STUDIA INFORMATICA 2013
Volume 34 Number 2B (112)

Krzysztof CZAJKOWSKI, Radosław KORKOSZ
Instytut Teleinformatyki, Politechnika Krakowska

ZARZĄDZANIE UPRAWNIENIAMI Z WYKORZYSTANIEM ORACLE ENTITLEMENTS SERVER

Streszczenie. W artykule przedstawiono tematykę zintegrowanego zarządzania uprawnieniami na przykładzie Oracle Entitlements Server. Zaprezentowano również porównanie OES z systemami IBM Tivoli Security Policy Manager oraz Novell Access Manager.

Słowa kluczowe: zarządzanie uprawnieniami, uwierzytelnianie, autoryzacja

ENTITLEMENTS MANAGEMENT USING ORACLE ENTITLEMENTS SERVER

Summary. The article describes an issue of integrated entitlements management on the base of Oracle Entitlements Server. The paper introduces IBM Tivoli Security Policy Manager and Novell Access Manager systems as well.

Keywords: entitlements management, authentication, authorization

1. Wstęp

W przeszłości nazwy oraz hasła użytkowników przechowywane były w prostych plikach tekstowych. Definiowanie zabezpieczeń nieznacznie różniło się tylko od modyfikacji tych plików w edytorze tekstowym, uwierzytelnianie zaś było po prostu porównaniem ciągu znaków z hasłem użytkownika. Obecnie znaczna część organizacji przechowuje dane o użytkownikach korzystających z ich systemów przy użyciu narzędzi umożliwiających realizację protokołu LDAP, stosując przy tym wyspecjalizowane rozwiązania do zabezpieczeń, które są zintegrowane z mechanizmami SSO (ang. *Single Sign-On*). Nadal jednak można napotkać obszary związane z zarządzaniem tożsamością, które są obsługiwane bezpośrednio przez aplikacje – przykładem jest szczegółowa autoryzacja (ang. *fine-grained authorization*). Na-

wet jeśli autoryzacja oparta na ujednoliconym formacie adresowania zasobów jest wyeksportowana do zewnętrznego narzędzia, główna część autoryzacji w aplikacji realizowana jest poprzez wewnętrzną implementację w kodzie. Realizacja wymogów bezpieczeństwa bezpośrednio w kodzie programu oznacza:

- kruchość zasad bezpieczeństwa oraz konieczność przejścia każdej zmiany reguł przez długie cykle testów,
- niezdolność do szybkiego reagowania na zagrożenia i naruszenia bezpieczeństwa,
- trudność w analizowaniu oraz kontrolowaniu zarówno polityk bezpieczeństwa, jak i decyzji o autoryzacji,
- wymuszenie na twórcach aplikacji wykonywania działań, które już zostały wcześniej zrealizowane, co prowadzi do spowolnienia rozwoju aplikacji, wyższych kosztów oraz często stosowania sztywnych implementacji zabezpieczeń.

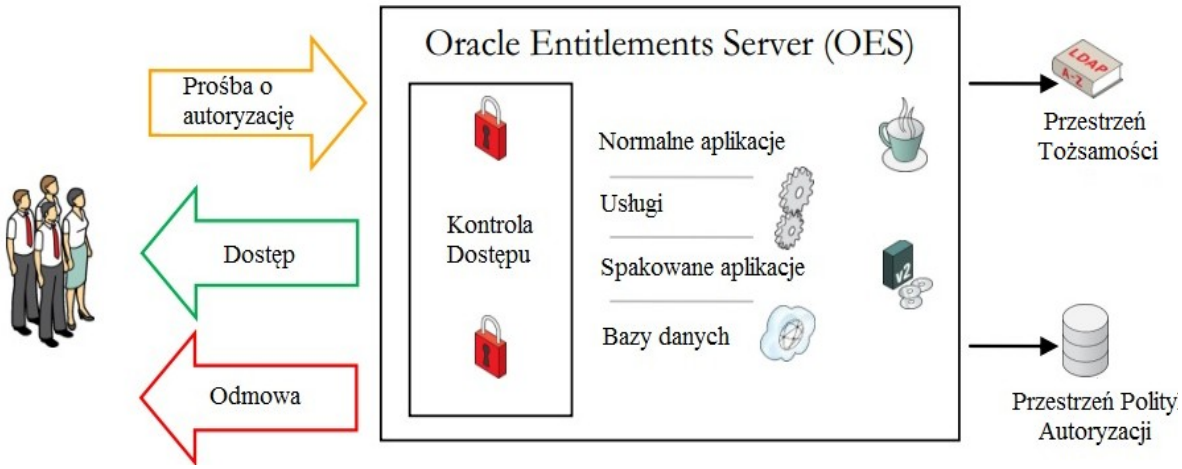
Powyższe problemy ukazują, że bezpieczeństwo systemów informatycznych jest wyspecjalizowanym obszarem, który powinien być realizowany na zewnątrz kodu aplikacji. Dzięki wykorzystaniu zewnętrznych mechanizmów autoryzacji, w skomplikowanych systemach aplikacje nie muszą uwzględniać zawiłości zarządzania bezpieczeństwem. Mogą skupiać się na dostarczaniu najlepszych rozwiązań dla problemów biznesowych, bez obaw o aspekty związane z bezpieczeństwem oraz zgodnością.

W artykule zaprezentowano narzędzie firmy Oracle, które ma za zadanie umożliwić zarządzanie tożsamością w złożonych systemach informatycznych. Przedstawiono konstrukcję, zasady wykorzystania oraz możliwości Oracle Entitlements Server. Środowisko OES zostało również porównane z narzędziami IBM Tivoli Security Policy Manager oraz Novell Access Manager.

1.1. Wprowadzenie

Oracle Entitlements Server (OES) jest narzędziem umożliwiającym ochronę wewnętrznych zasobów danej organizacji poprzez definiowanie i zarządzanie tzw. politykami, które kontrolują dostęp oraz dają możliwość wykorzystania tych zasobów [1]. Sposoby dostępu zdefiniowane są w politykach poprzez określenie, kto jest upoważniony do korzystania z odpowiednich zasobów oraz kiedy i jak dana operacja może zostać wykonana. Konkretna polityka może egzekwować dostęp do wszystkich rodzajów zasobów zawierających zarówno elementy oprogramowania (tj. dresy URL, komponenty JSP oraz EJB, metody oraz serwlety użyte do budowy aplikacji), jak i obiekty biznesowe (np. reprezentacje kont, profili osobistych i umów, takich jak rachunki bankowe w aplikacjach bankowych). Oracle Entitlements Server obsługuje tworzenie polityk ról oraz polityk kontroli dostępu. Polityki ról stosowane

są do definiowania więzów użytkownicy-role. Polityki kontroli dostępu definiują natomiast dostęp do elementów oprogramowania oraz obiektów biznesowych dla określonych wcześniej użytkowników. Rozwiązanie to pozwala na wyodrębnienie cykli wytwarzania oraz wdrażania oprogramowania, a co za tym idzie programiści mogą mieć niewielką wiedzę na temat kwestii wdrażania. OES został zaprojektowany tak, by wspierał różne technologie oraz aby mógł sprostać wymaganiom dotyczącym wydajności i skalowalności dużych oraz złożonych systemów. Oracle Entitlements Sever zapewnia bogaty hierarchiczny model polityk, bazujący zarówno na standardzie Role-Based Access Control (RBAC) jak i na standardzie Attribute-Based Access Control (ABAC) [2]. Wspiera on także wielopoziomową delegowaną administrację politykami, co pozwala na dokładną kontrolę tworzenia i zarządzania politykami bezpieczeństwa.

