa czytania, pisania i wykonywania przysługujące właścicielowi pliku, innym członkom grupy, do której ten plik należy, i wszystkim pozostałym użytkownikom. Tworzy to hierarchię złożoną z właściciela, grupy i wszystkich innych, w ramach której są używane najbardziej niezbędne zbiory praw. Na rysunku 4.5a pokazano przykład, w którym właściciel pliku ma dostęp do czytania i pisania. Wszyscy inni członkowie grupy mającej do czynienia z danym plikiem mają dostęp do czytania go, a użytkownicy spoza tej grupy nie mają do niego żadnych praw. W odniesieniu do katalogu bity czytania i pisania udzielają prawa do wyprowadzania jego zawartości („listowania”) i do tworzenia, przemianowywania albo usuwania plików w danym katalogu⁵. Bit wykonywania udziela prawa do schodzenia w głąb katalogu lub wyszukiwania nazwy pliku.

Pozostałe trzy bity określają specjalne, dodatkowe zachowanie w odniesieniu do plików lub katalogów. Dwa z nich są pozwoleniami „ustawienia ID użytkownika” (ang. „*set user ID*” — SetUID) oraz „ustawienia ID grupy” (ang. „*set group ID*” — SetGID). Jeśli bity te są ustawione w odniesieniu do pliku wykonywalnego, to system operacyjny działa następująco. Gdy użytkownik (mający przywilej wykonywania tego pliku) wykonuje plik, system czasowo przydziela mu, prócz jego własnych, prawa przysługujące ID użytkownika będącego twórcą pliku lub — odpowiednio — grupy pliku. Są one znane jako „efektywny ID użytkownika” i „efektywny ID grupy” i są stosowane w uzupełnieniu „rzeczywistego ID użytkownika” i „rzeczywistego ID grupy” użytkownika wykonującego program podczas podejmowania decyzji dotyczącej kontroli praw dostępu do tego programu. Ta zmiana obowiązuje tylko przez czas działania programu. Właściwość ta umożliwia tworzenie i wykorzystywanie programów uprzywilejowanych, które mogą używać plików normalnie niedostępnych dla innych użytkowników, i pozwala użytkownikom na dostęp do pewnych plików w kontrolowany sposób. Alternatywnie, w przypadku zastosowania do katalogu, pozwolenie SetGID wskazuje, że nowo utworzone pliki będą dziedziczyły grupę tego katalogu. Pozwolenie SetUID jest ignorowane.

⁵ Zauważmy, że pozwolenia odnoszące się do katalogu różnią się od odnoszących się do dowolnego pliku lub zawartego w nim katalogu. Fakt, że użytkownik ma prawo do pisania w katalogu, nie daje mu prawa do zapisywania pliku odnotowanego w tym katalogu. Tym drugim rządzą pozwolenia dotyczące konkretnego pliku. Użytkownik mógłby jednak mieć prawo przemianowywania pliku.



Rysunek 4.5. Kontrolowanie dostępu do pliku w UNIX-ie

Ostatnim bitem pozwoleń jest „bit przylepnosci” (ang. „sticky” bit). Ustawiony dla pliku pierwotnie wskazywał, że system powinien zachować treść pliku w pamięci po jego wykonaniu. Tego już się nie używa. Jednak zastosowany do katalogu określa, że tylko właściciel każdego pliku w danym katalogu może ten plik przemianować, przenieść lub usunąć. Jest to przydatne do administrowania plikami we wspólnych katalogach tymczasowych.

Jeden szczególny identyfikator użytkownika oznacza **superużytkownika** (ang. *super-user*). Superużytkownik jest wolny od zwykłych ograniczeń kontroli dostępu do plików i ma dostęp ogólnosystemowy. Dowolny program będący własnością „superużytkownika” i mający ustawiony jego SetUID potencjalnie daje nieograniczony dostęp do systemu każdemu wykonującemu go użytkownikowi. Dlatego do pisania takich programów należy się odnosić z wielką ostrożnością.

Ten schemat dostępu jest odpowiedni, gdy wymagania dostępu do plików pokrywają się z użytkownikami i niewielką liczbą grup użytkowników. Załóżmy na przykład, że użytkownik chce dać dostęp do czytania pliku X użytkownikom A i B oraz dostęp do czytania

pliku Y użytkownikom B i C. Potrzebowalibyśmy co najmniej dwu grup użytkowników, a użytkownik B musiałby należeć do obu grup, aby mieć dostęp do obu plików. Gdyby jednak istniała duża liczba różnych zgrupowań użytkowników potrzebujących wyboru praw dostępu do różnych plików, wówczas, aby temu sprostać, należałoby utworzyć bardzo dużo grup. Szybko staje się to nieporęczne i trudne w obsłudze — nawet gdyby w ogóle było możliwe⁶. Jednym ze sposobów pokonania tego problemu jest użycie list kontroli dostępu, występujących w większości współczesnych systemów uniksowych.

Ostatnie, co należałoby odnotować, to implementowanie przez tradycyjny schemat kontrolowania dostępu do plików w UNIX-ie prostej struktury domen ochrony. Domena jest skojarzona z użytkownikiem, a przełączanie domen odpowiada czasowej zmianie identyfikatora użytkownika.

Listy kontroli dostępu w UNIX-ie

Wiele współczesnych systemów UNIX i systemów operacyjnych opartych na UNIX-ie, w tym systemy takie jak FreeBSD, OpenBSD, Linux i Solaris, realizuje **listy kontroli dostępu** (ang. *access control lists* — ACLs). W tym punkcie opisujemy podejście zastosowane w systemie FreeBSD, lecz inne implementacje mają zasadniczo te same własności i interfejs. Własność tę określa się jako rozszerzoną listę kontroli dostępu, natomiast tradycyjne podejście uniksowe nosi miano minimalnej listy kontroli dostępu.

FreeBSD umożliwia administratorowi przypisanie do pliku listy identyfikatorów użytkowników i grup UNIX-a za pomocą polecenia `setfacl`. Z plikiem można skojarzyć dowolną liczbę użytkowników i grup, wszyscy oni mają po trzy bity ochrony (czytanie, pisanie, wykonywanie), co stanowi elastyczny mechanizm przypisywania praw dostępu. Plik nie musi mieć listy ACL, może być chroniony wyłącznie przez konwencjonalny uniksowy mechanizm dostępu do plików. Pliki FreeBSD zawierają dodatkowy bit ochrony, który wskazuje, czy plik ma rozszerzoną listę ACL.

FreeBSD i większość implementacji UNIX-a, w których występują rozszerzone listy ACL, stosują następującą strategię (zob. np. rysunek 4.5b):

1. Wpisy klasy Właściciel i klasy Inny w 9-bitowym polu pozwoleń mają to samo znaczenie, jakie mają w przypadku minimalnej listy ACL.
2. Wpis klasy Grupa określa pozwolenia grupy właścicielskiej w odniesieniu do danego pliku. Te pozwolenia reprezentują maksimum praw, które można nadać nazwanym użytkownikom lub nazwanym grupom innym niż użytkownik-właściciel. W tej ostatniej roli wpis klasy Grupa działa jak maska.
3. Z plikiem można skojarzyć dodatkowych nazwanych użytkowników i nazwane grupy — każdego i każdą z 3-bitowym polem pozwoleń. Pozwolenia wymienione dla nazwa-

⁶ Większość systemów uniksowych wprowadza limit maksymalnej liczby grup, do których może należeć użytkownik, a także ograniczenie ogólnej liczby grup dopuszczalnych w systemie.

nych użytkowników i nazwanych grup są porównywane z polem maski. Dowolne prawo dotyczące nazwanego użytkownika lub nazwanej grupy, które nie występuje w polu maski, jest zakazane.

Gdy proces żąda dostępu do obiektu systemu plików, są wykonywane dwa kroki. W pierwszym wybiera się wpis listy ACL, który najlepiej pasuje do żądającego procesu. Wpisy ACL są przeglądane w następującym porządku: właściciel, nazwani użytkownicy, (właścicielskie lub nazwane) grupy i inni. Tylko jeden wpis określa dostęp. W drugim kroku sprawdza się, czy dopasowany wpis zawiera wystarczające pozwolenia. Proces może należeć do więcej niż jednej grupy, więc może być więcej niż jeden pasujący wpis grupy. Jeśli któryś z tych pasujących wpisów grup zawiera żądane pozwolenia, wybiera się właśnie ten (wynik jest ten sam, niezależnie od tego, który wpis zostanie wybrany). Jeśli żaden z dopasowanych wpisów nie zawiera żądanych pozwoleń, nastąpi odmowa dostępu, niezależnie od tego, który wpis został wybrany.

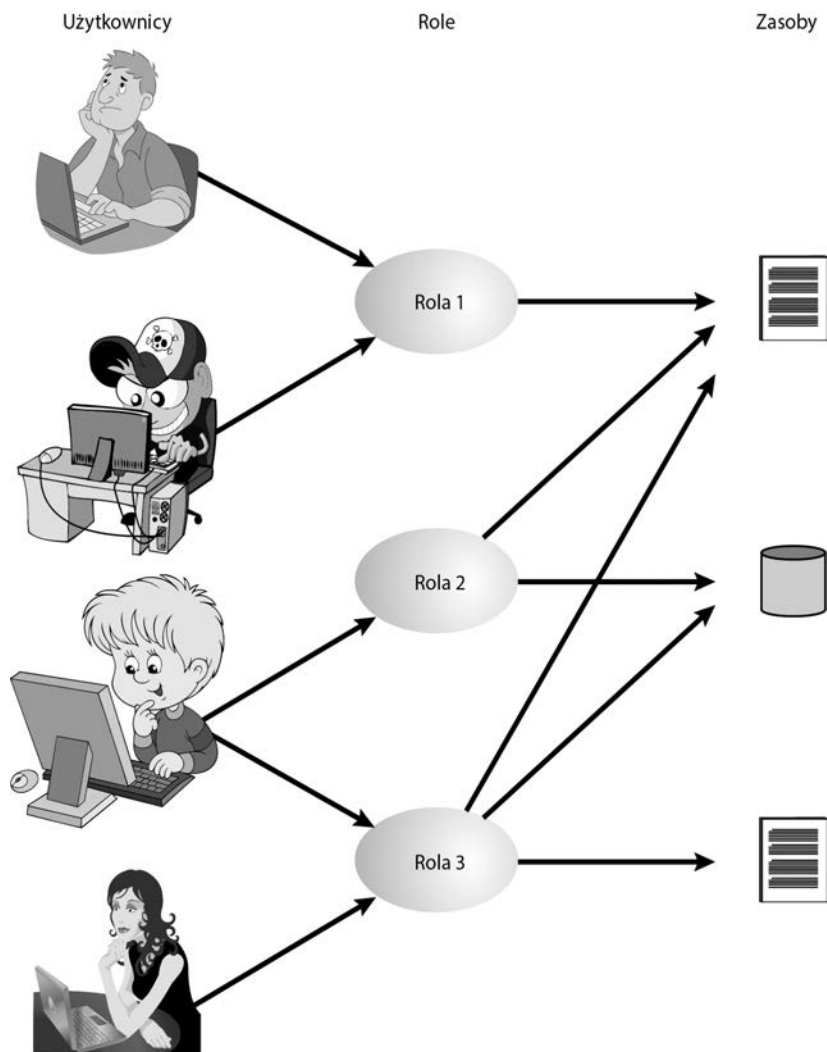
4.5. KONTROLOWANIE DOSTĘPU WEDŁUG RÓL

Tradycyjne systemy DAC definiują prawa dostępu poszczególnych użytkowników i grup użytkowników. Natomiast RBAC opiera się na rolach, które użytkownicy przyjmują w systemie, a nie na tożsamości użytkownika. Modele RBAC definiują zazwyczaj rolę jako stanowisko pracy w jakimś przedsiębiorstwie (instytucji). Systemy RBAC przypisują prawa dostępu do ról zamiast do poszczególnych użytkowników. W rezultacie użytkownicy są przypisywani do różnych ról statycznie albo dynamicznie, stosownie do ich obowiązków.

RBAC cieszy się obecnie szerokimi zastosowaniami handlowymi i pozostaje obszarem aktywnych badań. National Institute of Standards and Technology (NIST)⁷ wydał we wrześniu 2009 roku standard FIPS PUB 140-3 zatytułowany *Security Requirements for Cryptographic Modules* (z ang. „Wymagania bezpieczeństwa dotyczące modułów kryptograficznych”), w którym wymaga się zaplecza kontrolowania dostępu i administrowania na zasadzie ról.

Zależność między użytkownikami i rolami jest typu wielu na wiele, podobnie jak zależność między rolami i zasobami, czyli obiektami systemu (zob. rysunek 4.6). Zbiór zmian użytkowników — w niektórych środowiskach częstych — i przypisanie użytkownika do jednej lub większej liczby ról mogą również być dynamiczne. Zbiór ról systemowych w większości środowisk jest jednak raczej statyczny, ich dodawanie lub usuwanie jest tylko okazjonalne. Każda rola ma określone prawa dostępu do jednego lub większej liczby zasobów. Zbiór zasobów i specyficznych praw dostępu związanych z konkretną rolą również będzie się zmieniał niezbyt często.

⁷ Krajowy Instytut Standardów i Technologii amerykańskiego Ministerstwa Handlu — *przyp. tłum.*



Rysunek 4.6. Użytkownicy, role i zasoby

Do prostego przedstawienia podstawowych elementów systemu RBAC możemy użyć macierzy kontroli dostępu, jak pokazano na rysunku 4.7. Górna macierz wiąże poszczególnych użytkowników z rolami. Zwykle użytkowników jest znacznie więcej niż ról. Każdy wpis macierzy jest albo pusty, albo zaznaczony — w drugim przypadku znamionuje to przypisanie użytkownika do roli. Zauważmy, że jeden użytkownik może być przypisany do wielu ról (więcej niż jedno zaznaczenie w wierszu) i wielu użytkownikom może być przypisana jedna rola (więcej niż jedno zaznaczenie w kolumnie). Dolna macierz ma tę samą strukturę co macierz DAC — z rolami w funkcji podmiotów. Zwykle jest mało ról i wiele obiektów, czyli zasobów. W pozycjach tej macierzy występują konkretne prawa dostępu przypisane rolowi. Zauważmy, że rolę można traktować jak obiekt, co umożliwia definiowanie hierarchii ról.

	R_1	R_2	• • •	R_n
U_1	×			
U_2	×			
U_3		×		×
U_4				×
U_5				×
U_6				×
•				
•				
•				
U_m	×			

OBIEKTY

	R_1	R_2	R_n	F_1	F_2	P_1	P_2	D_1	D_2
R_1	Kontrola	Właściciel	Kontrola właścicielska	Czytaj *	Czytaj jako właściciel	Obudź	Obudź	Wyszukaj	Właściciel
R_2		Kontrola		Pisz *	Wykonaj			Właściciel	Wyszukaj *
•									
•									
•									
R_n			Kontrola		Pisz	Wstrzymaj			

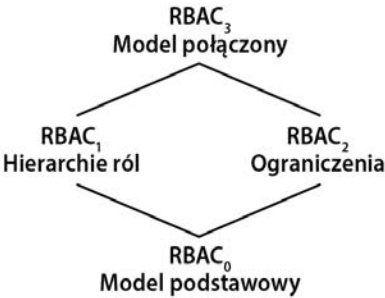
Rysunek 4.7. Reprezentacja RBAC (kontroli dostępu według ról) za pomocą macierzy kontroli dostępu

RBAC nadaje się do efektywnej implementacji zasady najmniejszych przywilejów, omówionej w rozdziale 1. Każda rola powinna skupiać minimalny zbiór praw dostępu wymaganych w danej roli. Użytkownikowi przypisanemu do roli wolno wykonywać tylko to, co jest w związku z nią wymagane. Wielu użytkowników przypisanych do tej samej roli rozporządza tym samym, minimalnym zbiorem praw dostępu.

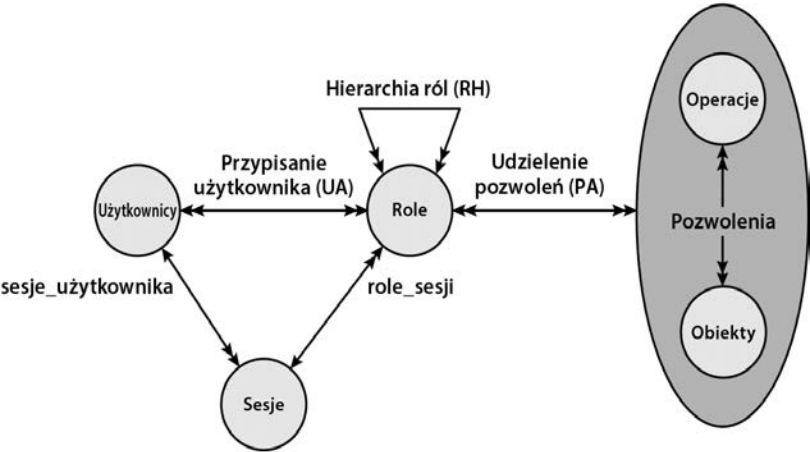
Modele wzorcowe RBAC

Do ogólnej metody RBAC można dołączyć rozmaite funkcje i usługi. Aby wyjaśnić różne aspekty RBAC, warto zdefiniować zbiór abstrakcyjnych modeli funkcjonalności RBAC.

