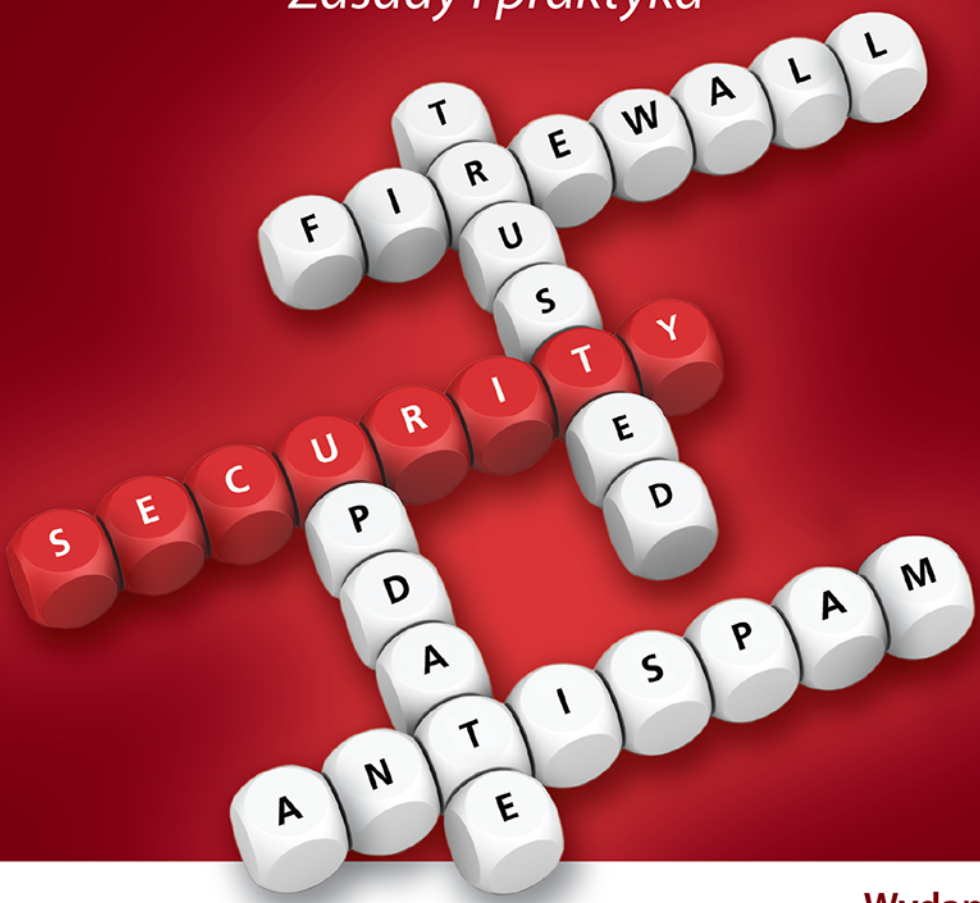
William Stallings • Lawrie Brown

BEZPIECZEŃSTWO SYSTEMÓW INFORMATYCZNYCH

Zasady i praktyka



TOM 1

Wydanie IV

Pearson

Helion

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.

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.

SPIS TREŚCI

ROZDZIAŁ 4

KONTROLOWANIE DOSTĘPU

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.

Rozpoczniemy ten rozdział od przeglądu kilku ważnych pojęć. Potem przyjrzymy się trzem powszechnie stosowanym technikom implementowania zasad kontrolowania dostępu. Następnie poszerzymy nasz ogląd 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 urządzeń (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 stosowalnymi 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. Chroń dostęp bezprzewodowy za pomocą uwierzytelniania i szyfrowania
18. Kontroluj połączenia urządzeń mobilnych
19. Szyfruj CUI w urządzeniach mobilnych
20. Weryfikuj i kontroluj lub ograniczaj połączenia i korzystanie z zewnętrznych systemów informacyjnych
21. Ograniczaj użycie w zewnętrznych systemach informacyjnych przenośnych urządzeń 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.