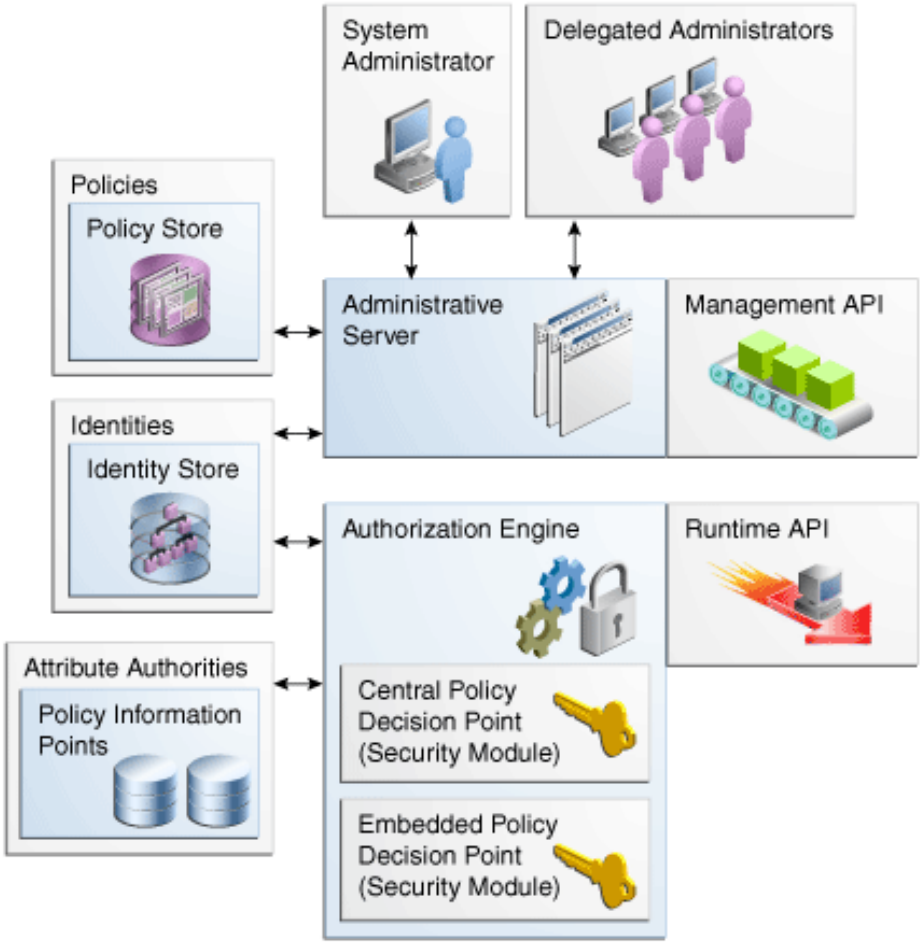


Rys. 1. Koncepcja Oracle Entitlements Server [2]
Fig. 1. An overview of Oracle Entitlements Server

Na rysunku 1 przedstawiono ogólną koncepcję Oracle Entitlements Server. Użytkownicy, w ramach swoich normalnych czynności, takich jak na przykład przeglądanie stron WWW, generują określone żądania dostępu. OES przekształca te żądania na postać znormalizowaną i sprawdza poprawność z ustalonymi politykami autoryzacji. Podczas oceny polityki OES może wykorzystywać informacje pobrane z zewnętrznych źródeł, takich jak: bazy danych, web services oraz narzędzia wspierające LDAP. Ostatnim krokiem podczas weryfikacji dostępu jest wysłanie odpowiedzi zwrotnej do użytkownika, który to żądanie wygenerował.

1.2. Architektura komponentów

Architektura OES bazuje na modelu, w którym zasadnicze znaczenie odgrywają interakcje pomiędzy poszczególnymi jednostkami, komunikującymi się na podstawie XACML (eXtensible Access Control Markup Language) [5]. Model ten zapewnia elastyczną architekturę, którą, zależnie od potrzeb przystosować można do różnych środowisk.



Rys. 2. Komponenty Oracle Entitlements Server [3]
Fig. 2. Components of Oracle Entitlements Server

Funkcje poszczególnych komponentów realizowane są przez różne narzędzia, np. polityki definiowane przez administratorów przechowywane są w bazie danych (Policy Store). W standardzie XACML, który jest wykorzystywany przez OES, wyodrębnione są 4 logiczne elementy, których funkcje rozłożone są pomiędzy poszczególne elementy Oracle Entitlements Server [6]:

1. Punkt Administrowania Politykami (ang. Policy Administration Point – PAP).
2. Punkt Decyzji Polityk (ang. Policy Decision Point – PDP).
3. Punkt Egzekwowania Polityk (ang. Policy Enforcement Point – PEP).
4. Punkt Informacji o Politykach (ang. Policy Information Point – PIP).

Punkt Administrowania Politykami odpowiada za tworzenie i zarządzanie zasadami (politykami) stosowanymi do ochrony określonego zasobu. Dostęp do serwera realizowany jest zazwyczaj poprzez przeglądarkę internetową. PAP umożliwia dostęp polityk dla Punktu Decyzyjnego (PDP) w celu wygenerowania decyzji *zezwól* lub *odmów* dla żądania dostępu do chronionego zasobu. PAP w Oracle Entitlements Server składa się z konsoli administracyjnej, interfejsów programowania aplikacji (API) do zarządzania oraz narzędzi umożliwiają-

cych zarządzanie z poziomu linii komend. Tworzone i modyfikowane polityki zapisywane są w przestrzeni polityk (Policy Store).

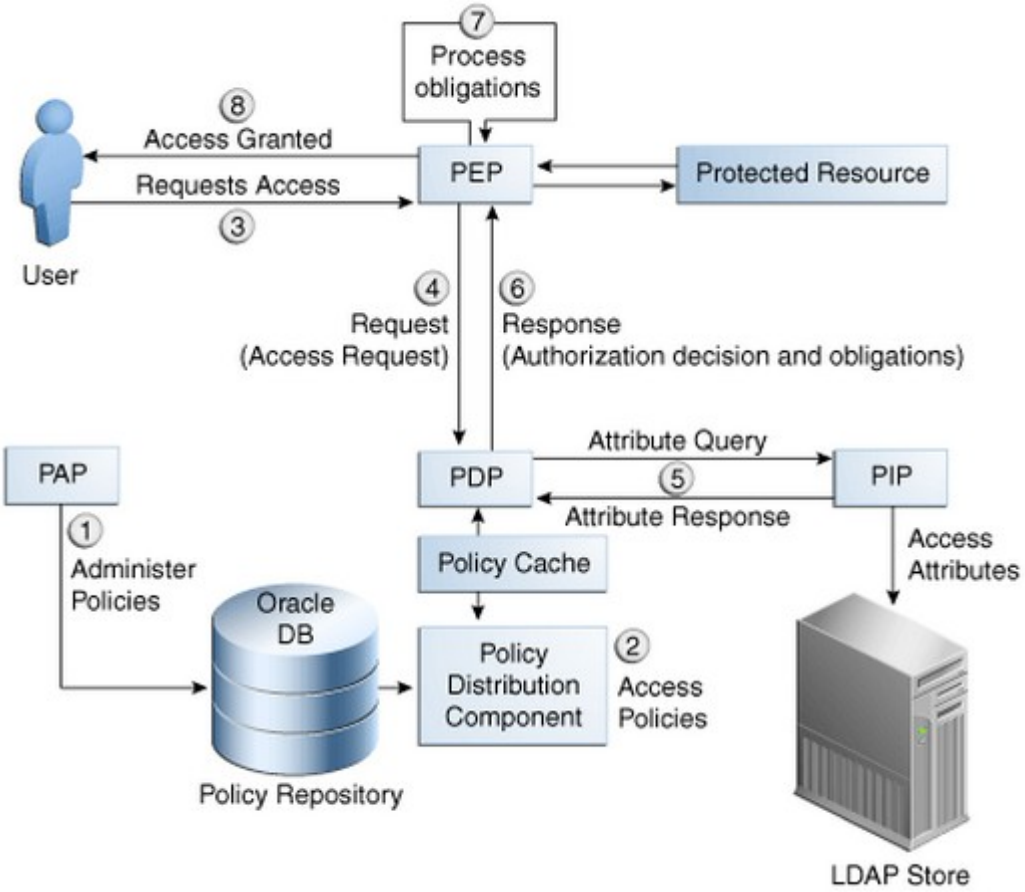
Punkt Decyzji Polityk otrzymuje żądanie autoryzacji, następnie ocenia je na podstawie zdefiniowanych polityk, podejmuje i przesyła decyzję do Punktu Egzekwowania Polityk w celu jej wykonania. Punkt Egzekwowania Polityk wprowadza w życie otrzymane od Punktu Decyzji informacje. PEP jest elementem oprogramowania, który przechwytuje żądanie dostępu użytkownika do chronionej aplikacji, przekazuje je do PDP, a następnie przetwarza i egzekwuje otrzymane od PDP informacje. Punkt Egzekwowania Polityk może być realizowany zarówno na poziomie chronionej aplikacji, jak również poprzez moduł bezpieczeństwa (Security Module). PEP jest zawsze zintegrowany poprzez aplikację w technologii Java Standard Enterprise (JSE) lub webowy kontener Java Enterprise Edition (JEE). Oracle Entitlements Server umożliwia realizację modułu bezpieczeństwa na dwa sposoby. Pierwszy z nich zakłada funkcjonowanie wyłącznie jako PDP poprzez odbieranie żądań i podejmowanie decyzji. Drugi natomiast sposób łączy w jednym narzędziu funkcje PDP oraz PEP.