[SAND96] określa rodzinę modeli wzorcowych, które posłużyły jako baza do dalszych prac standaryzacyjnych. Rodzina ta składa się z czterech powiązanych ze sobą modeli, jak to uwidoczniono na rysunku 4.8a i w tabeli 4.4. $RBAC_0$ zawiera minimum funkcji systemu RBAC. $RBAC_1$ zawiera funkcje $RBAC_0$, a ponadto hierarchie ról, co umożliwia dziedziczenie przez jedną rolę pozwoleń przysługujących innej roli. $RBAC_2$ zawiera $RBAC_0$ i dodaje ograniczenia sposobów, za pomocą których komponenty systemu RBAC mogą być konfigurowane. Funkcjonalność $RBAC_3$ obejmuje funkcje $RBAC_0$, $RBAC_1$ i $RBAC_2$.



(a) Zależności między modelami RBAC



(b) Modele RBAC

Rysunek 4.8. Rodzina modeli kontrolowania dostępu według ról (RBAC).
 $RBAC_0$ stanowi minimalne wymagania dla systemu RBAC. $RBAC_1$ dodaje hierarchie ról.
 $RBAC_2$ dodaje ograniczenia. $RBAC_3$ zawiera $RBAC_1$ i $RBAC_2$

Tabela 4.4. Zakres modeli RBAC

Modele	Hierarchie	Ograniczenia
RBAC ₀	Nie	Nie
RBAC ₁	Tak	Nie
RBAC ₂	Nie	Tak
RBAC ₃	Tak	Tak

MODEL PODSTAWOWY — RBAC₀

Rysunek 4.8b, bez hierarchii ról i ograniczeń, zawiera cztery typy jednostek w systemie RBAC₀:

- **Użytkownik.** Osoba mająca dostęp do danego systemu komputerowego. Każda osoba ma przydzielony ID użytkownika.
- **Rola.** Nazwane stanowisko pracy w ramach organizacji (instytucji lub przedsiębiorstwa⁸) mającej pod swoją opieką dany system komputerowy. Na ogół z każdą rolą jest skojarzony opis zakresu wynikających z niej uprawnień i odpowiedzialności, dotyczący każdego użytkownika wcielającego się w tę rolę.
- **Pozwolenie** (ang. *permission*). Zgoda na określony tryb dostępu do jednego lub większej liczby obiektów. Równoważnymi terminami są: *prawo dostępu*, *przywilej* i *upoważnienie*.
- **Sesja.** Odzworowanie między użytkownikiem i uaktywnionym podzbiorem zbioru ról, do którego przypisano użytkownika.

Linie ze strzałkami na rysunku 4.8b wskazują związki, czyli odwzorowania, przy czym pojedyncza strzałka symbolizuje jedno, a podwójna — wiele. Między użytkownikami a rolami występuje więc zależność „wiele na wiele”. Jeden użytkownik może mieć wiele ról, a wielu użytkowników może mieć przypisaną jedną rolę. Podobna zależność „wiele na wiele” występuje między rolami i pozwoleniami. Sesja jest używana do definiowania czasowego związku „jeden do wielu” między użytkownikiem a jedną lub wieloma rolami, do których dany użytkownik jest przypisany. Użytkownik ustanawia sesję, rozporządzając tylko tymi rolami, które są potrzebne w danym zadaniu. Jest to przykład urzeczywistnienia zasady najmniejszych przywilejów.

Związki „wiele na wiele” między użytkownikami a rolami i między rolami a pozwoleniami umożliwiają elastyczność i ziarnistość przydziału niewystępującą w konwencjonalnych schematach DAC. Bez tej elastyczności i ziarnistości rośnie ryzyko, że z powodu ograniczonej kontroli nad rodzajami dostępu, na które można zezwolić, użytkownik może otrzymać większy dostęp do zasobów niż trzeba. Dokument NIST RBAC podaje następujące przykłady: użytkownicy mogą chcieć wyprowadzać zawartości katalogów

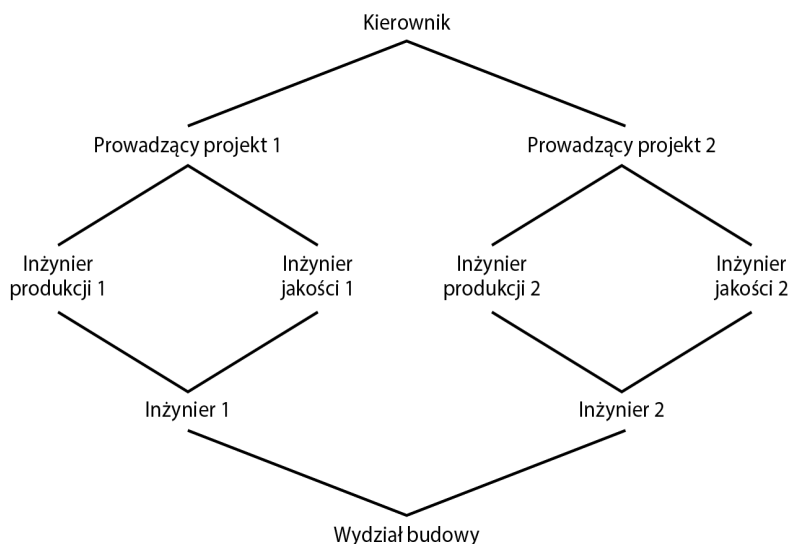
⁸ Przypominamy, że wielofunkcyjny w języku angielskim termin *organization* zastępuje w tym przekładzie, gwoili oszczędności, każdy z tu wymienionych — *przyp. tłum.*

i modyfikować istniejące pliki bez tworzenia nowych plików lub mogą potrzebować dopisywania rekordów do pliku bez tworzenia nowych albo mogą potrzebować dodawać rekordy do pliku bez modyfikowania rekordów już istniejących.

HIERARCHIE RÓL — RBAC_i

Hierarchie umożliwiają odzwierciedlenie hierarchicznej struktury ról w danej organizacji. Zazwyczaj stanowisko pracy o większej odpowiedzialności ma więcej uprawnień dostępu do zasobów. Podporządkowane stanowisko pracy może mieć podzbiór praw dostępnych w funkcji nadrzędnego stanowiska pracy. W hierarchiach ról jest wykorzystywana koncepcja dziedziczenia, aby umożliwiać domyślne włączanie do danej roli praw dostępu skojarzonych z rolą podległą.

Na rysunku 4.9 podano przykład diagramu hierarchii ról. Na zasadzie umowy role podległe są umieszczone niżej na diagramie. Linia między dwiema rolami symbolizuje, że górna rola zawiera wszystkie prawa dostępu roli dolnej, jak również inne prawa dostępu nieosiągalne dla roli dolnej. Jedna rola może dziedziczyć prawa dostępu wielu podporządkowanych ról. Na przykład na rysunku 4.9 rola prowadzącego projekt zawiera wszystkie prawa dostępu inżyniera produkcji i inżyniera jakości. Prawa tej samej podporządkowanej roli może dziedziczyć więcej niż jedna rola. Na przykład zarówno rola inżyniera produkcji, jak i rola inżyniera jakości zawiera wszystkie prawa dostępu roli inżyniera. Roli inżyniera produkcji są również przypisane dodatkowe prawa, a inny zbiór dodatkowych praw przysługuje roli inżyniera jakości. Tym samym pod względem praw dostępu obie te role zachodzą na siebie, mianowicie dzielą one prawa dostępu z rolą inżyniera.



Rysunek 4.9. Przykład hierarchii ról

OGRANICZENIA — RBAC₂

Ograniczenia stanowią środki adaptacji RBAC do konkretnych zasad administracji i bezpieczeństwa w danej organizacji. **Ograniczenie** (ang. *constraint*) jest zdefiniowaną zależnością między rolami lub warunkiem dotyczącym ról. [SAND96] wymienia następujące typy ograniczeń: role wzajemnie się wykluczające, licznosc i role z warunkami wstępnymi.

Role wzajemnie się wykluczające to takie, w których użytkownik może być przypisany tylko do jednej roli w zbiorze. To ograniczenie może być statyczne lub dynamiczne w tym sensie, że użytkownikowi można przypisać tylko jedną ze zbioru ról na czas sesji. Ograniczenie w myśl zasady wzajemnego wykluczania służy separowaniu obowiązków i zdolności w ramach organizacji. Wyseparowanie takie można wzmocnić lub ulepszyć, stosując wzajemnie wykluczające się przydziały pozwoleń. Z tym dodatkowym ograniczeniem zbiór wzajemnie wykluczających się ról ma następujące własności:

1. Użytkownik może być przypisany tylko do jednej roli w zbiorze (na czas sesji lub statycznie).
2. Dowolne pozwolenie (prawo dostępu) może być udzielone tylko jednej roli w zbiorze.

Tak więc zbiór wzajemnie wykluczających się ról ma niezachodzące na siebie pozwolenia. Jeśli dwóch użytkowników jest przypisanych do różnych ról w zbiorze, to użytkownicy ci, przybierając te role, mają niezachodzące na siebie pozwolenia. Celem wzajemnego wykluczania się ról jest zwiększenie trudności zawiązania się komitety między osobnikami o różnych specjalnościach lub odmiennych stanowiskach, mającej przeszkodzić w polityce bezpieczeństwa.

Licznosc (liczebność, ang. *cardinality*) dotyczy ustalenia maksymalnej liczby danych ról. Jedno z takich ograniczeń ma na celu określenie maksymalnej liczby użytkowników, których można przypisać do danej roli. Na przykład rola kierownika projektu lub dyrektora oddziału może być limitowana do jednego użytkownika. System mógłby również wymuszać ograniczenie liczby ról, które można przydzielić jednemu użytkownikowi, lub liczby ról, które użytkownik może uaktywnić w jednej sesji. Innym rodzajem ograniczenia jest ustalenie maksymalnej liczby ról, którym można nadać konkretne pozwolenie. Może to być pożądaną techniką łagodzenia ryzyka w przypadku wrażliwych lub mocnych pozwoleń.

System mógłby też określać **rolę z warunkiem wstępnym** (ang. *prerequisite role*), której przypisanie użytkownikowi wymaga, aby był on już przypisany do innej konkretnej roli. Warunek wstępny można zastosować do strukturalizacji urzeczywistnianej zasady najmniejszych przywilejów. W hierarchii może być wymagane, aby przypisanie użytkownikowi starszej (wyższej) roli było możliwe pod warunkiem, że ma on już przypisaną o jeden szczebel młodszą (niższą) rolę. Na przykład na rysunku 4.9 użytkownik z rolą prowadzącego projekt musi również być przypisany do podporządkowanych ról inżyniera produkcji i inżyniera jakości. Jeżeli wówczas użytkownik nie potrzebuje w danym zadaniu wszystkich pozwoleń roli prowadzącego projekt, może zainicjować sesję, korzystając tylko z wymaganej roli podległej. Zauważmy, że zastosowanie warunków wstępnych związanych z koncepcją hierarchii wymaga modelu RBAC₃.

4.6. KONTROLOWANIE DOSTĘPU WEDŁUG ATRYBUTÓW

Stosunkowo niedawnym osiągnięciem technologicznym w kontrolowaniu dostępu jest model **kontrolowania dostępu według atrybutów** (ang. *attribute-based access control* — ABAC). W modelu ABAC można definiować upoważnienia, które wyrażają warunki dotyczące cech zarówno zasobu, jak i podmiotu. Rozważmy na przykład konfigurację, gdzie każdy zasób ma atrybut identyfikujący podmiot, który utworzył dany zasób. Wówczas jedna reguła dostępu może określić przywilej własności dla wszystkich twórców każdego zasobu. Mocną stroną podejścia ABAC jest jego elastyczność i siła wyrazu. [PLAT13] podkreśla, że główną przeszkodą w jego adaptacji w rzeczywistych systemach był problem wpływu na sprawność obliczania przy każdym dostępie predykatów właściwości zarówno zasobów, jak i użytkownika. Jednak w zastosowaniach takich jak usługi współpracujące w Sieci i obliczenia chmurowe ten zwiększony koszt wydajności jest mniej zauważalny, gdyż i tak każdy dostęp jest obłożony stosunkowo wysokim kosztem działania. Zatem usługi Sieci były pionierskimi technologiami w implementowaniu modeli ABAC, zwłaszcza przez wprowadzenia języka XAMCL (ang. *eXtensible Access Control Markup Language* — XACML, z ang. rozszerzalny język adjustacji kontroli dostępu) [BEUC13], toteż utrzymuje się znaczne zainteresowanie stosowaniem modelu ABAC w usługach chmurowych [IQBA12, YANG12].

W modelu ABAC istnieją trzy główne elementy: atrybuty, definiowane w odniesieniu do jednostek konfiguracji, model polityki, który określa zasady ABAC, i model architektoniczny, odnoszący się do zasad wymuszania kontrolowania dostępu. Omówimy te elementy kolejno.

Atrybuty

Atrybuty charakteryzują specyficzne aspekty podmiotu, obiektu, warunków środowiskowych i (lub) zamawianych operacji, które są z góry zdefiniowane i przydzielone przez uprawniony organ. Każdy atrybut zawiera dane wskazujące klasę podawanych przez niego informacji, nazwę i wartość (np. `Class = HospitalRecordsAccess, Name = PatientInformationAccess, Value = MFBusinessHoursOnly`⁹).

Poniżej podajemy trzy rodzaje atrybutów w modelu ABAC:

- **Atrybuty podmiotu.** Podmiot jest jednostką aktywną (np. użytkownikiem, aplikacją, procesem lub urządzeniem), która powoduje przepływ informacji między obiektami lub zmiany w stanie systemu. Z każdym podmiotem są związane atrybuty, które definiują tożsamość i cechy podmiotu. Takie atrybuty mogą zawierać identyfikator podmiotu, jego nazwę, nazwę firmy, stanowiska pracy itd. Rolę podmiotu również można uważać za atrybut.

⁹ Odpowiednio z ang.: klasa = dostęp do rekordów szpitala, nazwa = dostęp do informacji o pacjencie, wartość = tylko w godzinach MF biznesu (MF, tu zapewne: *Mutual Fund* — inwestycyjne fundusze wzajemne) — *przyp. tłum.*

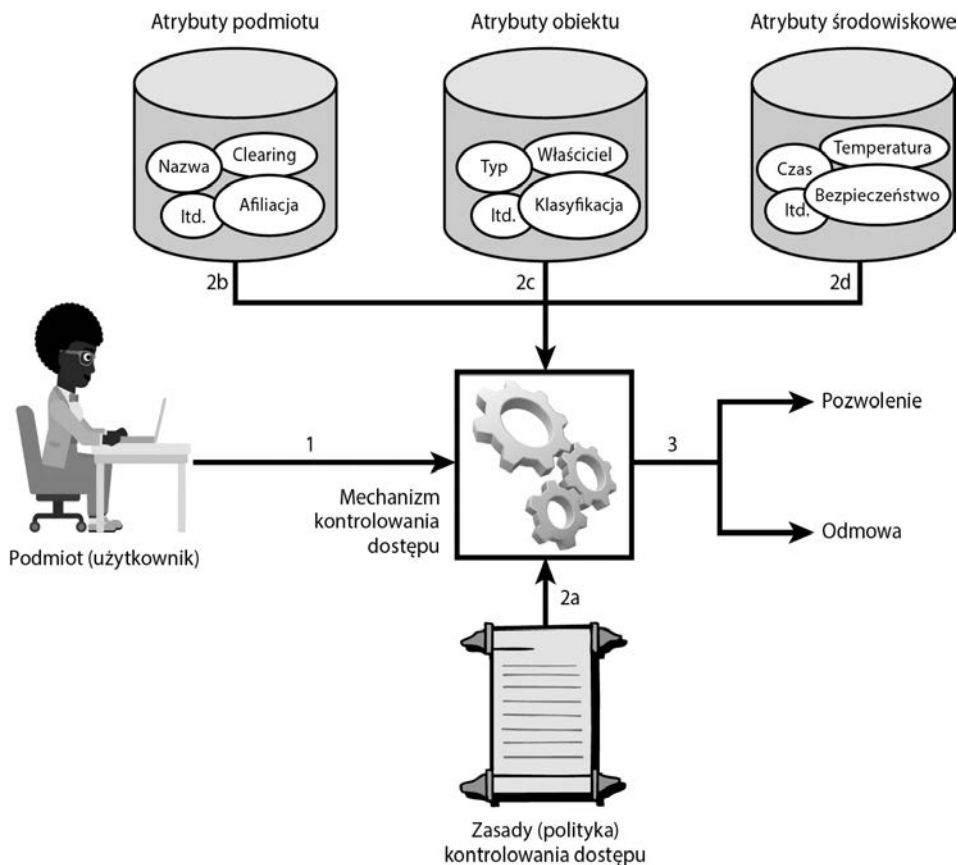
- **Atrybuty obiektu.** Obiekt, nazywany również **zasobem**, jest pasywną (w kontekście danego żądania) jednostką związaną z systemem informacyjnym (obiektami mogą być na przykład urządzenia, pliki, rekordy, tabele, procesy¹⁰, programy, sieci, domeny), zawierającą lub odbierającą informacje. Obiekty, podobnie jak podmioty, mają atrybuty, które można uwzględniać w podejmowaniu decyzji kontrolowania dostępu. Na przykład dokument microsoftowego edytora Word może mieć takie atrybuty jak tytuł, temat, datę i autora. Atrybuty obiektów często można wydobywać z metadanych obiektu. W szczególności do kontrolowania dostępu mogą być potrzebne różne atrybuty zawarte w metadanych usług Sieci, w tym takie jak własność, taksonomia usługi czy nawet atrybut **jakości obsługi** (ang. *quality of service* — QoS).
- **Atrybuty środowiskowe** (ang. *environmental attributes*). Te atrybuty były do tej pory na ogół ignorowane w większości zasad kontrolowania dostępu. Opisują one operacyjne, techniczne, a nawet sytuacyjne środowisko lub kontekst, w którym następuje dostęp do informacji. Na przykład atrybuty takie jak bieżąca data i czas, bieżąca aktywność wirusa lub hakera lub poziom bezpieczeństwa sieciowego (np. Internet czy intranet) nie są kojarzone z konkretnym podmiotem ani zasobem, lecz mimo to mogą mieć znaczenie w stosowaniu polityki kontrolowania dostępu.