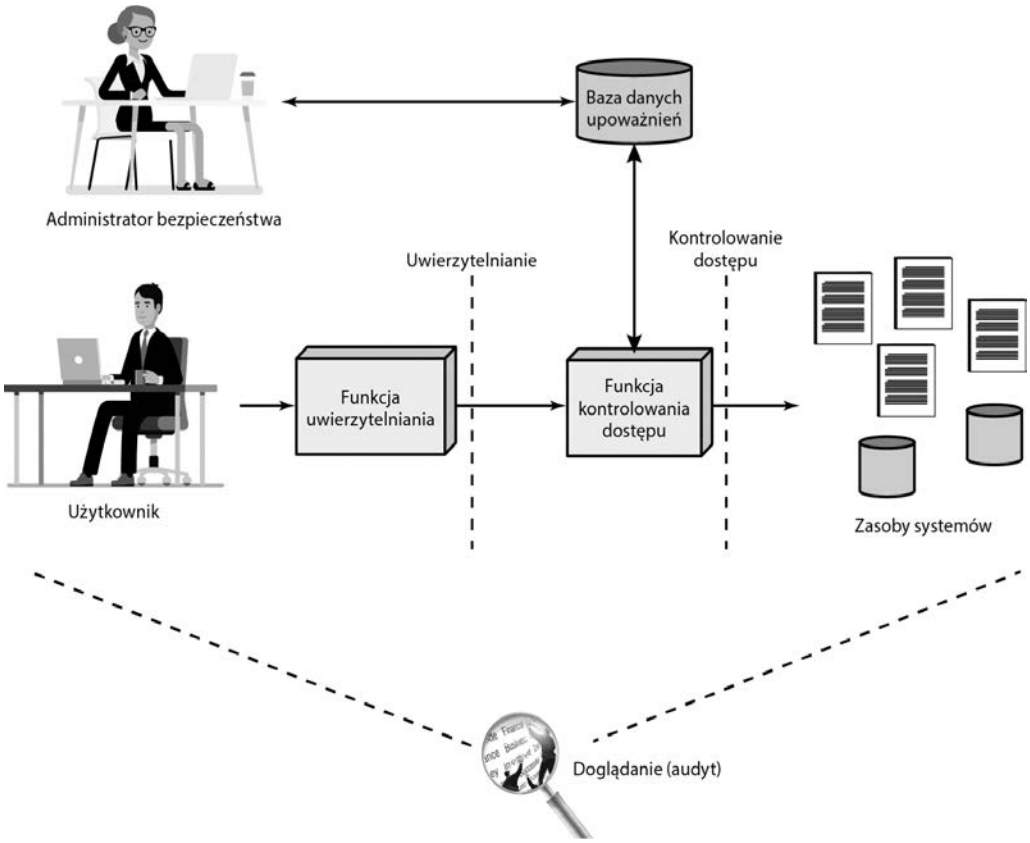
Kontekst kontrolowania dostępu

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 zajmiemy 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 procesem[1]. 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 pożądanych 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 DAC[2] stosowane przez system operacyjny lub system zarządzania bazą danych zasadza się na **macierzy dostępów** (ang. *access matrix*). Pomysł macierzy dostępów 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.

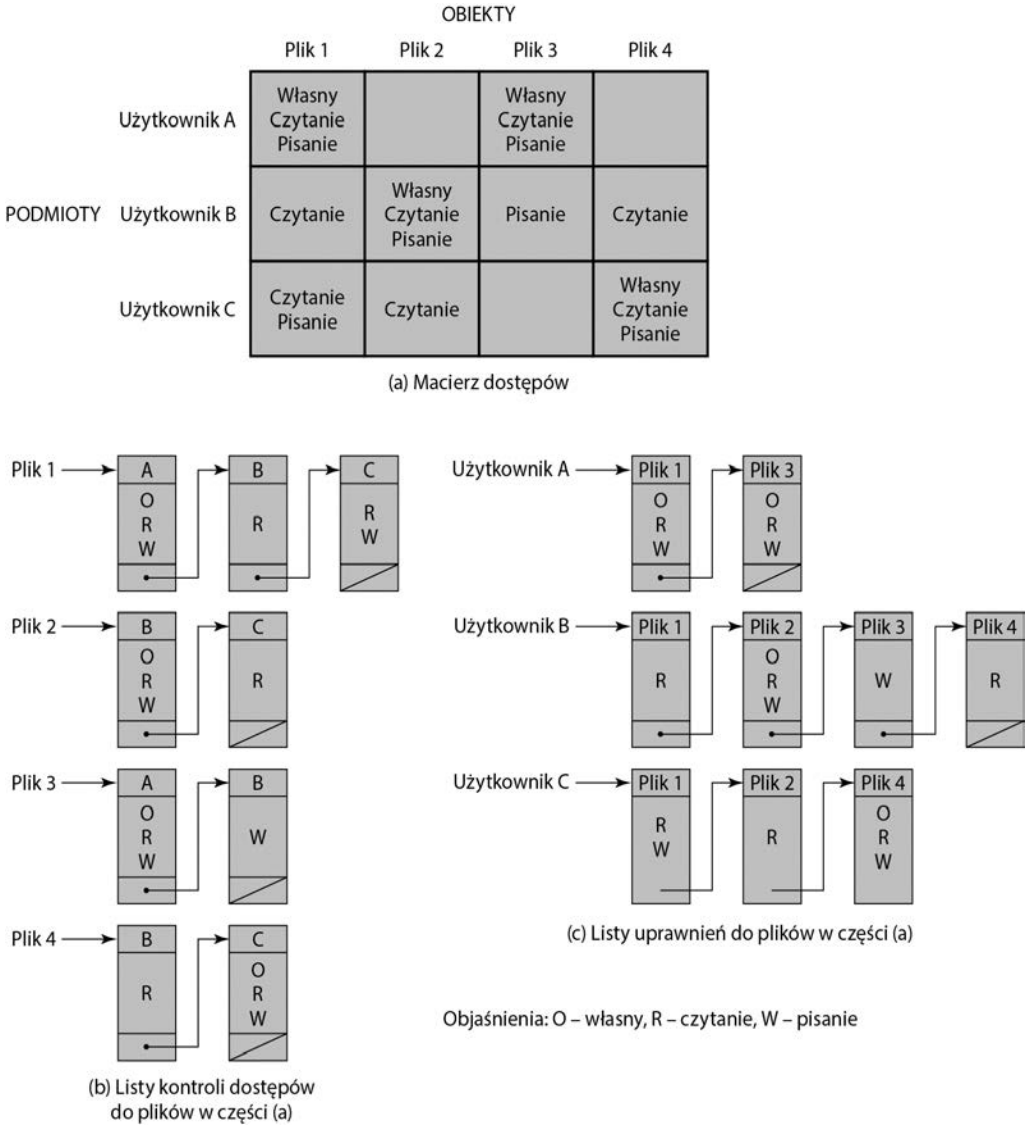
Rysunek 4.2a, oparty na rysunku z [SAND94], ukazuje prosty przykład macierzy dostępów. Mamy tu użytkownika A posiadającego pliki 1 i 3 oraz prawa ich czytania i zapisywania. Użytkownik B ma prawo czytania pliku 1 itd.

W praktyce macierz dostępów 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.*



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 dostępów, 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. Zdefiniujmy 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 dostępów 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 dostępów 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:

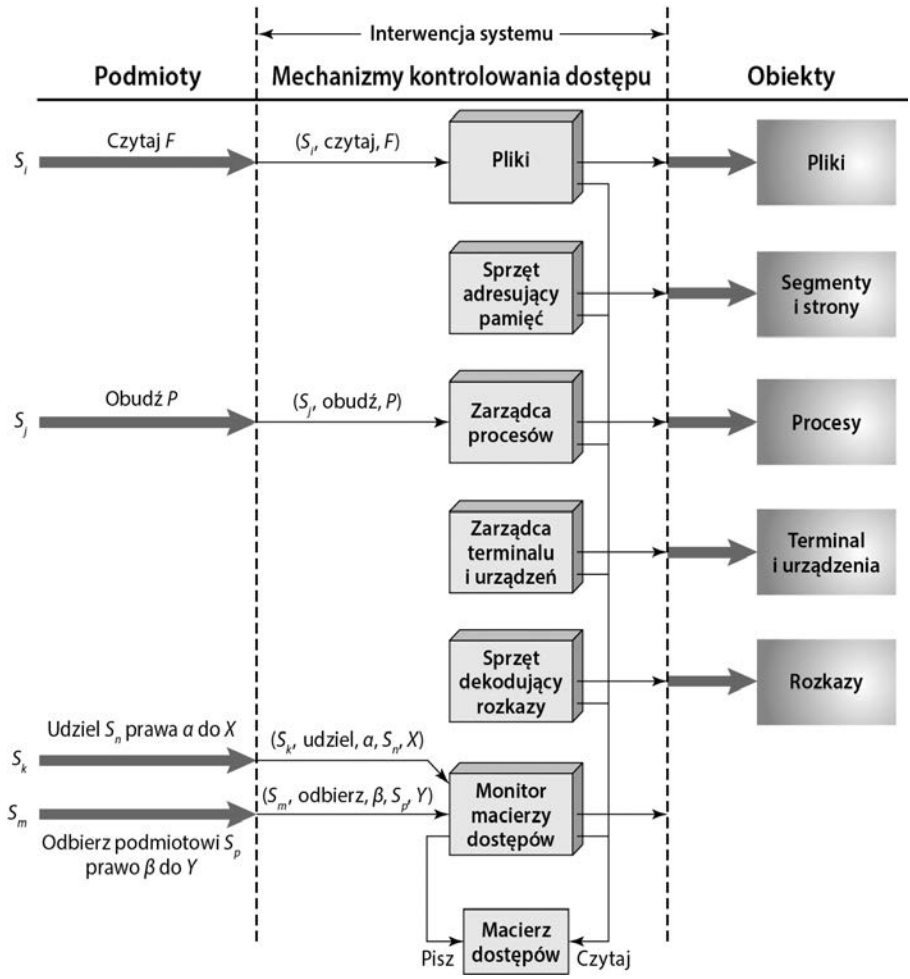
1. Podmiot S_0 wydaje żądanie typu α pod adresem obiektu X.
2. Żądanie to powoduje wygenerowanie przez system (system operacyjny lub jakiś moduł interfejsu kontroli dostępu) komunikatu postaci (S_0, α, X) dla kontrolera X.
3. Kontroler zagląda do macierzy A, żeby sprawdzić, czy α występuje w $A\ [S_0,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

OBIEKTY

PODMIOTY	Podmioty			Pliki		Procesy		Napędy dysków	
	S_1	S_2	S_3	F_1	F_2	P_1	P_2	D_1	D_2
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 S0)	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 $[S_0, X]$	kopiuj 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 dostępów. 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 dostępów 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 *właściciel* 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 dostępów 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 dostępów 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 dostępy 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 zwykło 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 drzewa[4]. 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.*

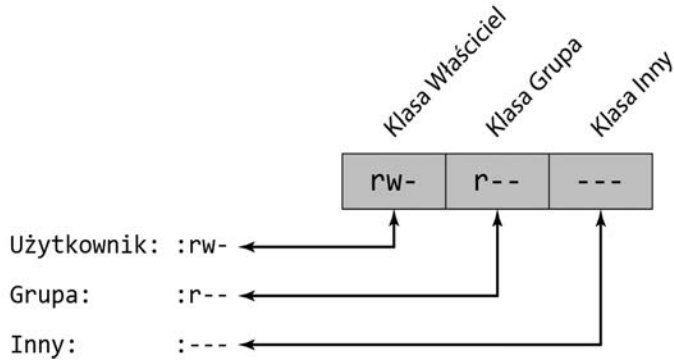
Tradycyjne kontrolowanie dostępu do plików w UNIX-ie

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.

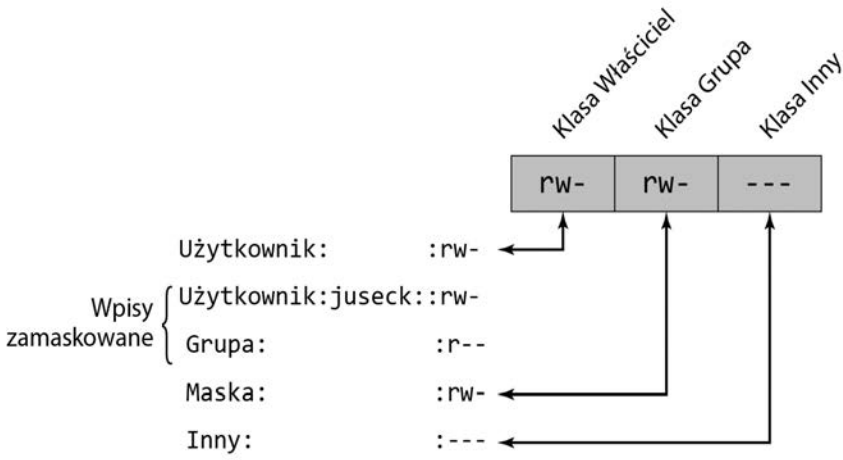
Dziewięć bitów ochrony określa prawa czytania, pisania i wykonywania przysługujące właścicielowi pliku, innym członkom grupy, do której ten plik należy, i wszystkim pozostałym użytkownikom. Tworzy to hierarchię złożoną z właściciela, grupy i wszystkich innych, w ramach której są używane najbardziej niezbędne 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 katalogu[5]. 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.



(a) Konwencjonalna metoda uniksowa (minimalna lista kontroli dostępu)



(b) Rozszerzona lista kontroli dostępu

Rysunek 4.5. **Kontrolowanie dostępu do pliku w UNIX-ie**

Ostatnim bitem pozwoleń jest „bit przylepności" (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, przemieścić lub usunąć. Jest to przydatne do administrowania plikami we wspólnych katalogach tymczasowych.

Jeden szczególny identyfikator użytkownika oznacza **superużytkownika** (ang. *superuser*). 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 możliwe[6]. 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.