Punkt Informacji o Politykach jest natomiast pewnego rodzaju repozytorium danych – źródłem, z którego potrzebne informacje mogą być pobierane do użytku przy ocenie polityki podczas procesu podejmowania decyzji o autoryzacji. Rozwiązanie to pozwala, aby polityki były sterowane poprzez dane w sytuacji, gdy pewien atrybut może mieć wpływ na decyzję dostępu. Wartość konkretnego wymaganego atrybutu jest pobierana z miejsca jego przechowywania.

Na rysunku 3 przedstawiono sposób przepływu informacji podczas procesu podejmowania decyzji o autoryzacji [3]:

1. Oracle Entitlements Server (działający jako PAP) używany jest do tworzenia oraz administrowania politykami ochrony zasobu.
2. Zasady dostępu zapisane w repozytorium polityk (Policy Repository) przesyłane są poprzez usługę dystrybucji (Policy Distribution Service) do tymczasowej przestrzeni przechowującej (Policy Cache). Punkt Decyzyjny odczytuje te informacje bezpośrednio z przestrzeni tymczasowej.
3. Żądanie dostępu do chronionego zasobu odbierane jest przez Punkt Egzekwowania Polityk. PEP może być realizowany poprzez mechanizmy zawarte w aplikacji klienckiej lub poprzez moduł bezpieczeństwa zaimplementowany na serwerze.
4. PEP przesyła żądanie autoryzacji do PDP.
5. PDP wysyła zapytanie do PIP odnośnie dodatkowego przedmiotu, źródła, akcji oraz atrybutów środowiskowych.
6. Punkt Decyzyjny przetwarza żądanie, a następnie wysyła odpowiedź (wraz z ewentualnymi dodatkowymi informacjami, np. powód odmowy) w formie decyzji (zezwól/odmów) odnośnie autoryzacji dostępu do Punkty Egzekwowania.

7. PEP wypełnia wszystkie zobowiązania, które są związane z podjętą decyzją (np. informacje dotyczące decyzji o odmowie mogą mieć dodatkowe informacje, które również są przetwarzane przez PEP).
8. Jeśli następuje zgoda, PEP przyznaje użytkownikowi dostęp do żądanego zasobu, w przeciwnym wypadku dostęp jest zabroniony.



Rys. 3. Przepływ informacji podczas procesu autoryzacji żądania [3]
Fig. 3. Data Flow in the Policy Authorization Process

1.3. Instalacja i konfiguracja Oracle Entitlements Server

Instalację Oracle Entitlements Server 11gR1 można podzielić na dwa etapy. Pierwszym z nich jest zainstalowanie oraz skonfigurowanie serwera pełniącego funkcję serwera administracyjnego, a drugim jest instalacja osobnego klienta, który będzie realizował funkcje modułu bezpieczeństwa [4].

Serwer administracyjny jest jednym z elementów pakietu oprogramowania Oracle Identity and Access Management. Oprócz omawianego narzędzia w skład pakietu Oracle Identity and Access Management wchodzą również:

- Oracle Identity Manager (OIM),
- Oracle Access Manager (OAM),

- Oracle Adaptive Access Manager,
- Oracle Identity Navigator,
- Oracle Security Token Service.

Komponenty konieczne do instalacji Oracle Entitlements Server:

- Oracle Database,
- Oracle Repository Creation Utility (RCU),
- Oracle WebLogic Server,
- Serwer LDAP – Oracle Internet Directory.

Do instalacji narzędzi odpowiadających za zarządzanie uprawnieniami konieczna jest baza danych, w której przechowywane będą tworzone polityki prywatności. Oracle rekomenduje swój własny serwer Oracle Server, wpiera jednak również rozwiązania innych firm (MS SQL Server, Sybase, IBM DB2). Po instalacji serwera baz danych konieczne jest utworzenie repozytorium za pomocą Oracle Repository Creation Utility. WebLogic pełni rolę serwera aplikacji, w ramach którego funkcjonuje interfejs OES. Standardowo rolę serwera LDAP pełni Oracle Internet Directory (element pakietu Oralce Identity Management).

Po zakończonej instalacji pakietu Oracle Identity and Access Managemet, w celu skonfigurowania narzędzia Oracle Entitlements Server konieczna jest modyfikacja pliku `weblogic.policy`:

```
IDM_HOME/common/bin/wlst.sh IDM_HOME/oes/modifygrants.py
```

Dalsze czynności:

1. Uruchomienie skryptu wlst.sh (znajdującego się w IDM_HOME/common/bin).
2. Połączenie się z serwerem administracyjnym:

```
connect('weblogic-username', 'weblogic-password','t3://host:port')
```