ABAC jest logicznym modelem kontrolowania dostępu wyróżniającym się tym, że kontroluje dostęp do obiektów przez zestawianie i ocenę reguł z atrybutami jednostek (podmiotu i obiektu), operacjami i środowiskiem właściwym danemu zamówieniu. ABAC zasadza się na ocenie atrybutów podmiotu, atrybutów obiektu oraz na formalnej zależności lub regule kontrolowania dostępu definiującej dopuszczalne operacje dla kombinacji atrybutów podmiot-obiekt w danym środowisku. Wszystkie rozwiązania ABAC zawierają te podstawowe, rdzenne zdolności oceny atrybutów i wymuszają reguły lub zależności między tymi atrybutami. Systemy ABAC potrafią wymuszać koncepcje DAC, RBAC i MAC. ABAC umożliwia drobnoziarnistą kontrolę dostępu, w której przy podejmowaniu decyzji można uwzględniać większą liczbę dyskretnych danych wejściowych, co zwiększa zasób ich możliwych kombinacji. Dzięki temu zbiór możliwych reguł, zasad lub ograniczeń systemu staje się większy i bardziej wyrazisty. ABAC dopuszcza zatem łączenie nieograniczonej liczby atrybutów warunkujących spełnienie dowolnej reguły kontrolowania dostępu. Co więcej, za pomocą systemów ABAC można urzeczywistniać szeroki asortyment wymagań, zaczynając od podstawowych list kontroli dostępu aż po zaawansowane i wyraziste modele polityki w pełni wykorzystujące elastyczność, którą ABAC posiada.

Architektura logiczna ABAC

Na rysunku 4.10 przedstawiono architekturę logiczną zasadniczych komponentów systemu ABAC. Dostęp podmiotu do obiektu odbywa się w następujących krokach:

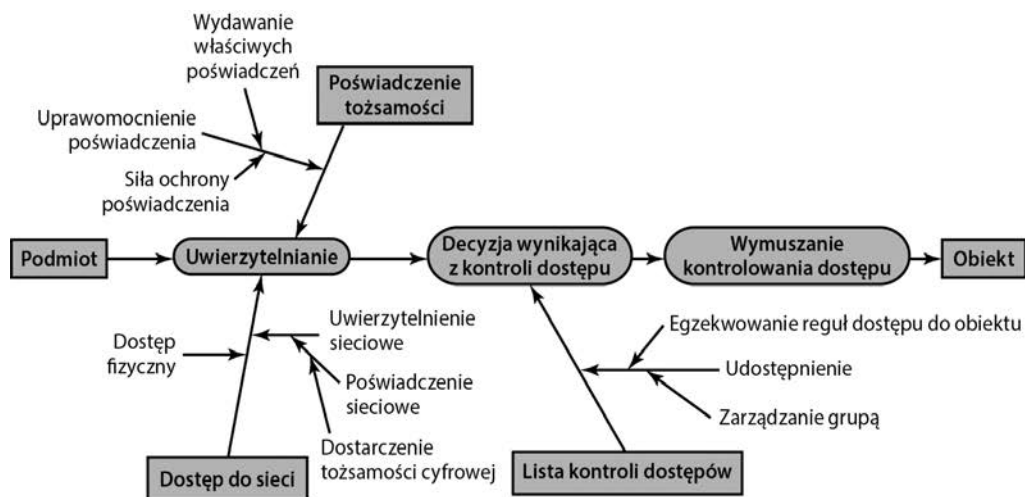
¹⁰ W ich statycznym aspekcie — *przyp. tłum.*



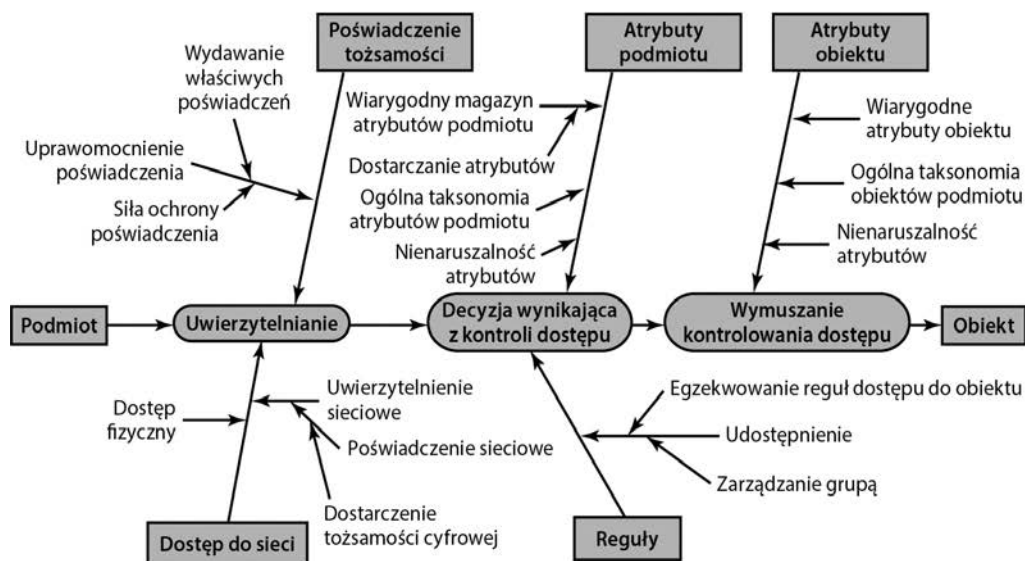
Rysunek 4.10. Scenariusz ABAC

1. Podmiot żąda dostępu do obiektu. To żądanie jest kierowane do mechanizmu kontrolowania dostępu.
2. Mechanizmem kontrolowania dostępu rządzi zbiór reguł, które: 2a) są zdefiniowane na podstawie zawczasu ukształtowanych zasad kontroli dostępu. Opierając się na tych regułach, mechanizm kontrolowania dostępu ocenia atrybuty: 2b) podmiotu, 2c) obiektu i 2d) bieżące warunki środowiska, aby rozstrzygnąć o upoważnieniu.
3. Mechanizm kontrolowania dostępu zezwala podmiotowi na dostęp do obiektu, jeśli dostęp jest prawomocny, a odmawia dostępu, gdy podmiot nie ma upoważnienia.

Łatwo dostrzec w tej architekturze logicznej, że do podejmowania decyzji związanych z kontrolowaniem dostępu są tutaj wykorzystywane cztery niezależne źródła informacji. Projektant systemu może rozstrzygnąć, które atrybuty są ważne z punktu widzenia kontrolowania dostępu w odniesieniu do podmiotów, obiektów i warunków środowiskowych. Projektant systemu lub inny decydent może wówczas zdefiniować zasady kontrolowania dostępu (w postaci reguł) dla każdej pożądanej kombinacji atrybutów podmiotu, obiektu i warunków środowiskowych. Powinno być jasne, że jest to podejście bardzo



(a) Łańcuch zaufania ACL



(b) Łańcuch zaufania ABAC

Rysunek 4.11. Relacje zaufania ACL i ABAC

silne i elastyczne. Jednak koszt, zarówno pod względem złożoności projektu i implementacji, jak i wpływu na wydajność, prawdopodobnie przewyższy koszty innych metod kontrolowania dostępu. To jest kompromis, który kierownictwo systemu musi wziąć pod uwagę.

Na rysunku 4.11, pochodzącym z dokumentu NIST SP 800-162 [*Guide to Attribute Based Access Control (ABAC) Definition and Considerations*, z ang. „Przewodnik po definicji i problematyce kontrolowania dostępu opartego na atrybutach”] ze stycznia 2014 roku, przedstawiono wygodny sposób ogarnięcia zakresu modelu ABAC porównanego z modelem DAC używającym list kontroli dostępu (ACLs). Rysunek ukazuje nie tylko względną złożoność obu modeli, lecz również wyjaśnia wymagania odnoszące się w obu modelach do kwestii zaufania. Z porównania reprezentatywnych relacji zaufania (ukazanych za pomocą strzałek) w obu przypadkach (użycia ACL i użycia ABAC) wynika, że do poprawnego działania model ABAC wymaga znacznie więcej złożonych relacji zaufania. Pomijając rzeczy wspólne w obu częściach rysunku 4.11, można zauważyć, że w wypadku list ACL punktem wyjścia zaufania jest właściciel obiektu ostatecznie wymuszający reguły dostępu do obiektu, umożliwiając dostęp do niego przez dodanie użytkownika do listy ACL. W ABAC „korzeń” zaufania jest wyprowadzany z wielu źródeł, na które właściciel obiektu nie ma wpływu, takich jak Subject Attribute Authorities, Policy Developers i Credential Issuers¹¹. Zgodnie z tym dokument SP 800-162 rekomenduje, aby na szczeblu kierowniczym przedsięwzięcia powstał zespół posiadający umiejętność wdrażania i wykorzystania całokształtu możliwości związanych z tożsamością, poświadczeniami i administrowaniem dostępem, a każda organizacja podporządkowana utrzymywała podobne ciało do zapewniania spójności w zarządzaniu rozwojem i zmianami paradygmatu realizacji modelu ABAC w skali zakładu. Ponadto jest zalecane, aby przedsiębiorstwo (organizacja) wypracowało model zaufania usprawniający obrazowanie relacji zaufania, ustalanie własności informacji i usług oraz odpowiedzialności za nie, pomocny w identyfikowaniu konieczności wprowadzania dodatkowych reguł polityki i rządzenia oraz określaniu technicznych wymagań oceny lub egzekwowania relacji zaufania. Wykorzystanie modelu zaufania może pomóc organizacjom wpływać na to, aby dzielenie się ich informacjami odbywało się z zachowaniem jasności co do sposobu, w jaki te informacje będą użytkowane i chronione, oraz nabrać zaufania do informacji, atrybutów i deklarowanych upoważnień przychodzących z innych organizacji.

Polityka ABAC

Polityka (zasady postępowania, ang. *policy*) jest zbiorem reguł rządzących dopuszczalnym zachowaniem wewnątrz organizacji, opartych na przywilejach podmiotów oraz sposobach ochrony zasobów lub obiektów z uwzględnieniem warunków środowiska. Z kolei **przywileje** (ang. *privileges*) reprezentują upoważnione zachowania podmiotu. Są one określone przez kierownictwo i urzeczywistniane przez politykę. Na określenie przywilejów powszechnie używa się innych terminów, takich jak **prawa** (ang. *rights*), **upoważnienia** (ang. *authorizations*) i **kompetencje** (tytuły, ang. *entitlements*). Zasady polityki są na ogół pisane z perspektywy obiektu wymagającego ochrony i przywilejów, którymi dysponują podmioty.

¹¹ Odpowiednio z ang.: organy rozstrzygające o atrybutach podmiotu, twórcy zasad postępowania i wydawcy poświadczeń (pełnomocnictw) — *przyp. tłum.*

Zdefiniujemy teraz model polityki ABAC oparty na modelu przedstawionym w [YUAN05]. Przyjmujemy następujące konwencje:

1. S , O i E są — odpowiednio — podmiotami, obiektami i środowiskami.
2. SA_k ($1 \leq k \leq K$), OA_m ($1 \leq m \leq M$) i EA_n ($1 \leq n \leq N$) są z góry ustalonymi atrybutami podmiotów, obiektów i środowisk (odpowiednio).
3. $ATTR(s)$, $ATTR(o)$ i $ATTR(e)$ są relacjami przypisania atrybutu do podmiotu s , obiektu o i środowiska e (odpowiednio):

$$ATTR(s) \subseteq SA_1 \times SA_2 \times \dots \times SA_K$$

$$ATTR(o) \subseteq OA_1 \times OA_2 \times \dots \times OA_M$$

$$ATTR(e) \subseteq EA_1 \times EA_2 \times \dots \times EA_N$$

Do określenia wartości przypisań poszczególnych atrybutów korzystamy również z notacji funkcyjnej. Przykłady:

$$Role(s) = \text{"Konsument usługi"}$$

$$WłaścicielUsługi(o) = \text{"XYZ, Inc."}$$

$$BieżącaData(e) = \text{"2018-05-07"}$$

4. W najogólniejszej postaci **reguła polityki** (ang. *policy rule*) decydująca o tym, czy podmiot s może uzyskać dostęp do obiektu o w konkretnym środowisku e , jest funkcją boolowską atrybutów s , o i c :

$$\text{Rule: can_access}(s, o, c) \leftarrow f(ATTR(s), ATTR(o), ATTR(e))$$

Jeśli z uwzględnieniem wszystkich przypisań atrybutów s , o i c wartość funkcji po obliczeniu jest prawdziwa, dostęp do zasobu zostaje udzielony, w przeciwnym razie następuje odmowa dostępu.

5. Baza reguł albo magazyn polityki może zawierać pewną liczbę reguł polityki dotyczących wielu podmiotów i obiektów w domenie bezpieczeństwa. Proces decyzyjny kontrolowania dostępu sprowadza się w istocie do oceny (obliczania) stosowalnych reguł polityki w magazynie polityki.

Przyjrzyjmy się teraz przykładowi witryny internetowej (magazynu rozrywek online), która w ramach miesięcznego abonamentu przekazuje użytkownikom strumienie filmów. Użyjemy tego przykładu do skonstruowania podejść RBAC i ABAC. Witryna musi egzekwować następującą politykę kontrolowania dostępu uwzględniającą wiek użytkownika i kwalifikację treści filmu:

Kwalifikacja filmu	Użytkownicy uprawnieni do dostępu
R	Wiek 17 lat lub więcej
PG-13	Wiek 13 lat lub więcej
G	Wszyscy

W modelu RBAC każdy użytkownik miałby przypisaną jedną z trzech ról: dorosły, nieletni, dziecko — być może podczas rejestracji. Utworzone zostałyby trzy pozwolenia: może

oglądać filmy o kwalifikacji R, może oglądać filmy o kwalifikacji PG-13, może oglądać filmy o kwalifikacji G. Roli dorosłego zostają przypisane wszystkie trzy pozwolenia, rola nieletniego dostaje pozwolenia „może oglądać filmy o kwalifikacji PG-13” i „może oglądać filmy o kwalifikacji G”, a rola dziecka tylko pozwolenie „może oglądać filmy o kwalifikacji G”. Zarówno przypisanie użytkownika do roli, jak i pozwolenia do roli są zadaniami administracyjnymi do ręcznego wykonania.

W podejściu ABAC do tej aplikacji nie trzeba jawnie definiować ról. Zamiast tego decyzja o tym, czy użytkownik u może uzyskać dostęp do oglądania filmu m (w środowisku bezpieczeństwa e , które tutaj pomijamy), zapadnie na podstawie obliczenia reguły polityki następującego rodzaju:

$$\begin{aligned} R1: \text{can_access}(u, m, e) \leftarrow \\ & (\text{Wiek}(u) = 17 \wedge \text{Kwalifikacja}(m) \in \{R, PG-13, G\}) \vee \\ & (\text{Wiek}(u) = 13 \wedge \text{Wiek}(u) < 17 \wedge \text{Kwalifikacja}(m) \in \{PG-13, G\}) \vee \\ & (\text{Wiek}(u) < 13 \wedge \text{Kwalifikacja}(m) \in \{G\}) \end{aligned}$$

gdzie Wiek jest atrybutem podmiotu, a Kwalifikacja jest atrybutem obiektu. Ukazana tutaj zaleta modelu ABAC polega na wyeliminowaniu definiowania statycznych ról i konieczności zarządzania nimi, a co za tym idzie — na wyeliminowaniu konieczności wykonywania zadań administracyjnych związanych z przypisywaniem użytkownika do roli i pozwolenia do roli.

Zaleta modelu ABAC uwidoczni się wyraźniej, gdy narzucimy bardziej szczegółową politykę. Przypuśćmy, że filmy są sklasyfikowane w kategoriach „nowość” lub „starość”, opartych na dacie ukazania się filmu porównywanej z datą bieżącą, a użytkownicy są zaliczeni do klas „premium” i „zwykły” według wysokości opłacanego abonamentu. Chcielibyśmy egzekwować zasady, że tylko użytkownicy klasy „premium” mogą oglądać nowe filmy. W modelu RBAC musielibyśmy podwoić liczbę ról, aby rozróżniać każdego użytkownika pod względem wieku i opłaty, i musielibyśmy również podwoić liczbę osobnych pozwoleń.

Uogólniając, jeśli mamy K atrybutów podmiotów i M atrybutów obiektów i jeśli dla każdego atrybutu $\text{Zakres}()$ oznacza przedział wartości możliwych do przyjęcia, to odpowiednia liczba ról i pozwoleń wymaganych w modelu RBAC wyniesie:

$$\prod_{k=1}^K \text{Zakres}(SA_k) \text{ i } \prod_{m=1}^M \text{Zakres}(SA_m)$$

Widzimy więc, że wraz ze wzrostem liczby atrybutów mającym na celu uszczegółowienie polityki liczba ról i pozwoleń rośnie wykładniczo. Natomiast w modelu ABAC dodatkowe atrybuty dochodzą w sposób ekonomiczny. W danym przykładzie zdefiniowana poprzednio polityka R1 zachowuje ważność. Potrzebujemy dwóch nowych ról:

$$\begin{aligned} R2: \text{can_access}(u, m, e) \leftarrow \\ & (\text{TypCzłonkostwa}(u) = \text{Premium}) \vee \\ & (\text{TypCzłonkostwa}(u) = \text{Zwykły} \wedge \text{TypFilmu}(m) = \text{Starość}) \\ R3: \text{can_access}(u, m, e) \leftarrow R1 \wedge R2 \end{aligned}$$

W modelu ABAC można też łatwo dodawać atrybuty środowiskowe. Załóżmy, że chcemy dodać nową regułę polityki, wyrażoną słownie w taki sposób: *Zwykłym użytkownikom wolno oglądać nowości w okresach promocyjnych*. Trudno byłoby to wyrazić w modelu RBAC. W modelu ABAC musimy tylko dodać w koniunkcji (I) regułę, która sprawdza, czy atrybut środowiskowy *dzisiejsza data* mieści się w okresie promocyjnym.