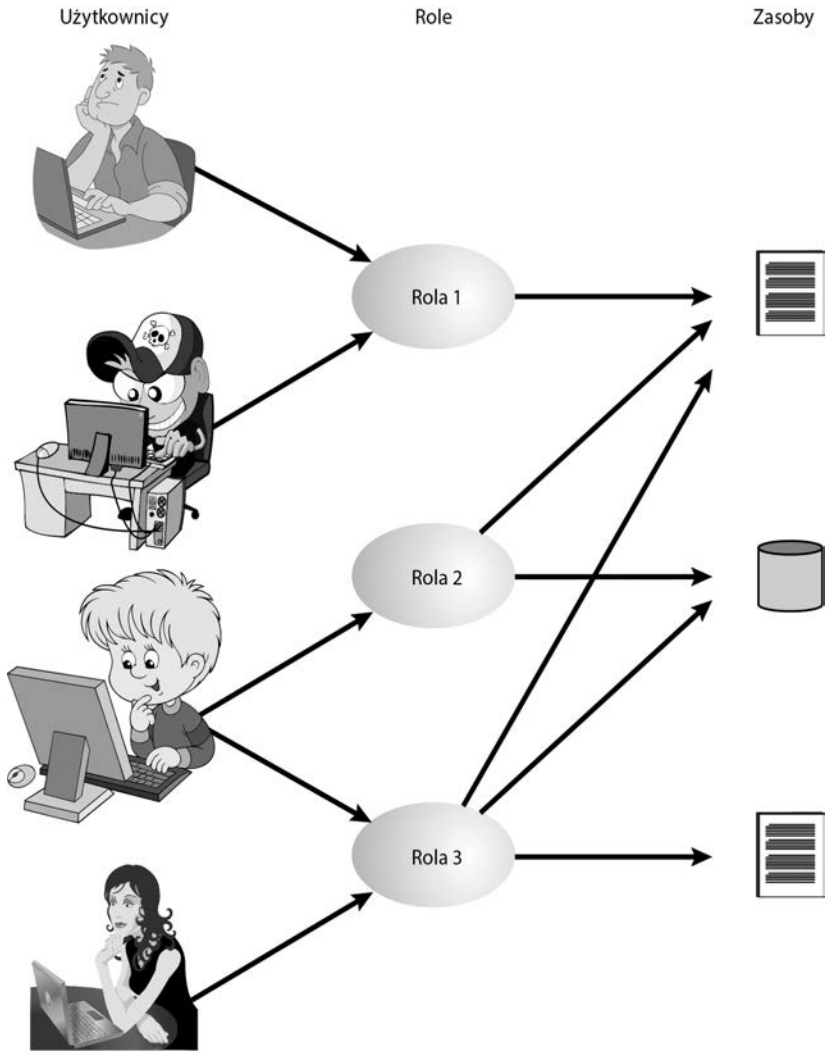
4.5. KONTROLOWANIE DOSTĘPU WEDŁUG RÓL

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.

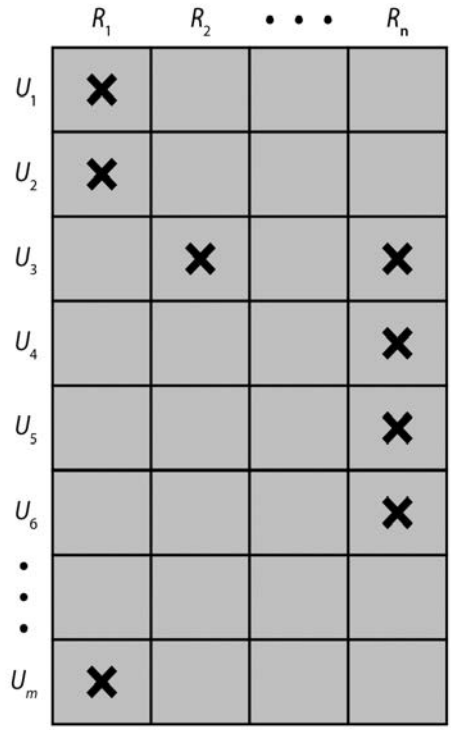
Zależność między użytkownikami i rolami jest typu wielu na wiele, podobnie jak zależność między rolami i zasobami, czyli obiektami systemu (zob. rysunek 4.6). Zbiór zmian użytkowników — w niektórych środowiskach częstych — i przypisanie użytkownika do jednej lub większej liczby ról mogą również być dynamiczne. 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 dostępów, 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 rolom. Zauważmy, że rolę można traktować jak obiekt, co umożliwia definiowanie hierarchii ról.



OBIEKTY

ROLE	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			

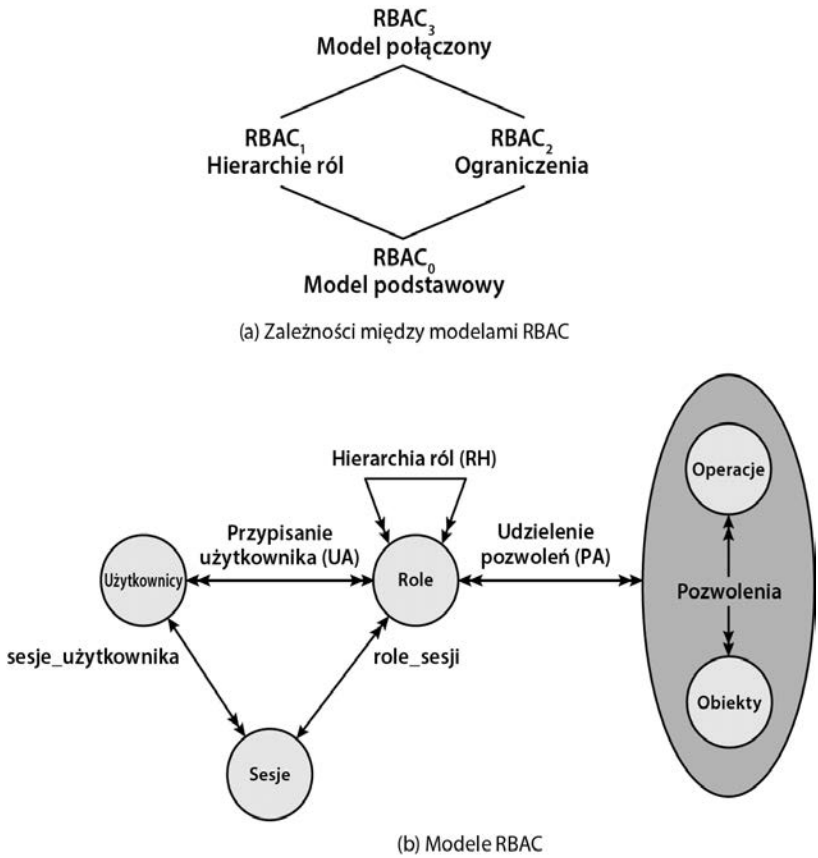
Rysunek 4.7. **Reprezentacja RBAC (kontroli dostępu według ról) za pomocą macierzy kontroli dostępów**

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 $RBAC_2$.