3. Wykonanie polecenia:

```
configureOESAdminServer(servertype="DB_ORACLE");
```

4. Sprawdzenie dostępu do serwera administracyjnego (domyślny numer portu to 7001):

```
http://hostname:port/apm/
```

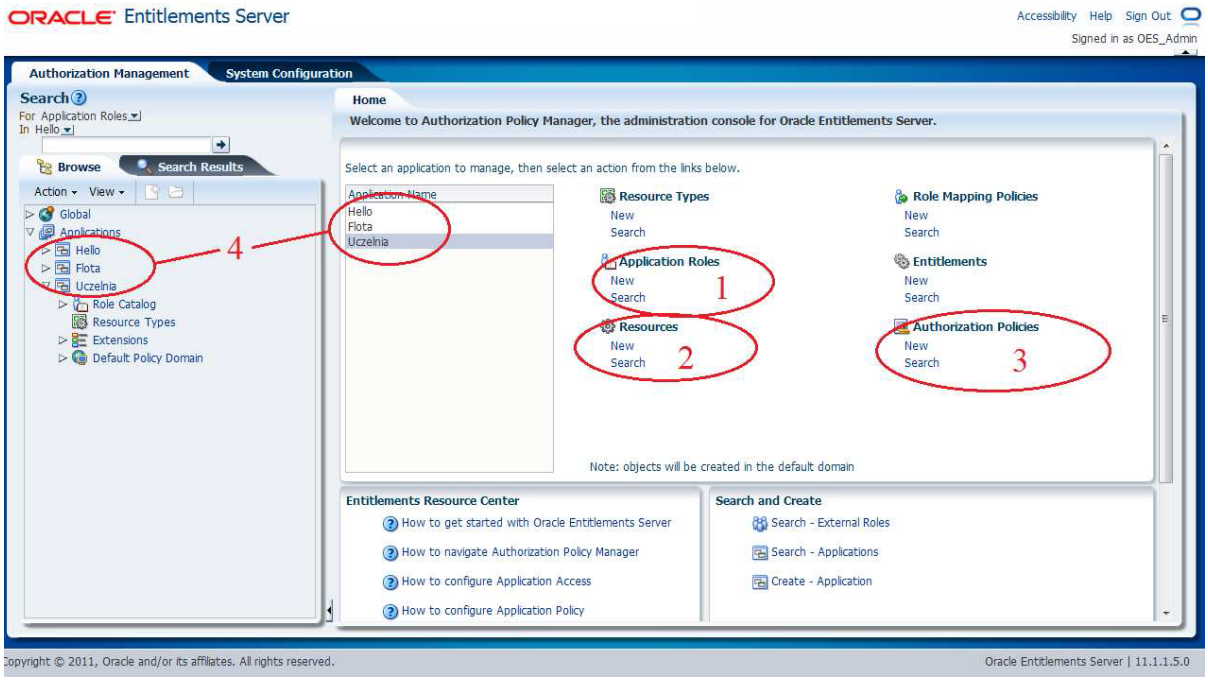
2. Przykład wykorzystania Oracle Entitlements Server

Elementy Oracle Entitlements Server przedstawione zostaną na przykładzie zarządzania uprawnianiami pracowników uczelni. Każdy użytkownik systemu przypisany jest do konkretnej roli aplikacyjnej. Oracle Entitlements Server umożliwia definiowanie ról jednostkowych oraz ról grupowych. W przypadku wykonywania pewnych czynności przez wszystkich

pracowników (np. przeglądanie WWW), można przypisać dostęp do tej właśnie funkcji użytkownikowi grupowemu, a następnie dodać poszczególnych pracowników do tej grupy.

W sytuacji zmiany reguł lub w przypadku konieczności przejęcia pewnych czynności przez innego pracownika (np. gdy jeden z pracowników zachoruje i jego obowiązki muszą być pełnione przez inną osobę z firmy), możliwe jest natychmiastowe dodanie konkretnego użytkownika do odpowiedniej roli w systemie. Mechanizmy te umożliwiają wprowadzanie zmian reguł bezpieczeństwa bez jakiejkolwiek ingerencji w kod aplikacji, a co za tym idzie nie trzeba jej testować oraz oczywiście mamy do czynienia z oszczędnością kosztów. Informacje o użytkownikach systemu przechowywane są na serwerze usług katalogowych OpenLDAP.

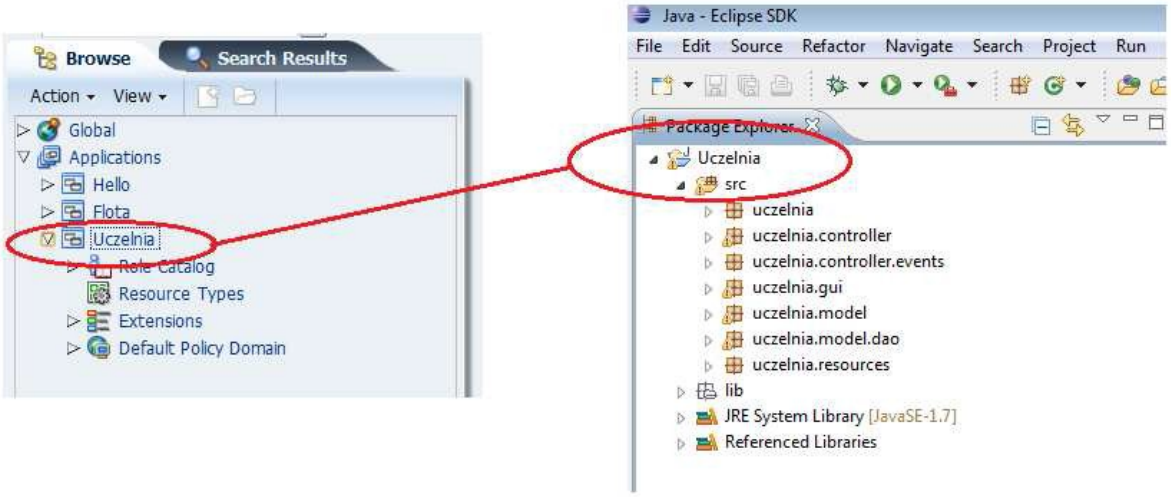
Do tworzenia i zarządzania politykami bezpieczeństwa w narzędziu Oracle Entitlements Server służy konsola administracyjna, do której dostęp mamy przez przeglądarkę internetową. Aby połączyć się z konsolą administracyjną, konieczne jest uruchomienie wcześniej serwera administracyjnego. Panel służący do zarządzania zasobami aplikacji został przedstawiony na rysunku 4 – można na nim wyróżnić m.in.: Role aplikacyjne (1), Zasoby (2), Polityki autoryzacji (3) oraz listę aplikacji zarządzanych przez OES (4).



Rys. 4. Konsola administracyjna Oracle Entitlements Server
Fig. 4. The administration console of Oracle Entitlements Server

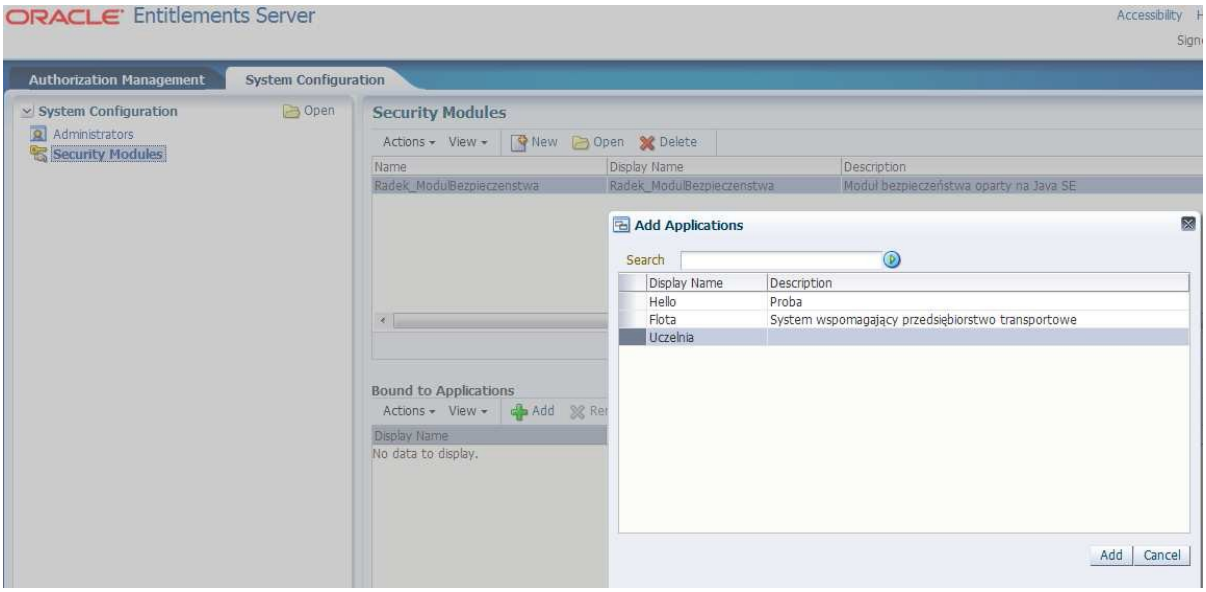
Pierwszym krokiem do wykorzystania mechanizmów bezpieczeństwa narzędzia Oracle Enitlements Server w naszym systemie jest utworzenie aplikacji w panelu administracyjnym (rysunek 5). Wymagane jest, aby nazwa aplikacji zdefiniowanej w Oracle Entitlements Server była taka sama jak nazwa rzeczywistej aplikacji.