4.7. TOŻSAMOŚĆ, POŚWIADCZENIA I ZARZĄDZANIE DOSTĘPEM

Przeanalizujemy obecnie pewne pojęcia niezbędne w kontrolowaniu dostępu skoncentrowanym na atrybutach. W tym podrozdziale podajemy przegląd koncepcji tożsamości, poświadczeń i zarządzania dostępem (ICAM¹²), a w podrozdziale 4.8 omówimy zastosowanie ramy zaufania do wymiany atrybutów.

ICAM stanowi ogólne podejście do zarządzania tożsamościami cyfrowymi (i skojarzonymi z nimi atrybutami), poświadczeniami i kontrolowaniem dostępu oraz do ich implementowania. ICAM powstał na zlecenie rządu USA, lecz znajduje zastosowanie nie tylko w agencjach rządowych — można go wdrażać w przedsiębiorstwach wymagających ujednoliconego podejścia do kontrolowania dostępu. Projektowaniu ICAM-u przyświecały następujące cele:

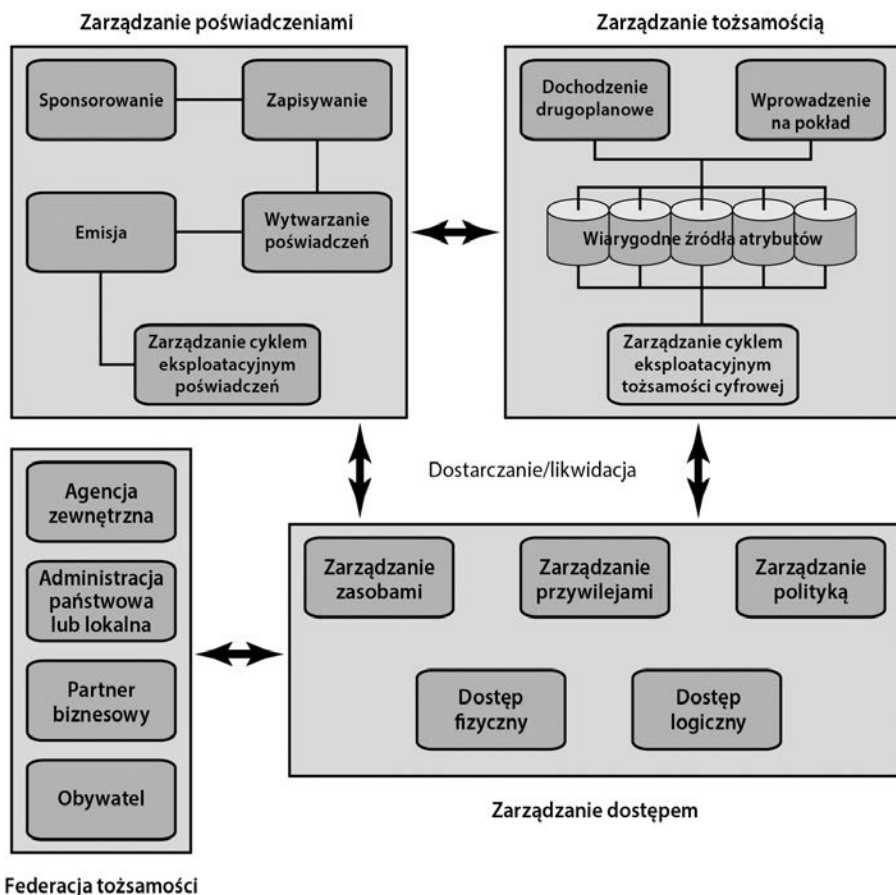
- Tworzenie godnych zaufania reprezentacji tożsamości cyfrowej osób i tego, co jest określane w dokumentacji ICAM-u jako **jednostki nieosobowe** (ang. *nonperson entities* — NPEs). Ostatnie określenie dotyczy procesów, aplikacji i zautomatyzowanych urządzeń potrzebujących dostępu do jakiegoś zasobu.
- Wiązanie tych tożsamości z poświadczeniami, które mogą służyć jako pośrednictwo dla poszczególnych NPE w dostępie do transakcji. **Poświadczenie** (ang. *credential*) stanowi obiekt lub strukturę danych, która w sposób wiarygodny wiąże tożsamość (i opcjonalnie dodatkowe atrybuty) z żetonem posiadanym i kontrolowanym przez abonenta (subskrybenta, sygnatariusza).
- Posługiwanie się poświadczeniami do udostępniania w sposób upoważniony zasobów danej placówki (agencji).

Na rysunku 4.12 dokonano przeglądu składowych logicznych architektury ICAM. W kolejnych punktach omówimy każdą z tych głównych składowych.

Zarządzanie tożsamością

Zarządzanie tożsamością dotyczy wyposażania tożsamości cyfrowej w atrybuty i łączenia tej tożsamości cyfrowej z osobą lub NPE. Celem jest osiągnięcie wiarygodności tożsamości cyfrowej w sposób niezależny od konkretnej aplikacji lub kontekstu. Podejście

¹² Skróć pochodzi od pierwszych liter angielskich odpowiedników tych nazw: *identity*, *credential*, *access management* — *przyp. tłum.*



Rysunek 4.12. Tożsamość, poświadczenia i zarządzanie dostępem (ICAM)

tradycyjne — i wciąż bardzo rozpowszechnione — czyli kontrolowanie dostępu do aplikacji i programów, polega na tworzeniu cyfrowej reprezentacji tożsamości do konkretnego użycia aplikacji lub programu. W rezultacie utrzymywanie i ochrona samej tożsamości są traktowane drugorzędnie wobec zadania związanego z aplikacją. Co więcej, często wysiłki związane ze stanowieniem tych swoistych dla aplikacji jednostek w dużym stopniu nakładają się na siebie.

W odróżnieniu od kont używanych do logowania się w sieciach, systemach lub aplikacjach rekordy (zapisy) tożsamości odnoszące się do całego przedsięwzięcia nie są związane ze stanowiskiem pracy, obowiązkami służbowymi, umiejscowieniem lub tym, czy potrzebny jest dostęp do konkretnego systemu. Jednostki te mogą stawać się atrybutami kojarzonymi z rekordem takiej **tożsamości inicjatywnej** (ang. *enterprise identity*) i mogą również być częścią czegoś, co jednoznacznie identyfikuje indywiduum w danej aplikacji. Decyzje dotyczące kontrolowania dostępu będą zależały od kontekstu i istotnych atrybutów użytkowników, a nie wyłącznie od ich tożsamości. Koncepcja tożsamości inicja-

tywnej ma tę właściwość, że poszczególne osoby będą miały jedną cyfrową reprezentację, która może służyć w różnych oddziałach i agencjach do wielu celów, w tym do kontrolowania dostępu.

Na rysunku 4.12 przedstawiono podstawowe funkcje występujące w zarządzaniu tożsamością. Zbudowanie **tożsamości cyfrowej** (ang. *digital identity*) zwykle zaczyna się od zebrania danych identyfikujących, co jest częścią procesu **zapisywania** (naboru, ang. *enrollment*). Tożsamość cyfrowa jest często tworzona ze zbioru atrybutów, które razem wzięte jednoznacznie identyfikują użytkownika w obrębie systemu lub przedsięwzięcia. Aby zapewnić wiarygodność osobie reprezentowanej przez tożsamość cyfrową, agencja może również przeprowadzić dochodzenie w tle. Atrybuty dotyczące danej osoby mogą być przechowywane w różnych autorytatywnych źródłach agencji i łączone w celu utworzenia obrazu tożsamości cyfrowej w ramach danego przedsięwzięcia. Tę tożsamość cyfrową można potem przedkładać aplikacjom do pomocy w uzyskiwaniu fizycznego lub logicznego dostępu (jako część „zarządzanie dostępem”) i wycofywać, gdy dostęp przestaje być potrzebny.

Finalnym elementem zarządzania tożsamością jest administrowanie cyklem eksploatacyjnym („cyklem życia”), na który składa się, co następuje:

- mechanizm, zasady i procedury ochrony osobistych informacji dotyczących tożsamości;
- kontrolowanie dostępu do danych dotyczących tożsamości;
- metody dzielenia uwiarygodnionych danych o tożsamości z potrzebującymi tego aplikacjami;
- wycofywanie tożsamości inicjatywnej.

Zarządzanie poświadczeniami

Jak wspomniano, **poświadczenie** (ang. *credential*) jest obiektem lub strukturą danych, która autorytatywnie wiąże jednostkę (i opcjonalnie dodatkowe atrybuty) z żetonem posiadanym przez użytkownika i pozostającym pod jego kontrolą. Przykładami poświadczeń są karty inteligentne, prywatne lub publiczne klucze kryptograficzne oraz certyfikaty cyfrowe. Przez zarządzanie poświadczeniami rozumie się zarządzanie ich cyklem eksploatacyjnym. Zarządzanie poświadczeniami obejmuje pięć następujących komponentów logicznych:

1. Upoważniona osoba występuje w sprawie osoby lub jednostki ubiegającej się o poświadczenie, uzasadniając jego potrzebę. Na przykład szef oddziału „sponsoruje” w ten sposób pracownika oddziału.
2. Sponsorowana osoba zapisuje się po poświadczenie — proces taki na ogół składa się z udowodnienia tożsamości i pobrania danych biograficznych i biometrycznych. Ten krok może także zawierać dołączenie autorytatywnych (wiarygodnych) danych atrybutowych, utrzymywanych przez komponent zarządzania tożsamością.

3. Następuje wytworzenie poświadczenia. Zależnie od rodzaju poświadczenia wytworzenie może zawierać szyfrowanie, użycie podpisu cyfrowego, wyprodukowanie karty inteligentnej lub inne czynności.
4. Poświadczenie zostaje wydane osobie lub NPE (jednostce nieosobowej).
5. Na koniec poświadczenie musi być utrzymywane przez czas posługiwania się nim (cykl eksploatacyjny), co może obejmować wycofywanie, wznowienie, ponowne wystąpienie o zapisanie, upłynięcie terminu ważności, nadanie nowego osobistego numeru identyfikacyjnego (PIN-u), zawieszenie lub przywrócenie.

Zarządzanie dostępem

Składowa zarządzania dostępem odpowiada za zarządzanie i nadzór nad sposobami, za pomocą których jednostki zyskują dostęp do zasobów. Obejmuje ona zarówno dostęp logiczny, jak i fizyczny i może być wewnętrzną częścią systemu lub elementem zewnętrznym. Celem zarządzania dostępem jest zapewnienie, że gdy dana osoba próbuje wejść do wrażliwych pod względem bezpieczeństwa budynków, systemów komputerowych lub sięgnąć po poufne dane, nastąpi właściwa weryfikacja jej tożsamości. Funkcja kontrolowania dostępu wykorzystuje poświadczenia przedstawiane przez ubiegających się o dostęp i cyfrową tożsamość pretendenta. Do kontrolowania dostępu w skali całego przedsięwzięcia są potrzebne trzy pomocnicze elementy:

- **Zarządzanie zasobami.** Ten element jest skoncentrowany na definiowaniu reguł dotyczących zasobu, który wymaga kontrolowania dostępu. Reguły mogą uwzględniać przedkładanie poświadczeń i wszelkich atrybutów użytkownika, atrybutów zasobów i warunków środowiskowych wymaganych do dostępu do danego zasobu w celu wykonania określonego działania.
- **Zarządzanie przywilejami.** Ten element odpowiada za ustanowienie i utrzymywanie kompetencji (tytułów) lub atrybutów przywilejów składających się na profil dostępu osoby. Atrybuty te reprezentują cechy osoby, które mogą służyć jako podstawa do podejmowania decyzji w kwestii dostępu zarówno do zasobów logicznych, jak i fizycznych. Przywileje są uważane za atrybuty, które można połączyć z tożsamością cyfrową.
- **Zarządzanie polityką.** Ten element rządzi tym, co jest dozwolone lub niedozwolone w transakcji dostępu. To znaczy mając tożsamość i atrybuty zamawiającego, atrybuty zasobu lub obiektu oraz warunki środowiskowe, polityka określa, jakie czynności dany użytkownik może wykonać na danym obiekcie.

Federacja tożsamości

Federacja tożsamości (zjednoczenie tożsamości, ang. *identity federation*) uwzględnia dwie kwestie:

1. Na jakiej podstawie ufasz tożsamościom osób z zewnętrznych organizacji, które chcą uzyskać dostęp do twoich systemów?

2. Jak poręczysz za tożsamości osób z twojej organizacji, gdy przyjdzie im współpracować z zewnętrznymi organizacjami?

Federacja tożsamości jest terminem używanym do opisu technologii, standardów, polityk i procesów, które umożliwiają organizacji pokładanie zaufania w tożsamościach cyfrowych, atrybutach tożsamości i poświadczeniach tworzonych i wydawanych przez inne organizacje. Federacje tożsamości omówimy w następnym podrozdziale.

4.8. RAMY ZAUFANIA

Powiązane ze sobą koncepcje zaufania, tożsamości i atrybutów stały się podstawowymi problemami przedsiębiorczości internetowej, dostawców usług sieciowych i dużych inicjatyw. Zagadnienia te można wyraźnie obserwować w organizacji e-handlu. Aby uzyskać skuteczność, prywatność i prostotę prawną, strony transakcji na ogół stosują zasadę **wiedzy koniecznej** (ang. *need to know*): co musisz wiedzieć o kimś, żeby z nim robić interesy¹³? Odpowiedź zmienia się od przypadku do przypadku i zawiera takie atrybuty jak rejestracja potwierdzająca kwalifikacje zawodowe lub numer licencji, przedsiębiorstwo i dział, ID personelu, certyfikat bezpieczeństwa, numer referencyjny klienta, numer karty kredytowej, jednoznaczny numer karty zdrowia, uczulenia, grupa krwi, numer ubezpieczenia społecznego, adres, status obywatelski (zameldowanie), kontakt z siecią społecznościową, pseudonim itd. Zależnie od sytuacji atrybuty jednostki muszą być znane i sprawdzone, aby można było pozwolić na transakcję.

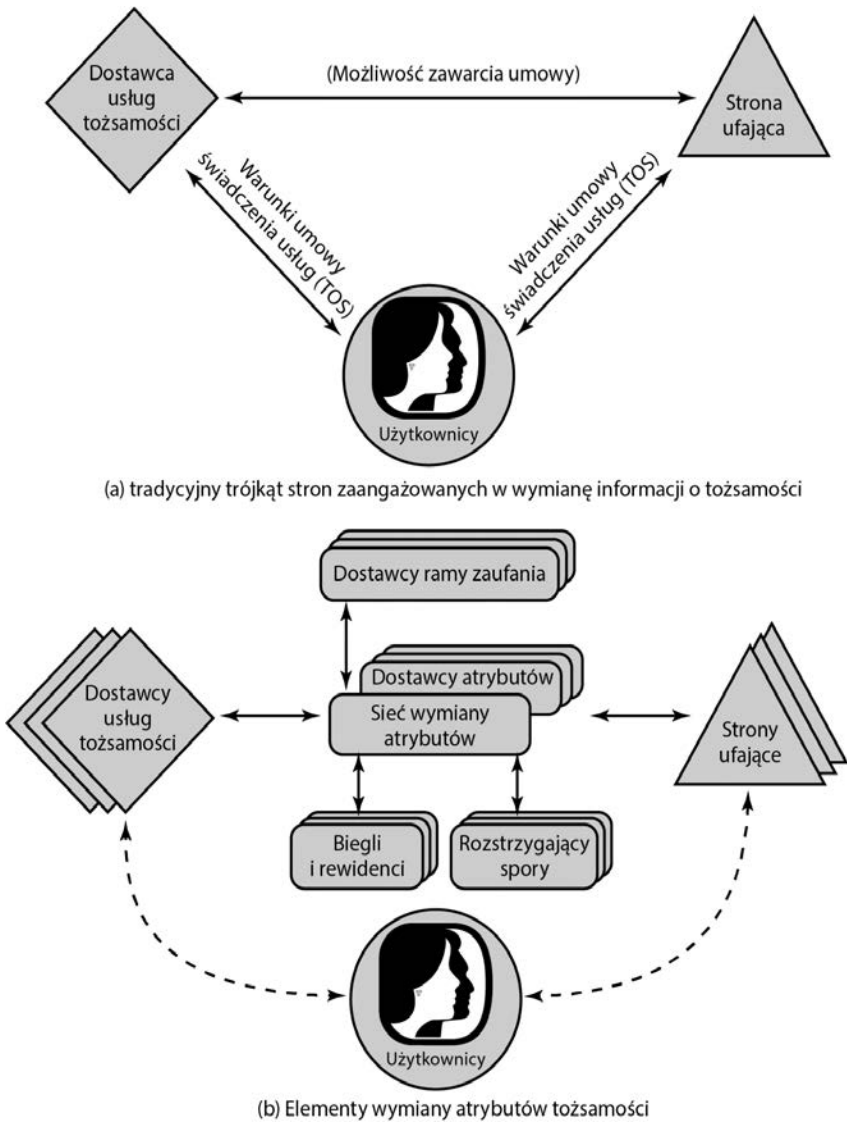
Troska o atrybuty nabiera coraz większego znaczenia we wszystkich rodzajach sytuacji kontrolowania dostępu, nie tylko w e-handlu. Na przykład dane przedsiębiorstwo może potrzebować dostępu do zasobów klientów, użytkowników, dostawców i partnerów. Zależnie od kontekstu dostęp będzie zdeterminowany nie tylko przez tożsamość, lecz także przez atrybuty zamawiającego i zasobu.