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 — $RBAC_0$

Rysunek 4.8b, bez hierarchii ról i ograniczeń, zawiera cztery typy jednostek w systemie $RBAC_0$:

- **Użytkownik.** Osoba mająca dostęp do danego systemu komputerowego. Każda osoba ma przydzielony ID użytkownika.
- **Rola.** Nazwane stanowisko pracy w ramach organizacji (instytucji lub przedsiębiorstwa[8]) 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.** Odwzorowanie 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 „wielu 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

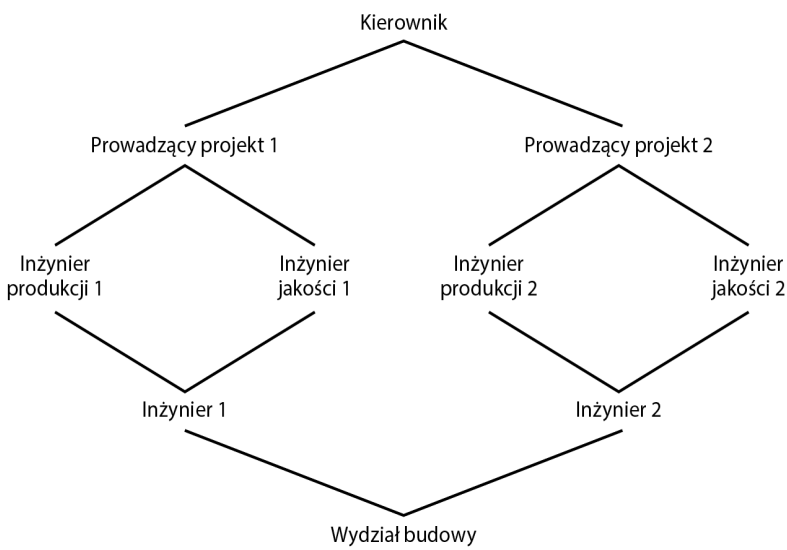
[8] Przypominamy, że wielofunkcyjny w języku angielskim termin *organization* zastępuje w tym przekładzie, gwoli oszczędności, każdy z tu wymienionych — *przyp. tłum.*

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 — $RBAC_1$

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 — $RBAC_2$

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, liczność 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 komitywy między osobnikami o różnych specjalnościach lub odmiennych stanowiskach, mającej przeszkodzić w polityce bezpieczeństwa.

Liczność (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 $RBAC_3$.

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 = PatientInforma`
`↪tionAccess`, `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.*

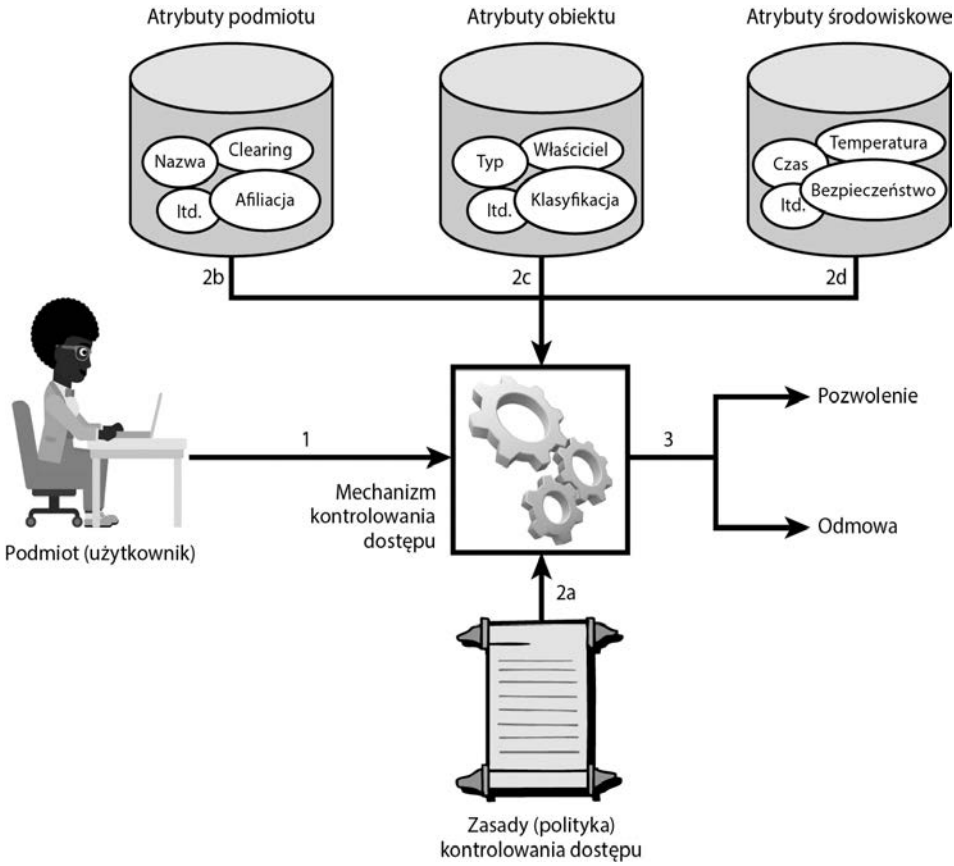
- **Atrybuty obiektu.** Obiekt, nazywany również **zasobem**, jest pasywną (w kontekście danego żądania) jednostką związaną z systemem informacyjnym (obiektami mogą być na przykład urządzenia, pliki, rekordy, tabele, procesy[10], programy, sieci, domeny), zawierającą lub odbierającą informacje. Obiekty, 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.*