Kolejnym elementem koniecznym do poprawnego działania systemu jest przypisanie utworzonej aplikacji do skonfigurowanego wcześniej modułu bezpieczeństwa. Również w tym przypadku moduł bezpieczeństwa (PEP) oraz ten zdefiniowany w konsoli administracyjnej muszą mieć takie same nazwy.



Rys. 5. Tworzenie aplikacji w panelu administracyjnym
Fig. 5. Creating an application in the administrative panel

Po zdefiniowaniu modułu bezpieczeństwa należy powiązać z nim aplikację, w której będziemy chcieli zarządzać politykami bezpieczeństwa – rysunek 6.



Rys. 6. Powiązanie aplikacji z modułem bezpieczeństwa
Fig. 6. A connection between an application and a security module

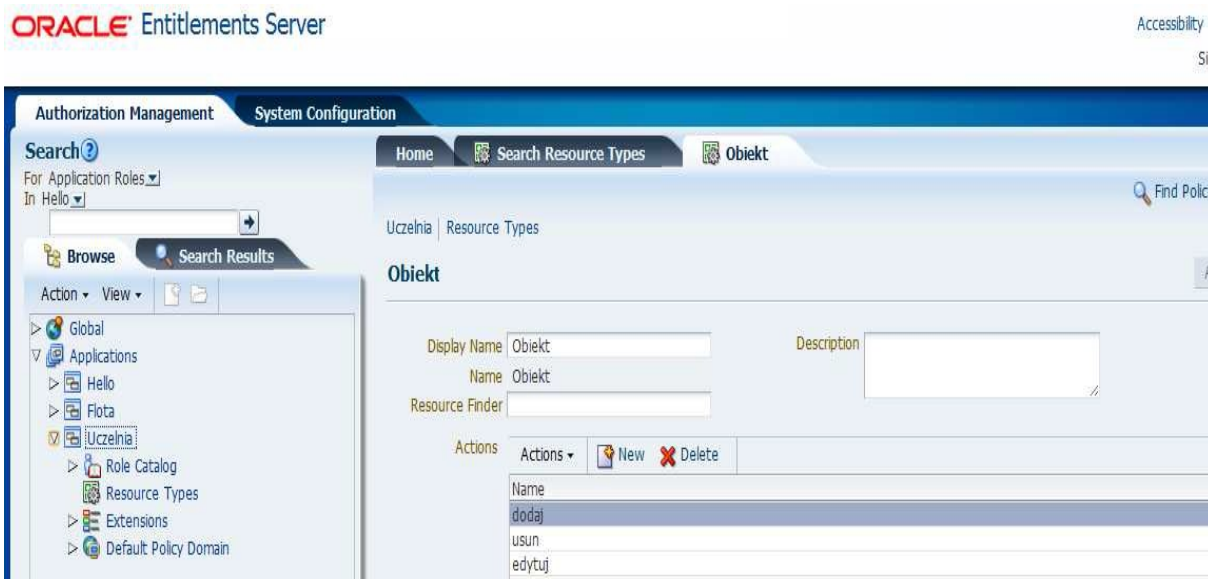
Po powiązaniu aplikacji z odpowiednim modułem bezpieczeństwa można utworzyć, a następnie wyeksportować, polityki do modułu bezpieczeństwa, w celu ich egzekwowania.

Etapy tworzenia polityki autoryzacji:

1. Zdefiniowanie typów zasobów – rysunek 7.

2. Zdefiniowanie zasobów – rysunek 8.
3. Zdefiniowanie ról aplikacyjnych.
4. Zdefiniowanie reguł dostępu – rysunek 9.
5. Zdefiniowanie polityki autoryzacji (powiązanie roli oraz reguły dostępu) – rysunek 10.

Typ zasobu reprezentuje typ chronionego obiektu. Typy zasobów mogą być bardzo różnorodne, może być to zarówno plik, jak i na przykład konto bankowe (w aplikacji bankowej). Podczas definiowania typu zasobu określane są akcje (czynności), jakie można dokonywać na danym obiekcie. Biorąc pod uwagę plik, możemy z niego czytać lub do niego zapisywać, tak więc akcjami będą zapis i odczyt. Z kolei rozpatrując sytuację, w której chronionym elementem ma być konkretny obiekt, akcjami, które możemy wykonać, będą: dodaj, edytuj, usuń oraz listuj.



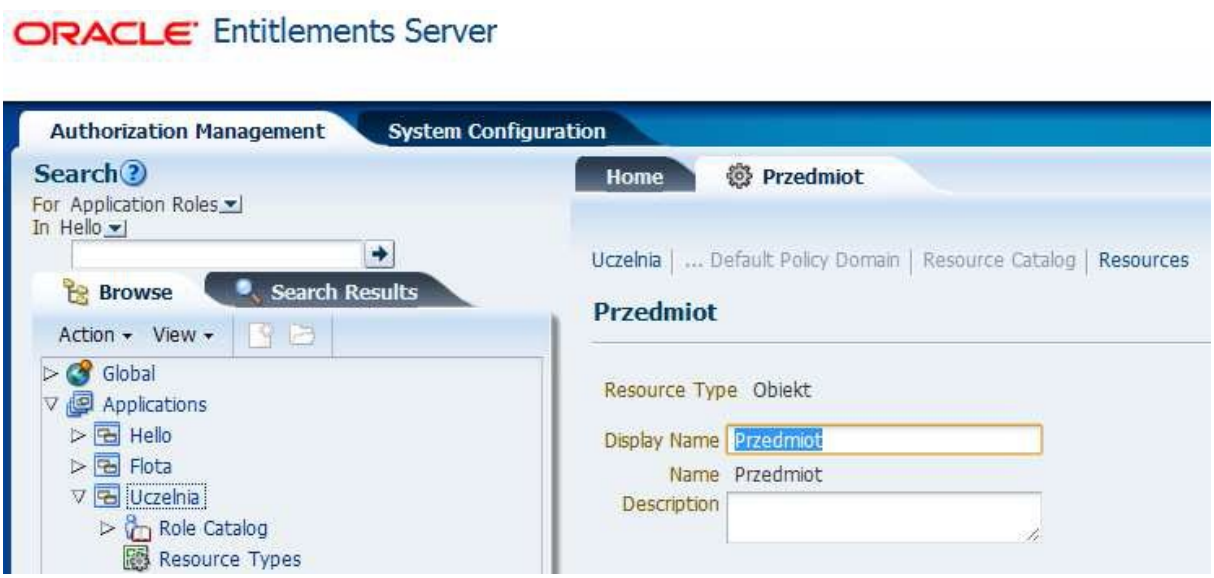
Rys. 7. Tworzenie typu zasobu
Fig. 7. Creating a type of resource

Zasobem jest konkretny chroniony obiekt, do którego możemy uzyskać zezwolenie lub zakaz dostępu. Może on być reprezentacją zarówno obiektu biznesowego, jak i elementu aplikacji, do którego będzie konieczne uzyskanie dostępu. Podczas tworzenia zasobu konieczne jest przypisanie go do istniejącego już typu – rysunek 8.