Tradycyjne podejście do wymiany tożsamości

Transakcje bezpośrednie lub sieciowe dotyczące przedstawicieli różnych firm lub dokonywane między firmą a indywidualnym użytkownikiem, na przykład kupującym online, z reguły wymagają dzielenia informacji dotyczących tożsamości. Prócz prostej nazwy lub numeru identyfikacyjnego te informacje mogą zawierać mnóstwo powiązanych atrybutów. Zarówno strona ujawniająca takie informacje, jak i strona, która je otrzymuje, muszą mieć pewien poziom zaufania co do kwestii bezpieczeństwa i prywatności związanych z tymi informacjami.

Na rysunku 4.13a pokazano tradycyjną technikę wymiany informacji dotyczących tożsamości. Występują tu użytkownicy nawiązujący kontakt z **dostawcą usług tożsamości**

¹³ W innych źródłach zasada wiedzy koniecznej jest synonimem tego, co w tej książce jest nazywane zasadą najmniejszych przywilejów — *przyj. tłum.*



Rysunek 4.13. Podejścia do wymiany informacji dotyczących tożsamości

(ang. *identity service provider*), aby uzyskać tożsamość cyfrową i poświadczenia, oraz kontakty z podmiotami dostarczającymi docelowe usługi i aplikacje. Wszyscy oni chcą polegać na tożsamości i poświadczonych informacjach wygenerowanych przez dostawcę usług tożsamości.

W sytuacji przedstawionej na rysunku 4.13a musi być spełnionych kilka wymagań. **Strona ufająca** (strona powiernicza, ang. *relying party*) wymaga, aby użytkownik został uwierzytelniony z pewnym stopniem pewności, że atrybuty podsuwane mu przez dostawcę usług tożsamości są rzetelne, oraz że dostawca usług tożsamości jest kompetentny (wiarogodny) w kwestii tych atrybutów. Dostawca usług tożsamości wymaga zapewnienia,

że informacje, jakie posiada o użytkowniku, są rzetelne, oraz że gdy podzieli się nimi ze stroną ufającą, użyje ich ona zgodnie z zasadami i warunkami umowy oraz przepisami prawa. Użytkownik chce mieć pewność, że dostawcy usług tożsamości i stronie ufającej można powierzyć wrażliwe informacje i że będą oni przestrzegać jego preferencji i prywatności. Co najważniejsze, wszystkie strony chcą wiedzieć, czy praktyki opisane przez inne strony są naprawdę przez nie wdrożone i do jakiego stopnia strony te są wiarygodne.

Rama zaufania otwartej tożsamości

Bez jakiegoś uniwersalnego standardu i ramowego ujęcia (ramy, ang. *framework*) układ przedstawiony na rysunku 4.13a trzeba powtarzać w wielu kontekstach. Znacznie lepsze jest podejście polegające na opracowaniu otwartej, standardowej metody osiągania wiarygodności wymiany tożsamości i atrybutów. W pozostałej części tego podrozdziału przyjrzymy się takiemu podejściu, spotykającemu się z rosnącą akceptacją.

Niestety, ten temat ugina się pod ciężarem licznych akronimów, najlepiej więc będzie, jeśli rozpoczniemy od definicji najważniejszych z nich.

- **OpenID** (z ang. otwarty identyfikator). Jest to otwarty (ogólnodostępny) standard umożliwiający uwierzytelnianie użytkowników przez pewne kooperujące witryny (nazywane stronami ufającymi, ang. *relying parties*), korzystające z usług stron trzecich, co uwalnia opiekunów witryn („webmasterów”) od dostarczania własnych, budowanych ad hoc systemów i umożliwia użytkownikom konsolidację ich cyfrowych tożsamości. Użytkownicy mogą tworzyć konta z wykorzystaniem OpenID ich ulubionych dostawców tożsamości, a potem korzystać z tych kont jako baz do zapisywania się w dowolnej witrynie sieciowej, która akceptuje uwierzytelnianie z OpenID.
- **OIDF**. *OpenID Foundation* jest międzynarodową, niedochodową organizacją skupiającą osoby indywidualne i firmy dążące do stosowania, promowania i ochrony technologii OpenID. OIDF pomaga społeczności, zaopatrując ją w niezbędną infrastrukturę i udzielając pomocy w promowaniu i wspieraniu rozwiniętej adaptacji OpenID.
- **ICF**. *Information Card Foundation* (z ang. Fundacja Kart Informacyjnych) jest niedochodową wspólnotą firm i osób indywidualnych współpracujących nad rozwojem ekosystemu Karty Informacyjnej. Karty informacyjne są osobistymi tożsamościami cyfrowymi, których można używać online, i podstawowym komponentem metasysemów tożsamości. Z wyglądu każda karta informacyjna przypomina kształtem kartę i jest zaopatrzona w nazwę, co umożliwia ludziom organizowanie ich cyfrowych tożsamości i łatwe wybieranie którejś z nich, potrzebnej w danej interakcji.
- **OITF**. *Open Identity Trust Framework* (z ang. Rama Zaufania Otwartej Tożsamości) jest ustandaryzowaną, otwartą specyfikacją rama zaufania do wymiany tożsamości i atrybutów, opracowaną wspólnie przez OIDF i ICF.

- **OIX.** *Open Identity Exchange Corporation* (z ang. Korporacja Wymiany Otwartej Tożsamości) jest niezależnym, neutralnym, międzynarodowym dostawcą zaufanych ram (platform) certyfikacji spełniających wymagania modelu *Open Identity Trust Framework*.
- **AXN.** *Attribute Exchange Network* (z ang. Sieć Wymiany Atrybutów) jest bramą online o skali internetowej dla dostawców usług tożsamości i stron ufających, usprawniającą masowy dostęp online po umiarkowanych kosztach do zadeklarowanych przez użytkownika, dozwolonych i zweryfikowanych atrybutów tożsamości.

Zarządcy systemów muszą mieć podstawy do zaufania, że atrybuty skojarzone z obiektem lub sam obiekt są wiarygodne i wymieniane bezpiecznie. Jednym z rozwiązań zapewniających to zaufanie w ramach organizacji jest model ICAM, a w szczególności komponenty ICAM (rysunek 4.12). W połączeniu z funkcjonalnością federacji tożsamości, dzieloną z innymi organizacjami, atrybuty można wymieniać w sposób godny zaufania, co umożliwia bezpieczne kontrolowanie dostępu.

W systemach tożsamości cyfrowej **rama zaufania** (ang. *trust framework*) działa jak program certyfikacji. Umożliwia temu, kto akceptuje poświadczenie cyfrowej tożsamości (zwanemu stroną ufającą), darzenie zaufaniem co do tożsamości, bezpieczeństwa i zasad prywatności tego, kto wydaje poświadczenie (nazywanego dostawcą usług tożsamości) i na odwrót. Nieco formalizując, OIX definiuje ramę zaufania jako zbiór weryfikowalnych zobowiązań poczynionych przez każdą z różnych stron transakcji względem ich partnerów. Te ustalenia zawierają (1) uregulowania (w tym regulacje urzędowe i zobowiązania umowne) mające zapewnić, że zobowiązania będą wypełniane i (2) sposoby reagowania na niespełnienie takich zobowiązań. Rama zaufania jest budowana przez społeczność, której członkowie mają podobne cele i perspektywy. Określa prawa i obowiązki członków społeczności, specyficzne dla danej społeczności zasady i standardy oraz definiuje specyficzne dla danej społeczności procesy i procedury materializowania pewności. Mogą istnieć różne ramy zaufania, a poszczególne grupy uczestników mogą kształtować ramy zaufania tak, aby spełniały ich konkretne potrzeby.

Rysunek 4.13b ukazuje elementy występujące w OITF. W danej organizacji lub agencji częściami ogólnej ramy są następujące role:

- **Strony ufające** (strony powiernicze, ang. *relying parties* — RPs), nazywane również dostawcami usług. Są to jednostki świadczące usługi poszczególnym użytkownikom. RP muszą mieć pewność co do tożsamości i (lub) atrybutów swoich planowanych użytkowników i muszą polegać na różnych poświadczeniach przedstawianych w celu uprawnocnienia tych atrybutów i tożsamości.
- **Podmioty.** Są to użytkownicy usług RP, w tym klienci, pracownicy, partnerzy handlowi i abonenci.
- **Dostawcy atrybutów** (ang. *attribute providers* — APs). AP są jednostkami, którym wspólnota interesów powierzyła rolę i które wyposażała w zdolność weryfikowania atrybutów przedstawianych przez podmioty; zostały one również wyposażone przez

AXN w możliwość tworzenia zgodnych z regułami i ustaleniami AXN poświadczeń atrybutów. Niektórzy dostawcy atrybutów będą źródłami uwiarygodniania pewnych informacji, częściej jednak będą pośrednikami pochodnych atrybutów.

- **Dostawcy tożsamości** (ang. *identity providers* — IDPs). Są to jednostki zdolne do uwierzytelniania poświadczeń użytkowników oraz do poręczania nazw (lub pseudonimów, lub kontaktów do) podmiotów, wyposażone przez AXN lub jakiś inny kompatybilny system zarządzania tożsamościami i dostępem (ang. *Identity and Access Management* — IDAM) w możliwości tworzenia tożsamości cyfrowych, które mogą być używane do indeksowania atrybutów użytkownika.

Istnieją również następujące ważne elementy pomocnicze, będące częścią AXN:

- **Biegli** (ang. *assessors*). Biegli oceniają („ewaluują”) dostawców usług tożsamości i strony ufające (RP) oraz certyfikują ich zdolność postępowania według strategii dostawców IOTF.
- **Rewidenci** (audytorzy, ang. *auditors*). Te jednostki mogą być powoływane do sprawdzania, że uprawiane przez strony praktyki zachowują zgodność z ustaleniami OITF.
- **Rozwiązujący spory** (ang. *dispute resolvers*). Te jednostki pełnią role arbitrażowe i rozstrzygają spory stosownie do zaleceń OIX.
- **Dostawcy ram zaufania** (ang. *trust framework providers*). Dostawca ramy zaufania jest organizacją, która przekłada wymagania twórców polityki (zasad) na własną strategię ramy zaufania, a następnie wdraża ją w sposób spójny z minimalnymi wymaganiami przyjętymi w specyfikacji OITF. Prawie we wszystkich przypadkach znajdzie się organizacja w widoczny i rozsądny sposób kandydująca do przyjęcia tej roli w każdym sektorze przemysłu lub w dużej instytucji, która uzna, że jest odpowiednia do współpracy z AXN.

Strzałki poprowadzone ciągłymi liniami na rysunku 4.13b symbolizują porozumienia z dostawcą ramy zaufania w kwestii realizacji wymagań technicznych, operacyjnych i prawnych. Strzałki poprowadzone liniami przerywanymi wskazują inne uzgodnienia, potencjalnie wynikające z tych wymagań. W ogólnym zarysie model przedstawiony na rysunku 4.13b działałby następująco. Osoby odpowiedzialne w organizacjach uczestniczących ustalają techniczne, operacyjne i prawne wymagania dotyczące wymiany informacji tożsamości, leżące w zakresie ich kompetencji. Następnie wybierają dostawców OITF do realizacji tych wymagań. Dostawcy OITF tłumaczą wymagania na strategię ramy zaufania, która może zawierać dodatkowe warunki dostawcy OITF. Dostawca OITF sprawdza dostawców usług tożsamości i strony ufające (RP) i umawia się z nimi co do przestrzegania wymagań jego ramy zaufania podczas dokonywania wymiany informacji tożsamości. Te umowy dostarczają rozstrzygającym spory i rewidentom (audytorom) materiał do interpretacji umów i ich umacniania.

4.9. PRZYKŁAD KONKRETNY: KONTROLOWANIE RÓL W SYSTEMIE BANKOWYM

Bank Drezdeński¹⁴ ma zaimplementowany system RBAC, który służy jak pożyteczny przykład praktyczny [SCHA01]. Bank używa rozmaitych aplikacji komputerowych. Wiele z nich było na początku opracowanych do środowiska mainframe; niektóre z tych starszych aplikacji są obecnie dostępne w sieci klient-serwer, inne pozostają na komputerach głównych. Istnieją również nowsze aplikacje na serwerach. Przed rokiem 1990 na każdym serwerze i komputerze głównym eksploatowano prosty system DAC. Administratorzy sprawowali lokalną kontrolę nad dostępem do plików na każdym hoście¹⁵ z osobną i definiowali prawa dostępu dla każdego pracownika w odniesieniu do każdej aplikacji na każdym hoście. Taki system był nieporęczny, czasochłonny i podatny na błędy. Aby go ulepszyć, bank wprowadził w skali całego systemu schemat RBAC, w którym dla większego bezpieczeństwa określanie praw dostępu rozczłonkowano w trzy jednostki administracyjne.

Role wewnątrz tej instytucji są zdefiniowane przez połączenie stanowiska i obowiązków służbowych. W tabeli 4.5a podano przykłady. Różni się to nieco od pojęcia roli w standardzie NIST, w którym rolę określa stanowisko pracy. Do pewnego stopnia różnica ta jest kwestią terminologii. Tak czy owak strukturalizacja ról bankowych tworzy w sposób naturalny środki budowy hierarchii dziedziczenia opartej na stanowiskach służbowych. W obrębie banku, w każdej jego jednostce organizacyjnej, występuje ściśle częściowe uporządkowanie stanowisk służbowych, odzwierciedlające hierarchię odpowiedzialności i znaczenia. Na przykład stanowiska naczelnika oddziału, kierownika grupy i urzędnika (kasjera) mają porządek malejący. W efekcie połączenia stanowiska służbowego z zakresem obowiązków następuje uporządkowanie praw dostępu, co uwidoczniło w tabeli 4.5b. Tak więc rola analityka finansowego i kierownika grupy (rola B) ma więcej praw dostępu niż rola analityka finansowego i urzędnika (rola A). W tabeli pokazano, że rola B ma przynajmniej tyle (lub więcej) praw dostępu, co rola A w trzech aplikacjach, ma też prawa dostępu do czwartej aplikacji. Z drugiej strony nie ma zależności hierarchicznej między rolą bankowości biurowej a kierownikiem grupy a rolą analityka finansowego i urzędnika, gdyż obszary ich funkcjonowania są różne. Możemy więc zdefiniować hierarchię ról, w której jedna rola jest nadrzędna względem innej, jeśli przypisane jej stanowisko jest starsze, a funkcje są identyczne. Hierarchia ról umożliwia ekonomizację definicji praw dostępu, co zasugerowano w tabeli 4.5c.

W oryginalnym schemacie bezpośrednie przypisanie praw dostępu indywidualnemu użytkownikowi występowało na poziomie aplikacji i dotyczyło pojedynczej aplikacji. W nowym schemacie administracja aplikacji ustala zbiór praw dostępu związanych z każdą

¹⁴ Obecnie Commerzbank — *przyp. tłum.*

¹⁵ Wieloznaczność tego angielskiego terminu w zastosowaniach informatycznych: komputer sieciowy, macierzysty, goszczący bywa wygodna, dlatego używamy go w wielu miejscach w tym przykładzie; poza tym jest krótki — *przyp. tłum.*

indywidualną aplikacją. Jednak dany użytkownik wykonujący dane zadanie nie musi mieć wszystkich praw dostępu związanych z aplikacją. Gdy użytkownik wywołuje aplikację, udziela ona dostępu na podstawie dostarczanego centralnie profilu bezpieczeństwa. Osobna administracja upoważnień łączy prawa dostępu z rolami i tworzy profil bezpieczeństwa do użycia na podstawie roli użytkownika.

Użytkownik jest przypisany do roli statycznie. W zasadzie (w tym przykładzie) każdy użytkownik może mieć statycznie przypisanych do czterech ról i może wybrać daną rolę do wykorzystania podczas rozpoczynania konkretnej aplikacji. Odpowiada to pojęciu sesji NIST. W praktyce większość użytkowników ma statycznie przypisaną jedną rolę według stanowiska i zakresu obowiązków użytkownika.