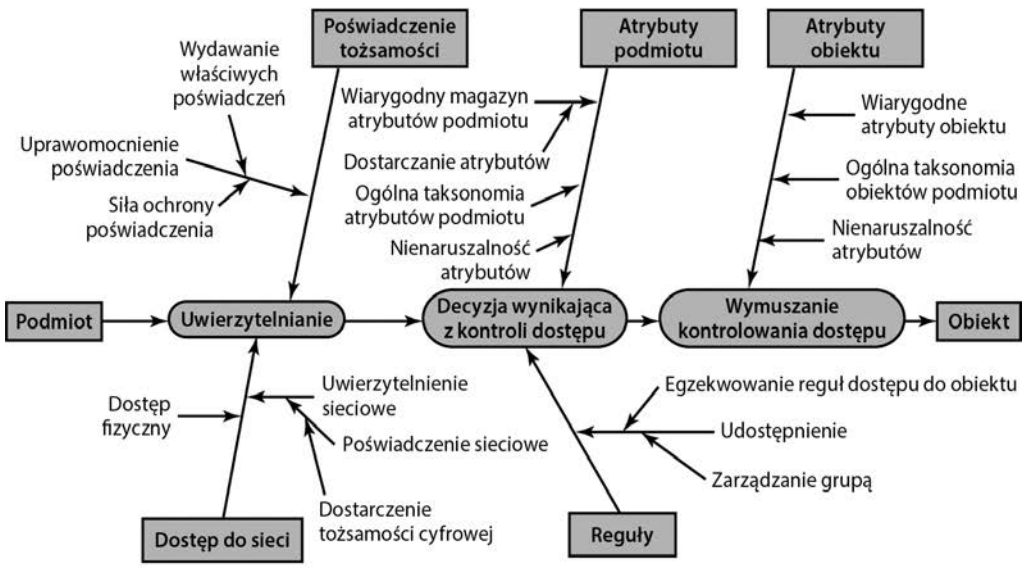


Rysunek 4.10. **Scenariusz ABAC**

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, umożliwiający 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 Issuers [11]. 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 ($1 \le k \le K$), OA_m ($1 \le m \le M$) i EA_n ($1 \le n \le N$) 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):

   $ATTR(s) \subseteq SA_1 \times SA_2 \times \ldots \times SA_k$
   $ATTR(r) \subseteq SA_1 \times SA_2 \times \ldots \times SA_m$
   $ATTR(c) \subseteq SA_1 \times SA_2 \times \ldots \times SA_n$

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

   ```
   Role(s) = "Konsument usługi"
   WłaścicielUsługi(o) = "XYZ, Inc."
   BieżącaData(e) = "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*:

   ```
   Rule: can_access (s, o, c) ← 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 skontrastowania 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:

```
R1:can_access(u, m, e) ←
   (Wiek(u)   17 ∧ Kalifikacja(m) ∈ {R, PG-13, G}) ∨
   (Wiek(u)   13 ∧ Wiek (u) < 17 ∧ Kalifikacja(m) ∈ {PG-13, G}) ∨
   (Wiek(u) < 13 ∧ Kalifikacja(m) ∈{G})
```

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 przypisywaniami 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 „staroć", opartych na dacie ukazania się filmu porównywanej z datą bieżącą, a użytkownicy są zaliczeni do klas „premium" i „zwykła" 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 `Zakres()` oznacza przedział wartości możliwych do przyjęcia, to odpowiednia liczba ról i pozwoleń wymaganych w modelu RBAC wyniesie:

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

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:

```
R2:can_access(u, m, e) ←
   (TypCzłonkowstwa(u) = Premium) ∨
   (TypCzłonkowstwa(u) = Zwykły ∧ TypFilmu(m) = Staroć)
R3:can_access(u, m, e) ← R1 ∧ R2
```

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 (ICAM[12]), 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 przedsięwzięciach wymagających ujednoliconego podejścia do kontrolowania dostępu. Projektowaniu ICAM-u przyświecały następujące cele:

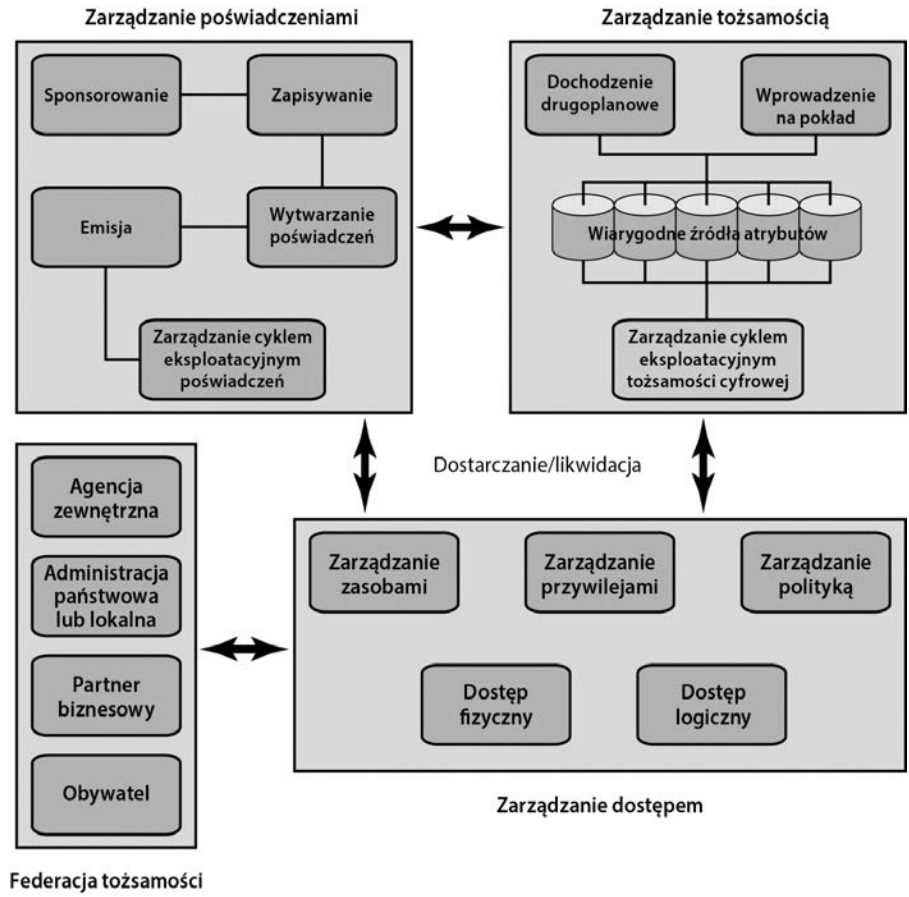
- Tworzenie godnych zaufania reprezentacji tożsamości cyfrowej osób i tego, co jest określane w dokumentacji ICAM-u jako **jednostki nieosobowe** (ang. *nonperson entities* — NPEs). Ostatnie określenie dotyczy procesów, aplikacji i zautomatyzowanych urządzeń potrzebujących dostępu do jakiegoś zasobu.
- 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] Skrót 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 opisania 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 uzyskiwać 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 interesy[13]? 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.