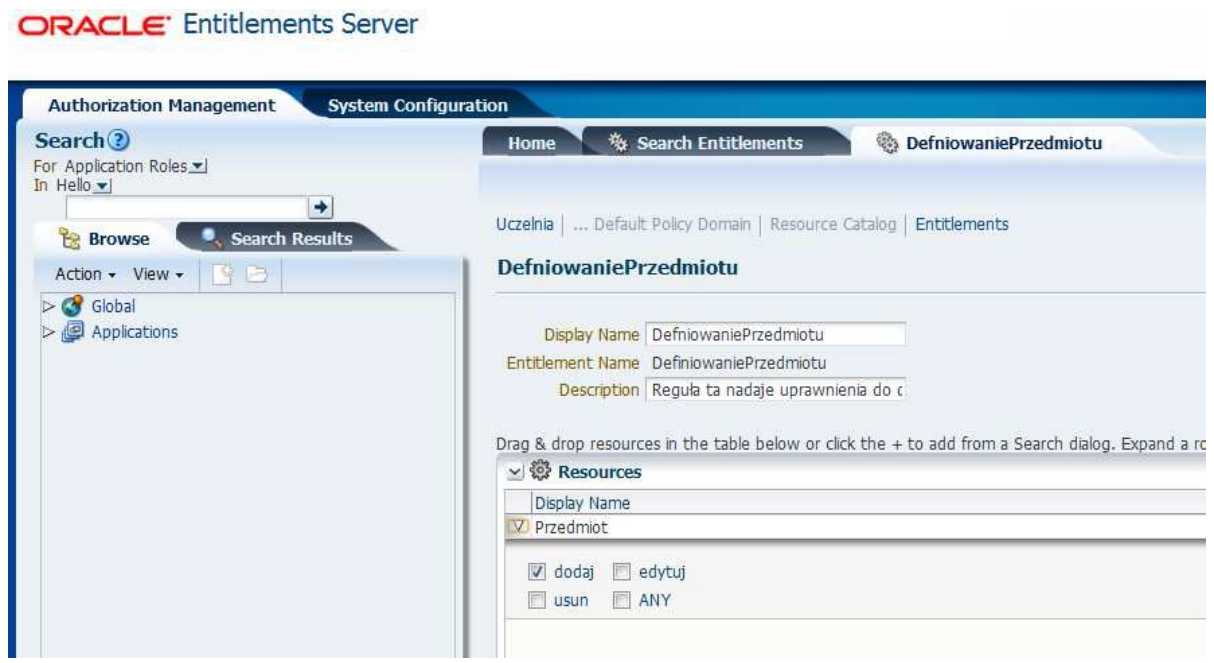
Następnym etapem jest utworzenie roli. Rola aplikacyjna jest reprezentacją użytkowników, którzy będą mieli dostęp do konkretnego zasobu, np. pliku.

Kolejny krok dotyczy reguł dostępu. Reguła dostępu określa, jakie operacje możemy wykonywać na konkretnym zasobie. Rozpatrując nasz przykład zasobem jest Przedmiot, a akcją, która będzie wykonywana, jest Dodaj – rysunek 9.

Ostatnim krokiem do stworzenia polityki autoryzacji jest powiązanie zdefiniowanych wcześniej roli aplikacyjnej oraz reguły dostępu. To właśnie w tym kroku definiowane jest to, kto i co może dokładnie wykonywać.



Rys. 8. Tworzenie zasobu
Fig. 8. Creating a resource

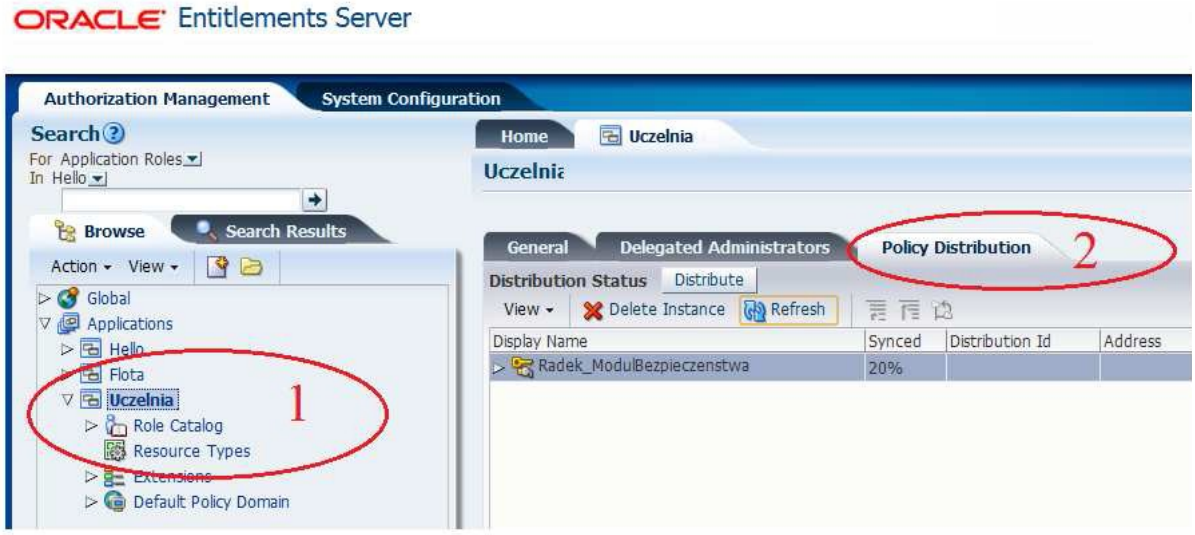


Rys. 9. Definiowanie reguły dostępu
Fig. 9. Defining an access rule

Po zdefiniowaniu zbiory reguł należy wyeksportować do przypisanego do aplikacji modułu bezpieczeństwa, aby reguły stały się dostępne dla aplikacji – rysunek 10. Każdą zmianę (czy to usunięcie, czy modyfikację) należy wyeksportować, aby aplikacja działała w zaplanowany sposób.

Po wyeksportowaniu polityk do modułu bezpieczeństwa możliwe jest już korzystanie ze zdefiniowanego zbioru reguł w aplikacji.

Ponieważ kwestia zarządzania tożsamością jest problemem złożonym, istnieją również inne możliwe podejścia do tego zagadnienia. W kolejnym rozdziale przestawione zostały dwa środowiska, które mogą stanowić alternatywę dla OES. Mają one różną złożoność i nie koniecznie wykorzystują te same elementy architektury, umożliwiając w ten sposób użytkownikowi wybór najbardziej adekwatnego rozwiązania dla konkretnego wdrożenia.



Rys. 10. Powiązanie aplikacji z modułem bezpieczeństwa
Fig. 10. Creating a connection between an application and a security module

3. Rozwiązania alternatywne

3.1. IBM Tivoli Security Policy Manager

Firma IBM oferuje produkt o możliwościach zbliżonych do Oracle Entitlements Server – Tivoli Security Policy Manager (TSPM). Tivoli Security Policy Manager zawiera cztery powiązane ze sobą komponenty: PAP, PDP, PEP oraz PIP [7]:

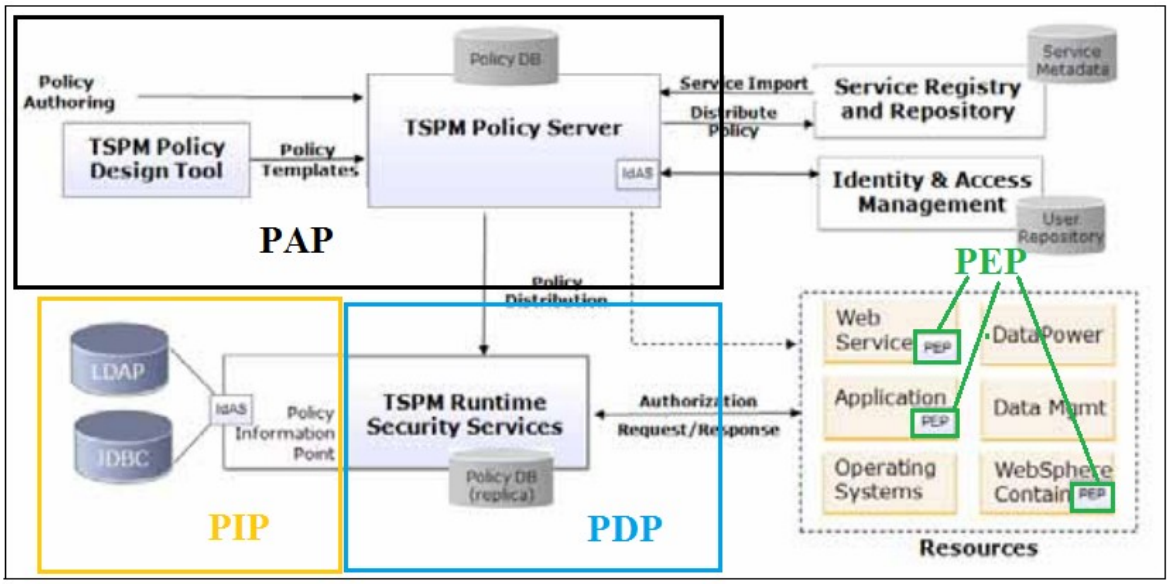
- Punkt Administrowania Politykami realizowany jest poprzez serwer polityk (TSPM Policy Server/Console).
- Punkt Decyzyjny realizowany jest poprzez usługę bezpieczeństwa (TSPM Runtime Security Service).
- Punkt Egzekwowania Polityk realizowany jest jako część aplikacji lub odrębny kontener (TSPM Plug-in).
- Jako Punkt Informacyjny służą w tym przypadku zewnętrzne źródła, takie jak: bazy danych, narzędzia implementujące protokół LDAP itp.

Narzędzie to opiera się na architekturze łatwo skalowalnej, która zawiera w sobie usługę IdAS (Identity Attribute Service) w celu zapewniania integralności z istniejącymi mechani-

zmami zarządzania tożsamością, istniejącymi repozytoriami tożsamości i atrybutów, silnikami reguł itp. IBM TSPM opiera się na architekturze Eclipse Plug-in.

Narzędzie do tworzenia polityk bazuje na Eclipse Rich Client Platform (Eclipse RCP) i oferuje dużą liczbę różnych widoków do modelowania, analizowania oraz debugowania polityk kontroli dostępu. Narzędzie to ma wbudowany PDP oparty na XACML, dzięki któremu można zasymulować żądanie dostępu oraz przetworzenie decyzji, tak jak wyglądałoby to w normalnym środowisku [8].

Serwer TSPM umożliwia zarówno importowanie definicji oraz metadanych z zewnętrznych repozytoriów (m.in. rejestry usług), jak i korzystanie z istniejących już systemów identyfikacji oraz kontroli dostępu. Polityki mogą być konstruowane zarówno poprzez narzędzie do tworzenia polityk zawarte w IBM TSPM, jak i poprzez import polityk do Tivoli Security Policy Manager jako szablonów. Usługa Bezpieczeństwa (Runtime Security Services) może także zostać zintegrowana z istniejącymi już źródłami tożsamości i atrybutów oraz silnikami reguł [11].

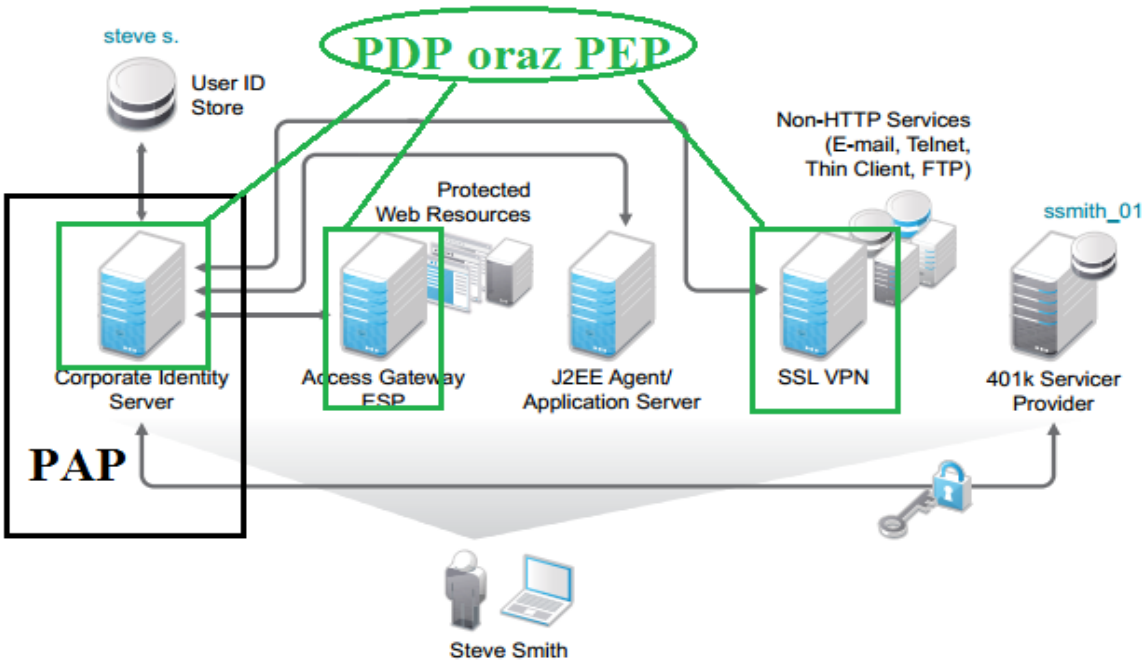


Rys. 11. Komponenty IBM Tivoli Security Policy Manager [7]
Fig. 11. Components of IBM Tivoli Security Policy Manager

3.2. Novell Access Manager

Novell Access Manager, podobnie jak analogiczne rozwiązania, realizuje, poprzez swoje komponenty, poszczególne funkcje standardu XACML (tj. PAP, PDP, PEP oraz PIP) [13]. W przeciwieństwie do rozwiązań Oracle Entitlements Server oraz IBM Tivoli Security Manager, narzędzie Novell Access Manager nie umożliwia jednoznacznego wyodrębnienia elementów składowych, takich jak: Punkt Decyzyjny, Punkt Egzekwowania Polityk oraz Punkt Informacyjny. Jedynie w pełni autonomiczną jednostką jest Punkt Administracyjny, którego założenia realizowane są poprzez Serwer Tożsamości (Corporate Identity Server) – rysunek 12.

Polityki definiowane są w Serwerze Tożsamości, natomiast ich egzekwowanie odbywa się w różnych komponentach oprogramowania, tj. dostęp do zasobów opartych na Internecie uzyskiwany jest dzięki bramom dostępu, natomiast dostęp do źródeł zlokalizowanych poza zaporą sieciową realizowany jest poprzez serwer SSV/VPN.



Rys. 12. Koncepcja Novell Access Manager [10]
Fig. 12. A conception of Novell Access Manager

Novell Access Manager umożliwia standardowy dla tego typu rozwiązań schemat tworzenia oraz administrowania politykami. Po zebraniu potrzebnych informacji oraz zaprojektowaniu reguł przez architektów są one definiowane w systemie dzięki administratorom, poprzez serwer tożsamości (PAP). Jednak kolejne kroki różnią się nieco od sposobu tworzenia polityk, chociażby w Oracle Entitlements Server. W narzędziu OES dopiero po kompletnym utworzeniu lub wprowadzeniu zmian w zestawie reguł zostają one wyeksportowane do komponentów, jakimi są moduły bezpieczeństwa (SMs), natomiast w NAM przewidziano nieco inny schemat postępowania. Uwzględniając fakt, że różne komponenty zapewniają dostęp do innych zasobów organizacji (tj. bramy dostępu – zasoby internetowe, mechanizmy SSL/VPN – zasoby poza zaporą sieciową), na samym początku definiowany jest konkretny punkt PEP, który będzie kontrolowany przez tę regułę. Podejście takie pozwala na zminimalizowanie listy opcji dla poszczególnych reguł, ponieważ po przypisaniu reguły do konkretnego komponentu wyświetlane są tylko te wartości i funkcje, które realizowane są w danym Punkcie Egzekwowania Polityk.