Tabela 4.5. Funkcje i role w przykładzie bankowym

(a) Funkcje i stanowiska służbowe

Rola	Funkcja	Stanowisko służbowe
A	Analitik finansowy	Urzędnik
B	Analitik finansowy	Kierownik grupy
C	Analitik finansowy	Naczelnik oddziału
D	Analitik finansowy	Junior
E	Analitik finansowy	Senior
F	Analitik finansowy	Specjalista
G	Analitik finansowy	Asystent
...	...	...
X	Technik udziałów	Urzędnik
Y	Pomoc e-handlu	Junior
Z	Bankowość biurowa	Naczelnik oddziału

(b) Przydział pozwoleń

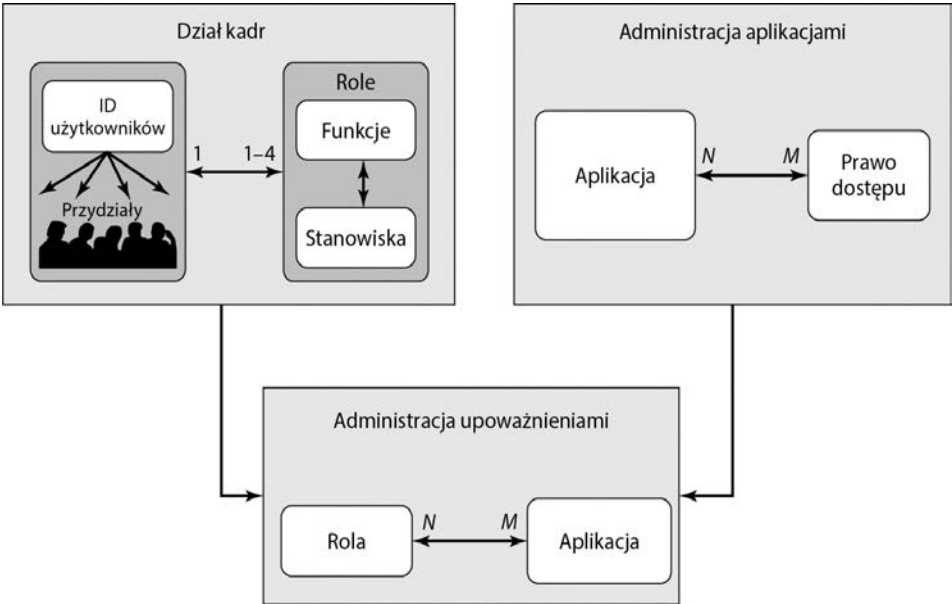
Rola	Aplikacja	Prawa dostępu
A	Instrumenty rynku pieniężnego	1, 2, 3, 4
	Handel instrumentami pochodnymi	1, 2, 3, 7, 10, 12
	Instrumenty odsetkowe	1, 4, 8, 12, 14, 16
B	Instrumenty rynku pieniężnego	1, 2, 3, 4,7
	Handel instrumentami pochodnymi	1, 2, 3, 7, 10, 12, 14
	Instrumenty odsetkowe	1, 4, 8, 12, 14, 16
	Instrumenty prywatnego konsumenta	1, 2, 4, 7
...	...	...

Tabela 4.5. Funkcje i role w przykładzie bankowym — *ciąg dalszy*

(c) Przydział pozwoleń z dziedziczeniem

Rola	Aplikacja	Prawa dostępu
A	Instrumenty rynku pieniężnego	1, 2, 3, 4
	Handel instrumentami pochodnymi	1, 2, 3, 7, 10, 12
	Instrumenty odsetkowe	1, 4, 8, 12, 14, 16
B	Instrumenty rynku pieniężnego	7
	Handel instrumentami pochodnymi	14
	Instrumenty prywatnego konsumenta	1, 2, 4, 7
...	...	...

Wszystkie te składniki są przedstawione na rysunku 4.14. Dział kadr przydziela jednoznaczny ID użytkownika każdemu pracownikowi, który będzie korzystał z systemu. Na podstawie stanowiska służbowego użytkownika i zakresu jego obowiązków dział kadr przydziela również użytkownikowi jedną lub więcej ról. Informacja o powiązaniu użytkownika z rolą trafia do administracji upoważnień, która tworzy profil bezpieczeństwa każdego użytkownika, kojarzący ID użytkownika i rolę ze zbiorem praw dostępu. Gdy użytkownik wywołuje aplikację, ta sprawdza jego profil bezpieczeństwa, aby ustalić, jaki podzbiór praw dostępu do aplikacji przysługuje mu w tej roli.



Rysunek 4.14. Przykład administrowania kontrolowaniem dostępu

Rola może służyć do dostępu do kilku aplikacji. Zatem zbiór praw związanych z rolą może zawierać prawa dostępu niezwiązane z którąś z aplikacji wywoływanych przez użytkownika. Jest to przedstawione w tabeli 4.5b. Rola A ma wiele praw dostępu, lecz tylko podzbiór tych praw jest przydatny w każdej z trzech aplikacji, które można wywoływać w roli A.

Na uwagę zasługują niektóre dane dotyczące tego systemu. W banku występuje 65 stanowisk służbowych, poczynając od urzędnika oddziału poprzez kierownika oddziału po członka zarządu. Te stanowiska są łączone z 368 różnymi zakresami obowiązków (funkcjami) zawartymi w bazie danych działu kadr. Potencjalnie istnieje 23 920 różnych ról, lecz liczba aktualnie używanych ról wynosi około 1300. Jest to zgodne z doświadczeniami zebranymi w innych implementacjach RBAC. Średnio każdego dnia moduł administracji upoważnień dystrybuje między aplikacje 42 000 profili bezpieczeństwa.

4.10. PODSTAWOWE POJĘCIA, PYTANIA SPRAWDZAJĄCE I ZADANIA

Podstawowe pojęcia

Atrybut	Lista kontroli dostępu	Rama zaufania
Atrybut obiektu	Macierz dostępu	Rewident (audytor)
Atrybut podmiotu	Najmniejsze przywileje	Rola z warunkiem wstępnym
Atrybut środowiska	Obiekt	Rozwiązujący spory
Attribute Exchange Network (z ang. Sieć Wymiany Atrybutów — AXN)	Obligatoryjne kontrolowanie dostępu (MAC)	Separacja obowiązków
Biegły	Ogólna hierarchia ról	Sesja
Bilet uprawnień (mandat zdolności)	Ograniczenia roli	Statyczna separacja obowiązków (SSD)
Domena ochrony	Ograniczona hierarchia ról	Strona ufająca
Dostawca atrybutów	Open Identity Exchange	Tożsamość
Dostawca ramy zaufania	Corporation (z ang. Korporacja Wymiany Otwartej Tożsamości — OIX)	Tożsamość, poświadczenia i zarządzanie dostępem (ICAM)
Dostawca tożsamości	Open Identity Trust Framework (z ang. Rama Zaufania Otwartej Tożsamości — OITF)	Tryb jądra
Dynamiczna separacja przywilejów (DSD)	OpenID Foundation (OIDF)	Tryb użytkownika
Federacja tożsamości	OpenID (z ang. otwarty identyfikator)	Upoważnienia (autoryzacje)
Grupa	Otwarta polityka kontrolowania dostępu	Uznaniowe kontrolowanie dostępu (DAC)
Hierarchie ról	Podmiot	Właściciel
Information Card Foundation (z ang. Fundacja Kart Informacyjnych — ICF)	Polityka (zasady postępowania)	Wzajemnie wykluczające się role
Kompetencje (tytuły)	Poświadczenie	Zamknięta polityka kontrolowania dostępu
Kontrolowanie dostępu	Pozwolenie	Zarządzanie dostępem
Kontrolowanie dostępu według atrybutów (ABAC)		Zarządzanie poświadczeniami
		Zarządzanie tożsamością

Kontrolowanie dostępu według ról (RBAC)	Prawa	Zasób
Liczność (liczebność)	Prawo dostępu	
	Przywilej	

Pytania sprawdzające

- 4.1. Zdefiniuj krótko różnice między DAC i RBAC.
- 4.2. Jak się ma RBAC do DAC i MAC?
- 4.3. Wymień i określ trzy klasy podmiotów w systemie kontrolowania dostępu.
- 4.4. Na czym polega różnica między podmiotem a obiektem w kontekście kontrolowania dostępu?
- 4.5. Co to jest prawo dostępu?
- 4.6. Co różni listę kontroli dostępu od biletu uprawnień?
- 4.7. Co to jest domena ochrony?
- 4.8. Zdefiniuj krótko cztery modele RBAC z rysunku 4.8a.
- 4.9. Wymień i zdefiniuj cztery rodzaje jednostek w podstawowym modelu systemu RBAC.
- 4.10. Opisz trzy rodzaje ograniczeń hierarchii ról.
- 4.11. Co różni SSD od DSD w modelu NIST RBAC?

Zadania

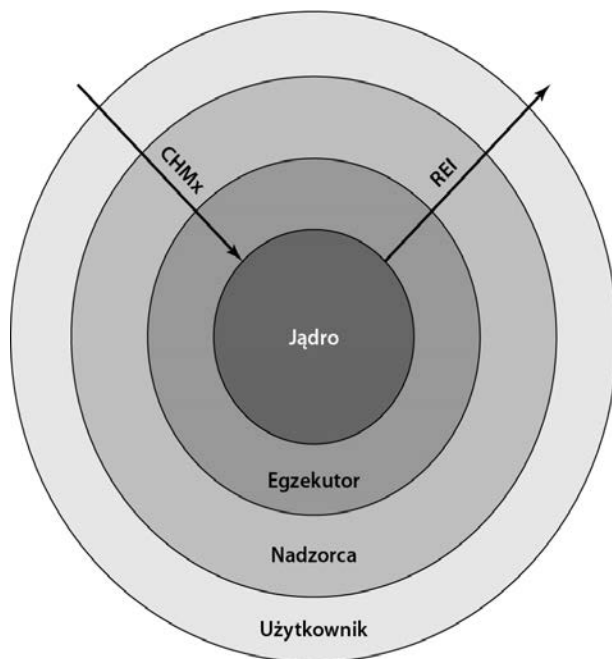
- 4.1. Alternatywną reprezentacją modelu DAC omówionego w podrozdziale 4.3 jest graf skierowany. Każdy podmiot i każdy obiekt w stanie ochrony jest przedstawiony jako węzeł (do oznaczenia jednostki będącej zarówno podmiotem, jak i obiektem używa się pojedynczego węzła). Skierowana krawędź biegnąca od podmiotu do obiektu symbolizuje prawo dostępu, a etykieta, którą jest opatrzona, określa to prawo.
 - a) Naszkicuj graf skierowany odpowiadający macierzy dostępu z rysunku 4.2a.
 - b) Naszkicuj graf skierowany odpowiadający macierzy dostępu z rysunku 4.3.
 - c) Czy między reprezentacją w postaci grafu skierowanego a reprezentacją w postaci macierzy dostępu występuje odpowiedniość jeden do jednego? Wyjaśnij to.
- 4.2.
 - a) Zaproponuj sposób realizacji domen ochrony za pomocą list kontroli dostępu.
 - b) Zaproponuj sposób realizacji domen ochrony za pomocą biletów uprawnień.

Wskazówka. W obu przypadkach jest wymagany pewien poziom pośredniości.
- 4.3. System operacyjny VAX/VMS wykorzystuje cztery tryby dostępu do procesora, aby zapewnić ochronę i dzielenie zasobów systemowych między procesy. Tryb dostępu określa:
 - **Przywileje wykonywania rozkazów.** Jakie rozkazy wolno wykonywać procesorowi.
 - **Przywileje dostępu do pamięci.** Do których komórek pamięci wirtualnej wolno mieć dostęp w bieżącym rozkazie.Cztery tryby to:
 - **Jądro.** Wykonywanie jądra systemu operacyjnego VAX/VMS, co obejmuje zarządzanie pamięcią, obsługę przerw i operacje wejścia-wyjścia.
 - **Egzekutor.** Wykonywanie wielu usług systemu operacyjnego, w tym procedur zarządzania plikami i rekordami (dyskowymi i taśmowymi).

- **Nadzorca.** Wykonywanie innych usług systemu operacyjnego, na przykład odpowiadanie na polecenia użytkownika.
- **Użytkownik.** Wykonywanie programów użytkownika oraz programów narzędziowych, takich jak kompilatory, edytory, konsolidatory i debugery.

Proces działający w mniej uprzywilejowanym trybie często potrzebuje wywołać procedurę działającą w bardziej uprzywilejowanym trybie, na przykład program użytkownika wymaga usługi systemu operacyjnego. To wywołanie jest osiągane za pomocą rozkazu zmiany trybu (CHM), który powoduje przerwanie przekazujące sterowanie do podprogramu w nowym trybie dostępu. Powrót jest wykonywany przez wykonanie rozkazu REI (powrotu z obsługi wyjątku lub przerwania).

- Niektóre systemy operacyjne mają dwa tryby działania: jądra i użytkownika. Jakie są zalety i wady występowania czterech trybów zamiast dwóch?
 - Czy znajdujesz uzasadnienie użycia więcej niż czterech trybów działania?
- 4.4. Omówiony w poprzednim zadaniu schemat VMS jest często określany jako struktura ochrony pierścieniowej, co przedstawiono na rysunku 4.15. Prosty schemat jądro-użytkownik jest w istocie strukturą dwupierścieniową. Wadą systemu kontrolowania dostępu o strukturze pierścieniowej jest to, że narusza on zasadę najmniejszych przywilejów. Jeśli na przykład chcemy mieć obiekt dostępny w pierścieniu X, lecz nie w pierścieniu Y, będzie to wymagało, aby $X < Y$. W tej sytuacji wszystkie obiekty dostępne w pierścieniu X są również dostępne w pierścieniu Y.



Rysunek 4.15. Tryby dostępu VAX/VMS

- a) Wyjaśnij bardziej szczegółowo, na czym polega problem i dlaczego zasada najmniejszych przywilejów zostaje naruszona.
 - b) Zaproponuj sposób, za pomocą którego system operacyjny o strukturze pierścieniowej mógłby poradzić sobie z tym problemem.
- 4.5. UNIX traktuje katalogi plików na równi z plikami. To znaczy i jedno, i drugie są definiowane za pomocą struktury danych tego samego typu, zwanej i-węzłem. Katalogi, tak jak pliki, zawierają 9-bitowy łańcuch ochrony. Jeśli nie zachowa się ostrożności, może to powodować problemy z kontrolowaniem dostępu. Rozważmy na przykład plik z trybem ochrony 644 (ósemkowo), zawarty w katalogu o trybie ochrony 730. Jak może dojść w tym wypadku do naruszenia pliku?
- 4.6. W konwencjonalnym uniksowym modelu dostępu do plików opisanym w podrozdziale 4.4 system UNIX zapewnia nowo tworzącym plikom i katalogom domyślne ustawienia, które mogą być potem zmienione przez właściciela. Domyślnie właścicielowi przysługuje zazwyczaj pełny dostęp w połączeniu z jednym z następujących ustawień: żadnego dostępu dla grupy i innych, czytanie i wykonywanie dla grupy i zero dostępu dla innych lub czytanie i wykonywanie zarówno dla grupy, jak i dla innych. Omów pokrótce zalety i wady każdego z tych przypadków, podając przykład sytuacji, w której każdy z nich byłby odpowiedni.
- 4.7. Rozważmy konta użytkowników systemu z serwerem Sieci¹⁶ skonfigurowanym tak, aby udostępniał obszary Sieci użytkownika. Na ogół używa się tu standardowej nazwy katalogu w rodzaju `public_html` w macierzystym katalogu użytkownika. Jeżeli taki katalog istnieje, jest traktowany jak obszar Sieci użytkownika. Aby jednak pozwolić serwerowi Sieci na dostęp do stron w tym katalogu, serwer musi mieć przynajmniej możliwość przeszukiwania (dostęp *execute*) macierzystego („domowego”) katalogu użytkownika, dostęp do czytania i wykonywania katalogu Sieci i dostęp do czytania każdej z zawartych w nim stron. Zastanówmy się nad oddziaływaniem tego wymogu w przypadkach omówionych przez Ciebie w poprzednim zadaniu. Jakie to będzie miało konsekwencje? Zauważmy, że serwer Sieci zwykle działa jako użytkownik specjalny i w grupie niemającej nic wspólnego z większością użytkowników systemu. Czy istnieją okoliczności, w których wykonywanie takiej usługi Sieci jest po prostu nieodpowiednie? Wyjaśnij.
- 4.8. Załóżmy, że mamy system z N stanowiskami pracy (zakresami obowiązków służbowych). Dla stanowiska i liczba indywidualnych użytkowników na tym stanowisku wynosi U_i , a liczba pozwoleń wymaganych na danym stanowisku równa się P_i .
- a) Ile związków między użytkownikami a pozwoleniami należy określić w tradycyjnym schemacie DAC?
 - b) Ile związków między użytkownikami a pozwoleniami trzeba określić w schemacie RBAC?
- 4.9. Standard NIST RBAC określa ograniczoną hierarchię ról jako taką, w której rola może mieć jednego lub więcej przodków (ascendentów), lecz tylko jednego potomka (descendenta). Które związki dziedziczenia na rysunku 4.10 są zakazane przez standard NIST ograniczonej hierarchii ról?

¹⁶ Przypominamy, że Siecią (pisaną dużą literą) określamy w tekście usługę WWW (ang. *the Web*) — *przyp. tłum.*

- 4.10. W standardzie NIST RBAC ogólną hierarchię ról możemy zdefiniować następująco: $RH \subseteq \text{ROLE} \times \text{ROLE}$ jest częściowym porządkiem w zbiorze ROLE , zwanym relacją dziedziczenia, zapisywaną symbolem $\geq$, gdzie $r_1 \geq r_2$ tylko wtedy, kiedy wszystkie pozwolenia w r_2 są również pozwoleniami r_1 i wszyscy użytkownicy r_1 są również użytkownikami r_2 . Określ zbiór *prawomocne_pozwolenia*(r_i) jako zbiór wszystkich pozwoleń związanych z rolą r_i . Określ zbiór *upoważnieni_użytkownicy*(r_i) jako zbiór wszystkich użytkowników przypisanych do roli r_i . Poza tym węzeł r_1 jest reprezentowany jako bezpośredni potomek r_2 zapisem $r_1 >> r_2$, jeśli $r_1 \geq r_2$, lecz żadna rola w hierarchii ról nie leży między r_1 a r_2 .
- Korzystając odpowiednio z powyższych definicji, podaj formalną definicję ogólnej hierarchii ról.
 - Podaj formalną definicję ograniczonej hierarchii ról.
- 4.11. W przykładzie z podrozdziału 4.8 użyj notacji *Rola*(x).*Stanowisko* w celu oznaczenia stanowiska skojarzonego z rolą x oraz *Rola*(x).*Funkcja* do oznaczenia funkcji skojarzonej z rolą x .
- Hierarchię ról definiujemy (w tym przykładzie) jako taką, w której jedna rola jest nadrzędna w stosunku do drugiej, jeśli jej stanowisko jest wyższe, a ich funkcje są identyczne. Wyraż ten związek formalnie.
 - Alternatywną hierarchią ról będzie taka, w której rola jest nadrzędna wobec innej, jeżeli jej funkcja ma wyższą rangę niezależnie od stanowiska. Wyraż ten związek formalnie.
- 4.12. Nawiązujemy do przykładu z magazynem rozrywek online z podrozdziału 4.6 z drobnostywną polityką uwzględniającą użytkowników premiiowanych i zwykłych. Wymień wszystkie role i wszystkie przywileje w tym przykładzie potrzebne do zdefiniowania modelu RBAC.