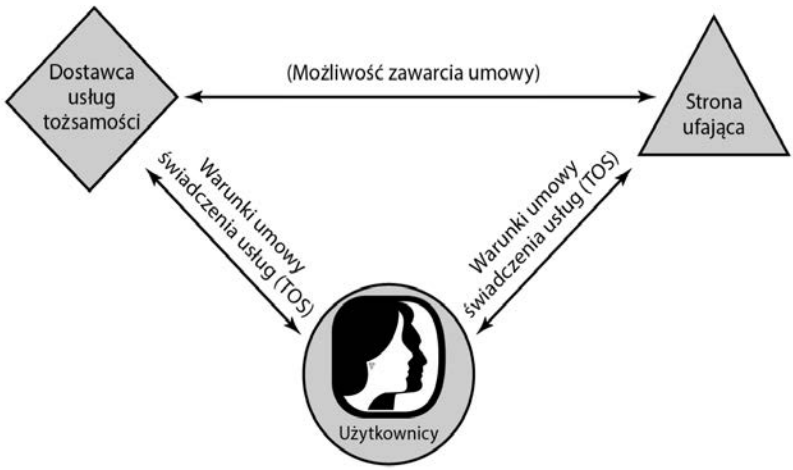
Troska o atrybuty nabiera coraz większego znaczenia we wszystkich rodzajach sytuacji kontrolowania dostępu, nie tylko w e-handlu. Na przykład dane przedsiębiorstwo może potrzebować dostępu do zasobów klientów, użytkowników, dostawców i partnerów. Zależnie od kontekstu dostęp będzie zdeterminowany nie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

Tradycyjne podejście do wymiany tożsamości

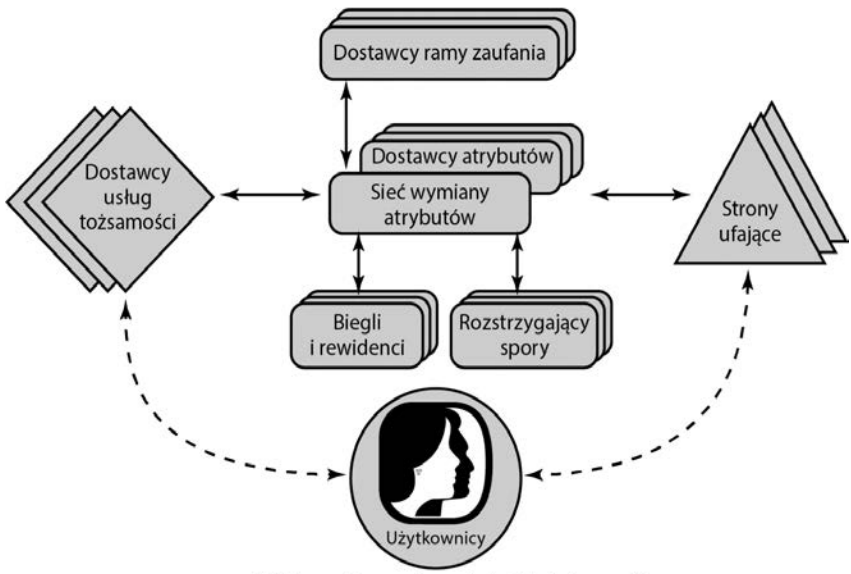
Transakcje bezpośrednie lub sieciowe dotyczące przedstawicieli różnych firm lub dokonywane między firmą a indywidualnym użytkownikiem, na przykład kupującym online, z reguły wymagają dzielenia informacji dotyczących tożsamości. Prócz prostej nazwy lub numeru identyfikacyjnego te informacje mogą zawierać 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 — *przyp. tłum.*



(a) tradycyjny trójkąt stron zaangażowanych w wymianę informacji o tożsamości



(b) Elementy wymiany atrybutów tożsamości

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 (wiarygodny) 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 metasystemów 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 ramy 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 prze 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 uprawomocnienia 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 wyposażyła 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 Drezdeński[14] 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 hoście[15] z osobna 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 ścisłe 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 uwidoczniono 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 bankowość biurowa i kierownik grupy a rolą analityk finansowy i urzędnik, 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 przekładzie; 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	Analityk finansowy	Urzędnik
B	Analityk finansowy	Kierownik grupy
C	Analityk finansowy	Naczelnik oddziału
D	Analityk finansowy	Junior
E	Analityk finansowy	Senior
F	Analityk finansowy	Specjalista
G	Analityk finansowy	Asystent
…	…	…
X	Technik udziałów	Urzędnik
Y	Pomoc e-handlu	Junior
Z	Bankowość biurowa	Naczelnik oddziału

(b) Przydział pozwoleń

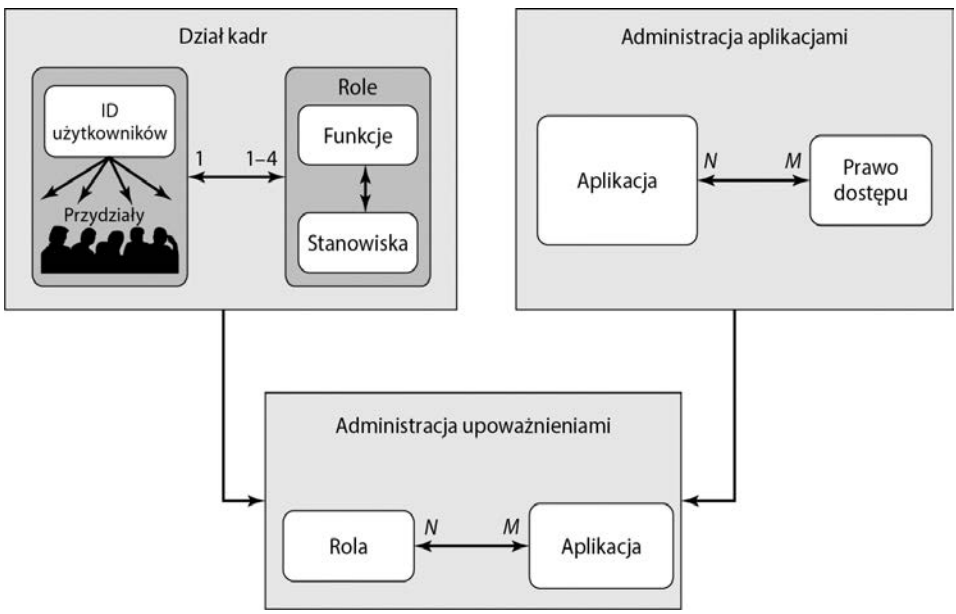
Rola	Aplikacja	Prawa dostępu
A	Instrumenty rynku pieniężnego	1, 2, 3, 4
	Handel instrumentami pochodnymi	1, 2, 3, 7, 10, 12
	Instrumenty odsetkowe	1, 4, 8, 12, 14, 16
B	Instrumenty rynku pieniężnego	1, 2, 3, 4,7
	Handel instrumentami pochodnymi	1, 2, 3, 7, 10, 12, 14
	Instrumenty odsetkowe	1, 4, 8, 12, 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 dystrybuuje między aplikacje 42 000 profili bezpieczeństwa.

4.10. PODSTAWOWE POJĘCIA, PYTANIA SPRAWDZAJĄCE I ZADANIA

Podstawowe pojęcia

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

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

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 dostępów z rysunku 4.2a.

b) Naszkicuj graf skierowany odpowiadający macierzy dostępów z rysunku 4.3.

c) Czy między reprezentacją w postaci grafu skierowanego a reprezentacją w postaci macierzy dostępów 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 przerwań i operacje wejścia-wyjścia.
- **Egzekutor.** Wykonywanie wielu usług systemu operacyjnego, w tym procedur zarządzania plikami i rekordami (dyskowymi i taśmowymi).

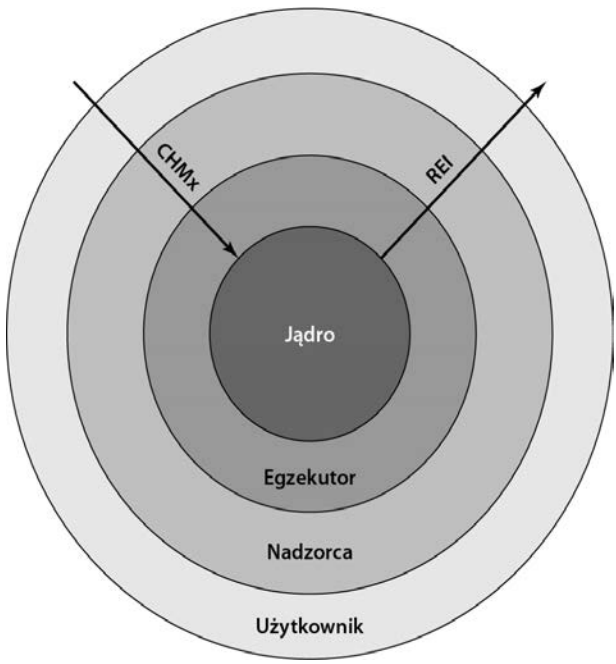
- **Nadzorca.** Wykonywanie innych usług systemu operacyjnego, na przykład odpowiadanie na polecenia użytkownika.
- **Użytkownik.** Wykonywanie programów użytkownika oraz programów narzędziowych, takich jak kompilatory, edytory, konsolidatory i debugery.

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).

a) Niektóre systemy operacyjne mają dwa tryby działania: jądra i użytkownika. Jakie są zalety i wady występowania czterech trybów zamiast dwóch?

b) Czy znajdujesz uzasadnienie użycia więcej iż 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 dostępnie 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 jedne, 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 tworzonym 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 Sieci[16] 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 możność 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 $\subseteq$ ROLE $\times$ 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.

a) Korzystając odpowiednio z powyższych definicji, podaj formalną definicję ogólnej hierarchii ról.

b) 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*.

a) 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. Wyraź ten związek formalnie.

b) Alternatywną hierarchią ról będzie taka, w której rola jest nadrzędna wobec innej, jeżeli jej funkcja ma wyższą rangę niezależnie od stanowiska. Wyraź ten związek formalnie.

4.12. Nawiązujemy do przykładu z magazynem rozrywek online z podrozdziału 4.6 z drobnoziarnistą polityką uwzględniającą użytkowników premiowanych i zwykłych. Wymień wszystkie role i wszystkie przywileje w tym przykładzie potrzebne do zdefiniowania modelu RBAC.

SKOROWIDZ

A

B

C

D

E

F

G

H

I

J

K

L

Ł

M

N

O

P

R

S

Ś

T

U

V

W

Z

Ź

Ż

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
helion.pl
HELION SA
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
9 788383 225500
Cena: 119,00 zł
Pearson