SKOROWIDZ

A

- ABAC, 177, 179, 182
- abonent, 104
- ACL, 179
- administrowanie użytkownikami, 530
- adres IP, 380
- adware, 246
- AES, Advanced Encryption Standard, 67
- agent ataku, 275
- akceptowalność psychologiczna, 47
- aktywa, 26, 38
- aktywator, 572
- aktywiści, 333
- aktywność sieciowa, 380
- alarmy, 357
- algorytmy
 - deszyfrowania, 63, 81
 - kryptograficzne, 585
 - podpisu cyfrowego, 86
 - metodą krzywych eliptycznych, 86
 - RSA, 86
 - szyfrowania, 63, 79
 - symetrycznego blokowego, 64
 - asymetrycznego, 83
- analiza
 - malware'u, 289
 - piaskownicowa, 288
 - ruchu, 41
- analizatory, 337
- antywirusy oparte na sygnaturach, 287
- AP, access point, 352
- aparatura na zapleczu, 95
- aplikacje, 521
 - biometryczne, 130
 - konfigurowanie, 521
- APT, 246
- architektura
 - agenta, 351
 - ataku DDoS, 311
 - chmury obliczeniowej, 553
- DBMS, 205
- logiczna ABAC, 177
- NIST, 554
- Snorta, 367
- systemu bankomatów, 146
- archiwizowanie danych, 524
- assembler, 436
- atak, 33
 - aktywny, 34, 41
 - APT, 249
 - trwały, 249
 - zaawansowany, 249
 - zagrożenia, 249
 - blokowania usług, 310
 - DDoS, 299
 - DNS ze wzmocnieniem, 320
 - fałszywych SYN, 305
 - małymi fragmentami, 385
 - mieszany, 248
 - na hosta, 140
 - na klienta, 140
 - na konkretne konto, 110
 - na poziomie warstwy
 - sieciowej, 356
 - transportowej, 356
 - zastosowań, 356
 - na przepływność, 312
 - na słownik offline, 110
 - odbijający, 315
 - odmowa świadczenia usługi, 142, 298, 325
 - oparty na HTTP, 313
 - oprawiający interfejs użytkownika, 268
 - pasywny, 34, 40
 - polegający na odmowie świadczenia usług, 356
 - poza pasmem, 216
 - przepelniający, 450
 - robakami, 262
 - siłowy, 64
 - skryptowy, 475

atak

- skutkujący odmową usług, 302
- SQLi, 212
 - drogi ataku, 213
 - metody przeciwdziałania, 216
 - metody wykrywania, 217
 - typy ataku, 215
 - z wnioskowaniem, 215
- typu powtórka, 141
- wewnętrzny, 34
- wodopojowy, 267
- wstrzykiwania SQL, 210, 470
- z odbicia, 315
- z przepełnieniem bufora, 417
- z przepełnieniem stosu, 439
- z użyciem trasowania źródłowego, 385
- za pomocą popularnych haseł, 110
- zatapiający, 298, 302, 307
- ze wzmocnieniem, 315, 318
- zewnętrzny, 35

ataki

- drzewa, 50
- powierzchnie, 49
- przeciwdziałanie, 35
- przywrócenie, 35
- zapobieżenie, 35

atrybuty, 176, 207

- obiektu, 177
- podmiotu, 176
- środowiskowe, 177

audyt, 154

autentyczność, 28

autoruter, 246

AXN, 190

B

bankomat, 144

bastion

- inline, 398
- T, 398

bazy danych, 202, 204

- bezpieczeństwo, 202
- kontrolowanie dostępu, 217
- na zapleczu, 211
- relacyjne, 206
- ataki SQLi, 210
- systemy zarządzania, 204

szyfrowanie, 226

wnioskowanie, 223

bezpieczeństwo

- aplikacji, 521
- baz danych, 202
- centrum danych, 231
- chmur, 545, 556, 561
- chmury jako usługa, 565
- funkcji haszowania, 77
- hiperwizora, 539
- infrastruktury zwirtualizowanej, 539
- internetu rzeczy, 545, 576
- komputerowe, 26, 27, 31
- kommunikacji, 578
- obliczeń chmurowych, 560
- oparte na rolach, 582
- oprogramowania, 415, 461
- systemów, 415
- systemów operacyjnych, 511, 514
- uwierzytelniania, 140
- VPN, 396
- w obliczeniach chmurowych, 558
- w systemach bankomatowych, 144
- w systemach Linux i Unix, 524
- w systemie Windows, 529
- wirtualizacji, 532, 537
- zaopatrywania w usługi, 578
- zarządzania danymi, 578
- bezpieczna analiza danych, 583
- bezpieczne
 - biblioteki, 445
 - techniki kodowania, 443
 - ustawienia na wypadek awarii, 46
 - użytkowanie plików tymczasowych, 501

biała lista, 478

biegli, 191

bilety uprawnień, 158

biometryczne uwierzytelnianie, 129, 132

bit niewykonywania, 448

blokada plikowa, 500

błędy programowe, 464

boczne

- drzwi, 280
- wejście, 246

bomba logiczna, 246, 275

botnet, 310

boty, 247, 275

brama, 579
 na poziomie aplikacji, 387
 poziomu układowego, 387

C

centrum danych, 231, 581
 elementy, 231
 model bezpieczeństwa, 234
 certyfikat
 CISSP, 13
 klucza publicznego, 86, 88
 chmura, 576
 hybrydowa, 552
 prywatna, 552
 publiczna, 551
 społecznościowa, 552
 chmurowa informatyzacja przedsięwzięć, 546
 chmury
 bezpieczeństwo, 556, 561
 jako usługa, 565
 o otwartym źródle, 569
 informacje o bezpieczeństwie, 568
 likwidowanie skutków katastrof, 568
 obliczeniowe, 547, 553
 oceny bezpieczeństwa, 568
 ochrona danych, 563
 reagowanie na zdarzenia, 568
 szyfrowanie, 568
 utrzymanie ciągłości działania firmy, 568
 włamania, 568
 zabezpieczanie
 dóbr, 564
 poczty elektronicznej, 567
 sieci, 567
 zagrożenia i przeciwdziałania, 561
 zapobieganie utracie danych, 566
 zarządzanie tożsamością i dostępem, 566
 chrootowe więzienie, 496
 ciasteczka, 214
 ciasteczka SYN, 324
 CISSP, 13
 cyberprzestępcy, 333
 cyfrowy system odpornościowy, 401
 czarna lista, 478
 czeladnicy, 334

czochranie wejścia, 480
 czujnik, 336, 572
 danych, 345
 inline, 352
 NIDS, 353
 pasywny, 352

D

dane, 34, 39, 40
 w stanie spoczynku, 94
 wejściowe użytkownika, 214, 468
 dbałość o bezpieczeństwo, 522
 DBMS, database management system, 203–205
 DDL, data definition language, 204
 DDoS, 298, 299, 311
 DEA, Data Encryption Algorithm, 65
 definicja bezpieczeństwa komputerowego, 26
 DES, data encryption standard, 65
 DML, data manipulation language, 204
 DMZ, demilitarized zone, 393
 dobra, 32
 dogłądanie bezpieczeństwa, 578
 dokładność biometryczna, 133
 dołączanie plików PHP, 474
 domeny ochrony, 164
 DoS, denial of service, 299
 dostawcy
 atrybutów, 190
 ram zaufania, 191
 tożsamości, 191
 usług poświadczających, 104
 dostęp
 do haseł, 118
 do informacji, 224
 do rejestru, 346
 oparty na języku SQL, 218
 zdalny, 527
 dostępność, 27–31, 298
 dotarcie do źródła ataku, 322
 druga odporność dziedziczna, 76
 drzewo ataków, 50, 52
 DSA, digital signature algorithm, 84
 działania ukradkowe, 280
 dzielenie plików, 258

E

ECC, elliptic curve cryptography, 85
 ekonomika mechanizmu, 45
 eksfiltracja danych, 279
 elektroniczna
 książka kodowa, 67
 karta identyfikacyjna, 126
 elementy chmury obliczeniowej, 548
 eskalacja przywilejów, 493

F

falszowanie
 adresów IP, 385
 adresów źródłowych, 303
 SYN, 300, 304
 faza
 rozsiewania, 252
 uśpienia, 252
 wykonania, 252
 wyzwalania, 252
 federacja tożsamości, 186
 filtr Blooma, 121
 filtrowanie pakietów, 383
 firewall, 379
 fizyczne wejście użytkownika, 214
 funkcja
 eID, 128
 ePass, 128
 eSign, 128
 haszowania, 69
 bezpieczeństwo, 75, 77
 jednokierunkowa, 73, 75
 kryptograficzna, 73
 zastosowania, 78
 funkcje
 biblioteczne, 496
 EID, 128
 kontroli, 560
 kontroli dostępu, 162
 funkcjonalne wymagania bezpieczeństwa, 42

G

generator prawdziwych liczb losowych, 93
 gromadzenie danych wywiadowczych, 292
 grupa, 157, 526

H

haker, 332
 HAR, host audit records, 350
 hartowanie systemów operacyjnych, 515
 hasła, 78
 ataki, 110, 115
 filtr Blooma, 121
 generowane, 120
 haszowane, 112
 kontrolowanie pliku dostępu, 118
 sprawdzarka, 121
 strategie wyboru, 119
 uwierzytelnianie użytkownika, 109
 hasłołamacz, 114
 haszowanie MAC z kluczem, 75
 heurystyki, 341
 HIDS
 heurystyczny, 348
 rozproszony, 349
 sygnaturowy, 348
 wykrywający anomalie, 346
 hierarchia ról, 174
 hiperwizor, 533
 typu 1, 534
 typu 2, 534
 honeypoty, 364
 host jako bastion, 389
 HTTP, 313

I

IaaS, infrastructure as a service, 550
 IAS, Information Assurance and Security, 11
 ICF, 189
 ICMP flooding, 308
 identyfikacja, 132
 częstotliwości radiowej, 572
 heurystyczna, 344
 IDS-y, 351
 hostowe, 338
 oparte na hostach, 344
 rozproszone, 338, 349
 sieciowe, 338
 implementacja
 algorytmu, 482
 bezpieczeństwa, 54
 infrastruktura jako usługa, 550
 instalowanie systemu operacyjnego, 516

integralność, 28
 systemu, 27
 interfejs użytkownika, 338
 internet rzeczy, 570
 bezpieczeństwo, 576
 bezpieczeństwo o otwartym źródle, 584
 elementy urządzeń, 572
 kontekst chmury, 573
 rama bezpieczeństwa, 580, 583
 rozwój urządzeń, 571
 wymagania prywatności, 578
 internetowy serwer banku, 51
 interpretacja
 wartości danych, 486
 wejścia programu, 469
 intruzi, 332
 IoT, Internet of Things, 570
 IP Security, 503
 IPS, intrusion prevention system, 398
 IPS-y
 hybrydowe, 401
 oparte na hostach, 399
 oparte na sieci, 401
 rozproszone, 401
 IR, internet rzeczy, 570
 ISO, 56
 ISOC, 56
 ITU-T, 56
 izolacja, 47

J

jakość oprogramowania, 464
 jednokierunkowość, 76
 jednostki
 nieosobowe, 183
 wiedzy IAS, 12, 13
 język
 definiowania danych, 204
 manipulowania danymi, 204
 PHP, 475
 zapytań, 204

K

kanal
 komunikacyjny, 51
 wnioskowania, 223

kanarek, 447
 kanonizacja, 480
 karty
 eID, 128
 identyfikacyjne, 126
 inteligentne, 125
 pamięci, 124
 katalog, 165
 usług, 569
 keylogery, 246, 277
 klasy kontroli, 560
 klient, 228
 klikorwanie, 268
 klucz
 Diffiego-Hellmana, 84
 główny, 206, 207
 obcy, 208
 prywatny, 81
 publiczny, 81
 tajny, 63
 kod
 maszynowy, 485
 powłokowy, 430, 433
 przenośny, 246, 266
 uwierzytelniający komunikatu, 71
 kodowanie defensywne, 216
 komentarz kończący wiersz, 215
 kompetencje, 180
 komplet napastniczy, 246, 248
 konfiguracja
 zapory sieciowej, 394
 aplikacji i usług, 521, 525, 531
 konie trojańskie, 269, 270
 kontekst
 chmury, 574
 IR, 574
 kontener wirtualizacji, 537
 kontenery, 537
 kontrolowanie
 bezpieczeństwa, 532
 dostępu, 151, 530
 do bazy danych, 217
 federacja tożsamości, 186
 listy, 168
 obligatoryjne, 156
 organizacja funkcji, 162
 polecenia systemu, 163
 poszerzona macierz, 162
 ramy zaufania, 187

kontrolowanie

- dostępu
 - uznaniowe, 155, 158
 - w uniksowym systemie plików, 165
 - według atrybutów, 156, 176
 - według ról, 156, 169, 221
 - wymagania bezpieczeństwa, 153
 - wymiana tożsamości, 187
 - zarządzanie dostępem, 186
 - zarządzanie poświadczeniami, 185
 - zarządzanie tożsamością, 183
 - zasady, 154
 - zdalnego, 527
- rol w systemie bankowym, 192
- koń trojański, 141, 247
- koperty cyfrowe, 90
- koszty bezpieczeństwa, 54
- kradzież, 140
 - informacji, 277
 - poświadczeń, 278
 - tożsamości, 278
- kryptoanaliza, 64
- kryptografia krzywych eliptycznych, 85
- kryptosystemy klucza publicznego, 82
- krzywe operacyjne charakterystyk pomiarów biometrycznych, 137

L

liczby

- losowe, 90
- pseudolosowe, 93

liczność, 175

linie, 39

lista kontroli dostępu, 158, 168

Ł

ładunek, 245, 252, 272, 277, 280

łamanie haseł

- podejścia nowoczesne, 117
- podejścia tradycyjne, 115

łańcuchowanie bloków szyfru, 72

M

MAC, message authentication code, 71

macierz

- dostępów, 158
- kontroli dostępu, 162

makrowirus, 246, 253, 256

malware, 243

maskarada, 37, 41

maszyna wirtualna, 284, 532

- zapory sieciowej, 541

mechanizm

- infekcji, 252
- mutacji, 257
- ochrony stosu, 446
- wywołania funkcji, 423

metody

- sygnaturowe, 344
- uczenia maszynowego, 343
- wykrywania anomalii, 356

mgła, 576

- obliczeniowa, 574

międzystanowiskowe ataki skryptowe, 475

mikrokontroler, 572

MiniSec

- jednonadawanie, 585
- rozgłaszanie, 585

miodownice, 364

mistrzowie, 334

model

- bezpieczeństwa centrum danych, 234
- bezpieczeństwa komputerowego, 32
- realizacyjny chmur, 551
- rozchodzenia się robaka, 259
- szyfrowania symetrycznego, 63
- usług chmurowych, 549
- wymiany komunikatów, 362
- wzorcowy RBAC, 172

modularność, 48

moduł

- transakcyjny, 204
- zarządcy plików, 204

modyfikacja komunikatów, 41

monitorowanie elektroniczne, 111

monitory

- wchodzenia, 292
- wychodzenia, 292

możliwość

- rozpoczęcia zdalnej sesji, 258
- zdalnego przesyłania, 258
- zdalnego wykonania, 258

N

nadużycie, 38
 nanoszenie poprawek, 525
 naruszenie
 bezpieczeństwa, 336
 polityki bezpieczeństwa, 53, 356
 NICs, network interface controllers, 232
 nieelastyczność, 227
 nienaruszalność, 28, 30, 146
 danych, 27
 nieoczekiwane usługi aplikacji, 356
 nieprzewidywalność, 92
 nieupoważnione ujawnienie, 35, 36
 niezależność, 92
 niezawodność, 464
 NIST, 56
 niszczenie plików, 498

O

obejście VM, 538
 obiekt, 157
 obliczenia w chmurze, 546
 obrona
 przed przepełnieniami bufora, 442
 w fazie kompilacji, 443
 w fazie wykonania, 447
 obsługa
 wejścia programu, 468
 wyjścia programu, 504
 obstrukcja, 37
 obudowanie, 48
 ocena, 55
 ochrona
 danych, 582
 danych w chmurze, 563
 protokołu internetowego, 582
 przed atakami, 406, 408
 odbicie XSS, 476
 odejście, 478
 odfiltrowywanie ataków, 321
 odgadywanie
 hasła, 110
 poświadczeń użytkownika, 52
 odkrywanie celu, 259
 odmowa świadczenia usług, 41, 141, 298, 321

odporność
 dziedziczna, 76
 na silne kolizje, 76
 odpowiedzialność, 28
 odrabianie strat, 55
 ograniczenie, 175
 OI DF, 189
 OITF, 189
 OIX, 190
 okablowanie
 poziome, 233
 szkieletowe, 233
 opanowywanie zagrożeń, 404
 OpenID, 189
 operacja niepodzielna, 500
 oprogramowanie, 34, 39
 jako usługa, 549
 szpiegujące, 247, 278
 zbójckie, 273
 organ rejestrujący, 104
 organizacje sponsorowane przez państwa, 333
 oszustwo, 36, 37
 otwartość projektu, 46

P

PaaS, platform as a service, 549
 pakiet SYN, 306
 pasywny czujnik NIDS, 353
 pełna mediacja, 46
 pewność, 55
 phishing, 277
 PHP, 475
 piaskownica, 288
 piractwo komputerowe, 332
 pisanie bezpiecznego kodu, 482
 planowanie bezpieczeństwa systemu
 operacyjnego, 514
 platforma jako usługa, 549
 plik
 dostępu do haseł, 118
 miodny, 366
 tymczasowy, 501
 pobranie uboczne, 246, 267
 poczta elektroniczna, 257
 podejścia analityczne, 341
 podmiot, 156, 190
 podpis cyfrowy, 84

podrobienie, 37
 podsumowanie zdarzeń, 361
 polityka, 180, 570
 bezpieczeństwa, 33, 53
 połączenie krzyżowe, 232
 poświadczenie, 104, 183, 185
 potencjalny wpływ, 107
 potrójny DES, 66
 poufność, 28, 146, 582
 danych, 27
 semantyczna, 586
 powierzchnia ataku
 dotyczącego ludzi, 50
 programowego, 49
 sieciowego, 49
 powtórka, 41, 141
 poziomy wywieranych skutków
 niski, 29
 umiarkowany, 29
 wysoki, 29
 pozwolenia, 526
 prawa, 180
 dostępu, 157, 219
 przeniesienia, 164
 pretendent, 104
 proces podpisu cyfrowego, 87
 profile
 charakterystyki biometrycznej, 135
 zachowania intruzów, 339
 programowanie defensywne, 463, 465
 programy spawskie, 247
 projektowanie
 bazy danych, 224
 bezpieczeństwa, 44
 protokoły wezwanie-odpowiedź, 138
 protokół
 aplikacji, 380
 biometryczny, 139
 hasła, 136
 uwierzytelniania, 125
 żetonu, 138
 prywatność, 27
 przechwycenie, 35
 przeciwdziałanie, 33
 rootkitom, 290
 szkodliwemu oprogramowaniu, 285
 przeciwnik, 33
 przejęcie poświadczeń użytkownika, 51

przełączniki górnoregłowe, 232
 przepełnienie, 457
 bufora, 415, 417, 421, 468
 bufora na stosie, 422
 obszaru danych globalnych, 455
 sterty, 452
 stosu, 417, 425, 427, 431
 przywileje, 180, 493
 psucie systemu, 272
 pułapka na ludzi, 233
 punkt
 dławienia, 380
 dostępowy, 352

R

rama zaufania, 187, 190
 ramka stosu, 423, 424
 RBAC, 172
 RDBMS, 202
 reakcja, 55
 na atak, 322
 reguła polityki, 181
 reguły Snorta, 368
 rejestratory klawiaturowe, 278
 rejestrowanie
 alarmów, 357
 naciskanych klawiszy, 140
 zdarzeń, 523
 rejestry x86, 437
 rekonesans, 279
 rekordy audytu, 345
 hosta, 350
 relacja, 207
 relacje zaufania ACL i ABAC, 179
 relacyjna baza danych, 207
 relacyjny język zapytań, 206
 rewidenci, 191
 RFC 4949, 152
 robak Morrisa, 261
 robaki, 247, 259, 357
 stan technologii, 265
 w telefonach komórkowych, 266
 robot, 247
 rodzaje
 czujników sieciowych, 352
 szkodliwego oprogramowania, 245
 szyfrowania symetrycznego, 68
 zapór sieciowych, 381

- role
 - definiowane przez użytkownika, 223
 - stałe serwera, 222
 - wzajemnie się wykluczające, 175
 - z warunkiem wstępnym, 175
 - rootkit, 247, 280, 281
 - w trybie jądra, 283
 - zewnętrzny, 284
 - rotacja zapisów w dzienniku, 528
 - router odsiewający, 398
 - rozkład jednostajny, 92
 - rozmiar danych wejściowych, 468
 - rozproszona blokada usług, 298
 - rozproszone
 - ataki blokowania usług, 310
 - zapory sieciowe, 395
 - rozproszony IDS, 349
 - rozsiewanie, 245, 250, 257, 269
 - rozwalanie stosu, 422
 - rozwiązujący spory, 191
 - rywalizacja
 - o pamięć dzieloną, 488
 - o wspólne zasoby systemowe, 499
 - ryzyko, 33, 35
- ## S
- SaaS, software as a service, 549
 - sanki NOP, 438
 - schemat hasel, 113
 - SDN, software defined networks, 536
 - separacja przywilejów, 46
 - serwer, 228
 - sieci
 - komunikacyjne, 34, 39
 - nakładkowe, 537
 - rdzenia, 575
 - szkieletowe, 575
 - zdefiniowane programowo, 536
 - zdemilitaryzowane, 393
 - sieć
 - mgły, 581
 - rdzenia, 581
 - SIEM, security information and event management, 360
 - skanery oparte na hostach, 287
 - skanowanie, 259, 356
 - obwodowe, 291
 - tęczówki, 143
 - składowanie danych, 524
 - skrypty powłokowe, 491
 - slowloris, 314
 - ślabość, 33
 - Snort, 367
 - architektura, 367
 - reguły, 368
 - socjotechnika, 269
 - spam, 269
 - pocztowy, 269
 - sposoby uwierzytelniania, 105
 - sprawdzanie składni wprowadzanych danych, 477
 - sprzeniewierzenie, 38
 - sprzęt, 34, 39
 - spyware, 277
 - SQL, structured query language, 203, 209
 - DOM, 217
 - prawa dostępu, 219
 - SQLi, 210
 - stałe role serwerowe, 222
 - stan technologii robaków, 265
 - standard TIA-492, 234
 - standardy, 56
 - sterowniki interfejsów sieciowych, 232
 - stos, 422, 425
 - strategia
 - bezpieczeństwa komputerowego, 53
 - wyboru hasel, 119
 - strony
 - strażnicze, 449
 - ufające, 104, 188, 190
 - strukturalny język zapytań, 209
 - strumień kluczy, 69
 - sumy kontrolne, 345
 - superużytkownik, 167
 - sygnatura, 287
 - system
 - Snort, 367
 - Snort Inline, 402
 - wykrywania włamań, 335, 351
 - zabezpieczania wirtualizacji, 539
 - zapobiegania włamaniom, 293, 377, 398
 - zarządzania bazą danych, 203, 204
 - systemy operacyjne, 511
 - bezpieczeństwo, 514, 522
 - bezpieczeństwo aplikacji, 521
 - hartowanie, 515
 - instalowanie, 516

systemy operacyjne
konfigurowanie użytkowników, 518
kontrola zasobów, 519
planowanie bezpieczeństwa, 514
testowanie bezpieczeństwa, 521
usuwanie usług, 518
uwierzytelnianie, 518
warstwy bezpieczeństwa, 513
szablon użytkownika, 132
szkodliwa rywalizacja, 500
szkodliwe oprogramowanie, 244
szpiegostwo, 279
szyfr
blokowy, 64, 68
strumieniowy, 68
szyfrowanie, 522
asymetryczne, 83
baz danych, 226
drugoplanowe, 95
przechowywanych danych, 93
symetryczne, 62, 68
taśm oparte na bibliotece, 95
z kluczem publicznym, 79, 89

Ś

ślady wywołań systemowych, 345
świat, 157

T

tabela
danych, 206
główna, 206
tęczowa, 115
upoważnień, 160
współbieżnego dostępu, 205
tautologia, 215
technika
czujnik-aktywator, 572
informacyjna, 571
operacyjna, 571
osobista, 572
szyfrowania, 522
wstrzykiwania, 212
tekst
jawnny, 63, 79
zaszyfrowany, 63, 81

terminal użytkownika i użytkownika, 51
testowanie bezpieczeństwa, 529, 532
systemu, 521
tęczówkowy system biometryczny, 142
TIA-492, 234
TLS, Transport Layer Security, 503
topologia zapór sieciowych, 397
tożsamość, 184, 569
cyfrowa, 185
inicjatywna, 184
użytkownika, 380
triada CIA, 27
trojany w telefonach przenośnych, 271
trwale zagrożenie, 249
tryby działania, 67
tylne drzwi, 246, 280

U

uczenie maszynowe, 343
uczniowie, 334
UDP, 308
ujawnienie treści komunikatów, 41
ujednolicone opanowywanie zagrożeń, 404
unieruchomienie, 37
uniksowy system plików, 165
upoważnianie, 154, 180, 582
kaskadowe, 219
uprowadzenie stacji roboczej, 110
uruchamianie maszyny wirtualnej, 571
urządzenia, 34
nadawczo-odbiorcze, 572
ustanowienie połączenia uwierzytelnianego
hasłem, 129
usuwanie
oprogramowania szpiegującego, 290
zbędnych usług, 518
uszkodzanie danych, 273
uszkodzenie, 37
uwarstwienie, 48
uwierzytelnianie, 154, 518, 582
komunikatów, 69
bez ich szyfrowania, 70
za pomocą jednokierunkowej funkcji
haszowania, 74
za pomocą kodu uwierzytelniającego, 72
z użyciem szyfrowania symetrycznego, 70

- użytkownika, 101
 - biometryczne, 129, 132
 - model architektoniczny, 104
 - obszary ryzyka, 108
 - ocena ryzyka, 105
 - oparte na hasłach, 109
 - oparte na żetonach, 124
 - potencjalny wpływ, 107
 - sposoby, 105
 - wieloczynnikowe, 105
 - zagadnienia bezpieczeństwa, 140
 - zasady, 103
 - zdalne, 135
- uzurpacja, 36, 37
- użytkowanie pamięci, 487
- użytkownik, 228, 526

V

- VM hosta-bastionu, 540
- VoIP, Voice over IP, 312
- VPN, virtual private network, 393
- VXLAN, 537

W

- wejście, 468
- weryfikacja, 132
- weryfikator, 104
- widok, 208
- więzienie chrootowe, 528
- wirtualizacja, 532
 - aplikacji, 537
 - goszczona, 536
 - kontenerowa, 537
 - rodzima, 536
- wirtualna zaporą sieciową, 391
- wirtualne
 - rozszerzone sieci lokalne, 537
 - sieci nakładkowe, 537
 - sieci prywatne, 393
 - zapory sieciowe, 540
- wirus, 247
 - komputerowy, 250
 - metamorficzny, 257
 - niewidzialny, 256
 - skryptowy, 250, 253
 - szyfrowany, 256

- wieloczęściowy, 256
- wielopostaciowy, 256
- włamanie, 37, 331
- własność jednokierunkowości, 76
- właściciel, 157
 - danych, 227
- włączka, 267
- wnioskowanie, 36, 223, 225
- wstrzykiwanie
 - do CGI, 472
 - drugiego rzędu, 214
 - kodu, 473
 - poleceń, 52, 471
 - SQL, 474
 - SQL na ślepo, 216
- wybór języka programowania, 443
- wyciek pamięci, 487
- wyjscie programu, 504
- wykorzystanie
 - omylek użytkownika, 111
 - wielokrotnego użycia hasła, 111
 - wrażliwych punktów, 257
- wykrywanie, 55, 321
 - anomalii, 341, 346, 356
 - ataku, 217
 - heurystyczne, 343
 - nadużyć, 341
 - naruszeń, 582
 - sygnatur, 341, 343, 355
 - włamań, 78, 331, 336, 355
 - hybrydowe, 358
 - oparte na gościu, 344
 - oparte na sieci, 351
 - rozproszone, 358
 - wnioskowania, 225
- wyłudzanie, 278
- wymagania
 - bezpieczeństwa, 43, 44
 - funkcjonalne, 42
 - prywatności, 578
- wymiana
 - informacji, 188
 - potwierdzeń, 305
 - tożsamości, 187
- wyrażenia regularne, 478
- wystawienie na widok, 35
- wywołania systemowe, 451, 496

wyzwalacz, 252
 wyzyskiwacze, 246
 wzajemne uwierzytelnianie, 578

Z

zaawansowane trwałe zagrożenie, 246
 zabezpieczanie
 aplikacji, 528
 poczty elektronicznej, 567
 sieci, 567
 zachowanie intruzów, 335
 zagrożenia, 33–35, 38
 zainfekowana treść, 250
 zakażacz
 plików, 256
 sektora rozruchowego, 256
 zakłócenie, 36, 37
 zalecenia NIST, 558
 załadowanie programu, 426
 zaniebdywanie miarodajności, 340
 zapobieganie, 54
 atak, 321
 utracie danych, 566
 w fazie wykonania, 217
 zaporą
 filtrująca pakiety, 382
 kontrolująca stan, 386
 sieciowa, 377, 379
 hiperwizora, 541
 konfiguracja, 394
 lokalizacja, 397
 oparta na goście, 390
 osobista, 391
 rezydująca w goście, 398
 rozproszona, 395
 topologia, 397
 uwzględnienie stanu, 386
 wirtualna, 391, 540
 w urządzeniach sieciowych, 391
 zaprzeczenie, 37
 zapytania
 błędne, 215
 parametryzowane, 216
 wleczone, 215
 zarządzanie
 dostępem, 186, 566
 kluczami, 227

polityką, 186
 poświadczeniami, 185
 przywilejami, 186
 tożsamością, 183, 566
 zasobami, 186
 zdarzeniami, 360
 zasada najmniejszych przywilejów, 47
 zasady
 cyfrowego uwierzytelniania użytkownika, 103
 egzekwowane przez sieć, 582
 kontrolowania dostępu, 154
 projektowania bezpieczeństwa, 44
 zasób systemu, 32, 33
 zastąpienie ramki stosu, 450
 zastosowania
 botów, 275
 IDS, 340
 zatapiacze, 246
 zatapianie
 HTTP, 314
 ICMP, 308
 poleceniem ping, 302
 SIP, 312
 TCP SYN, 309
 UDP, 308
 zatruty pakiet, 300
 zdalne
 przejmowanie kontroli, 276
 uwierzytelnianie, 135
 wstrzykiwanie kodu PHP, 474
 zdarzenia
 DDI, 361
 PEP, 361
 zmienne
 serwera, 214
 środowiskowe, 489
 zombie, 247, 275, 310
 zwiad, 356
 związek zaufania, 583

Ż

źródła ataków, 249

Ż

żeton, 124, 569

PROGRAM PARTNERSKI

— GRUPY 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>

GRUPA
Helion 

CYBEROBRONA: BĄDŹ CZUJNY I PRZYGOTUJ SIĘ!

Zapewnienie bezpieczeństwa systemu informatycznego jest dziś nie lada wyzwaniem. Między administratorami a napastnikami trwa ciągły wyścig zbrojeń. Agresorzy dysponują bardzo różnymi narzędziami i często postępują w sposób nieprzewidywalny. W efekcie każde zabezpieczenie usługi czy zasobu, mimo że początkowo wydaje się doskonałe, prędzej czy później okazuje się podatne na ataki. Jedyną rzeczą, jaką może zrobić administrator bezpieczeństwa systemu, jest ciągłe utrzymywanie stanu gotowości, a także odpowiednio wczesne wykrywanie prób ataku i sukcesywne ich neutralizowanie. Poza tym powinien cały czas się uczyć i aktualizować swoją wiedzę.

Ta książka to kolejne, zaktualizowane i uzupełnione wydanie znakomitego podręcznika przeznaczonego dla projektantów systemów i administratorów bezpieczeństwa. Poruszono w niej zagadnienia określania zagrożeń systemów komputerowych i sieci, oceny względnego ryzyka tych zagrożeń i opracowywania efektywnych kosztowo i przyjaznych dla użytkownika środków zaradczych. Wyjaśniono także najważniejsze zasady utrzymywania bezpieczeństwa systemu i wskazano, dlaczego ich przestrzeganie ma kluczowe znaczenie. Zaprezentowano również metody projektowe pozwalające na zaspokojenie wymagań bezpieczeństwa komputerowego, szeroko omówiono ważniejsze standardy w tej dziedzinie, a poszczególne kwestie zilustrowano za pomocą praktycznych przykładów.

Najciekawsze zagadnienia:

- Zasady bezpieczeństwa i ich wdrożenie
- Bezpieczeństwo oprogramowania i infrastruktury
- Elementy kryptografii
- Praca administratora bezpieczeństwa
- Zapewnienie bezpiecznej pracy sieci

Dr William Stallings jest autorem kilkudziesięciu książek i artykułów wielokrotnie publikowanych przez ACM i IEEE. Trzynastokrotnie zdobył nagrodę za najlepszy podręcznik informatyczny roku. Działa w branży od ponad 30 lat, projektował i implementował zestawy protokołów sieciowych dla różnych systemów.

Dr Lawrie Brown wykłada w School of Engineering and Information Technology w Australii. Specjalizuje się w zagadnieniach komunikacji oraz bezpieczeństwa systemów, a także kryptografii i projektowania bezpiecznych środowisk zdalnego wykonywania kodu.

Helion
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

KOD KORZYŚCI
Sięgnij po więcej! ▶



ISBN 978-83-8322-550-0



Cena: 119,00 zł