Podczas przydzielania wybranej reguły już do konkretnego komponentu może zostać wykonany audyt, dzięki któremu możliwe jest sprecyzowanie, które Punkty Egzekwowania Polityk będą najbardziej kompatybilne dla tej właśnie polityki.

Głównym elementem architektury narzędzia Novell Access Manager (NAM) jest Serwer Tożsamości, który zapewnia usługi uwierzytelniania na rzecz wszystkich komponentów pakietu NAM. Rolę interfejsu umożliwiającego tworzenie oraz administrowanie politykami oraz komponentami Novell Access Manager jest narzędzie Manager, stanowiące centralny punkt komunikacji w środowisku Novell, a wraz z Serwerem Tożsamości realizuje funkcję Puntu Administrowania Politykami.

4. Podsumowanie

Omawiane narzędzia różnią się między sobą, co można zauważyć na wielu płaszczyznach. Pierwsza z różnic dotyczy wspieranych elementów oprogramowania, koniecznych do wdrożenia całego środowiska (m.in. serwery aplikacji, bazy danych). Kolejne różnice przejawiają się w samej idei zarządzania uprawnieniami oraz zaimplementowanych standardach autoryzacji.

W tabeli 1 zawarto porównanie omawianych narzędzi w zakresie współpracy z różnymi serwerami aplikacji, natomiast w tabeli 2 przedstawiono, które bazy danych mogą służyć jako przestrzenie do przechowywania polityk autoryzacji dla poszczególnych narzędzi.

Tabela 1

Serwery aplikacji współpracujące z poszczególnymi narzędziami

	Oracle WebLogic	IBM WebSphere	Apache Tomcat
OES	X	X	X
IBM TSPM		X	
NAM	Nie wykorzystuje serwera aplikacji		

Tabela 2

Zestawienie baz danych

	Oracle Database	Sysbase	Microsoft SQL Server	IBM DB2	Apache Derby
OES	X	X	X	X	
IBM TSPM				X	X
NAM	Nie wykorzystuje zewnętrznej bazy danych jako przestrzeni polityk				

Spośród omawianych narzędzi produkt firmy Novell charakteryzuje się podejściem nieco odbiegającym od pozostałych systemów w kwestii definiowania chronionych zasobów. W przeciwieństwie do Oracle Entitlments Server oraz IBM Tivoli Policy Security Manager, Novell Access Manager nie zapewnia korzystania ze szczegółowej autoryzacji.

Płaszczyzną, na której firma Novell zyskuje znaczną przewagę nad konkurentami, jest łatwość wdrożenia tego narzędzia. Do wdrożenia oraz uruchomienia Novell Access Manager nie potrzeba serwera aplikacji oraz bazy danych, która w pozostałych rozwiązaniach wykorzystywana jest jako przestrzeń polityk. Rozwiązanie to w znacznym stopniu zmniejsza koszty oraz stopień złożoności wdrożenia oraz administrowania systemem opartym na produkcie firmy Novell.

Tabela 3

Zaimplementowane standardy autoryzacji

	XACML 3.0	XACML 2.0	OpenAZ PEP Decision API	JAAS 1.4	SAML 2.0	WS-SecurityPolicy 1.2
OES	X	X	X	X		
IBM TSPM	X	X				X
NAM					X	

Atutem przemawiającym za Oracle Entitlements Server jest jego najbardziej rozbudowana funkcjonalność oraz możliwość integracji z wieloma elementami oprogramowania konkurencyjnych firm. Jak przedstawiono w tabelach 1 i 2, OES może zostać wdrożony w środowisku różnych producentów oprogramowania (np. serwer aplikacji firmy IBM, system zarządzania bazą danych firmy Microsoft).

BIBLIOGRAFIA

1. Scheidel J.: Designing an IAM Framework with Oracle Identity and Access Management Suite. Osborne ORACLE Press Series, 2010.
2. Fine Grained Authorization: Technical Insights for Using Oracle Entitlements Server, http://www.oracle.com/technetwork/middleware/oes/oes-product-white-paper-405854.pdf.
3. Introducing Oracle Entitlements Server, http://docs.oracle.com/cd/E21764_01/doc.1111/e14096/intro.htm#sthref11.
4. Installing and Configuring Oracle Entitlements Server, http://docs.oracle.com/cd/E21764_01/install.1111/e12002/oes.htm.
5. Oracle Entitlements Server, http://www.oracle.com/us/products/middleware/identity-management/059385.pdf.
6. Anderson A.: OASIS eXtensible Access Control Markup Language (XACML), http://labs.oracle.com/projects/xacml/XMLCOP_060620_slides.pdf.
7. Buecker A., Forster C., Muppidi S., Safabakhsh B.: IBM Tivoli Security Policy Manager, http://www.redbooks.ibm.com/redpapers/pdfs/redp4483.pdf.

8. IBM Tivoli Security Policy Manager. Centralize security policy management and fine-grained data access control, http://public.dhe.ibm.com/common/ssi/ecm/en/tid14029-usen/ TID14029USEN.PDF.
9. Novell Access Manager. Kontrola dostępu, zarządzanie regułami, zapewnienie zgodności, http://www.novell.com/poland/resourcecenter/Novell_Access_Manager_Broszura_informacyjna.pdf
10. Novell Access Manager. Installation Guide, http://www.novell.com/documentation/novellaccessmanager31/pdfdoc/installation/installation.pdf.
11. Buecker A. et. al: IT Security Policy Management Usage Patterns Using IBM Tivoli Security Policy Manager, http://www.redbooks.ibm.com/redbooks/pdfs/sg247880.pdf.
12. IBM Security Policy Manager Version 7.1 Installation Guide, http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tspm.doc_7.1/PDF/tspm_install_pdf.pdf.
13. Novell Access Manager Administration Guide, http://www.novell.com/documentation/novellaccessmanager/pdfdoc/adminguide/adminguide.pdf.
14. IBM Tivoli Security Policy Security Manager Version 7.1 Administration Guide, http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tspm.doc_7.1/PDF/ tspm_admin_pdf.pdf.

Wpłynęło do Redakcji 14 stycznia 2013 r.

Abstract

Traditional approach to security directly within application code causes many problems. These problems are: necessity to test each change of rules, inevitability to quick response to threats and security violation, difficulty in analyzing and controlling security policies and authorization decisions. Moreover, such a type of solutions slows down the development of applications and increases implementation costs. Thanks to using external authorization mechanisms in complex systems we can simplify security management.

The article presents Oracle Entitlements Server (OES), which enable to secure internal enterprise resources through definition and management policies, that control the access and give possibility to use their resources. Oracle Entitlements Sever provides a rich hierarchical model of policies which is based on Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) standards.

IBM Tivoli Security Policy Manager and Novell Access Manager were presented for comparison. IBM Tivoli Security Policy Manager is based on Eclipse Rich Client Platform

(Eclipse RCP) and offers a large number of different views for modeling, analyzing and debugging access control policies. Novell Access Manager is based on XACML standard. In contrast to Oracle Entitlements Server and IBM Tivoli Security Manager, Novell Access Manager does not visibly separate components, such as Policy Decision Point, Policy Enforcement Point, Policy Information Point.

These tools differ from each other in the supported software components needed to implement the environment: application servers (Table 1) and database servers (Table 2). Moreover, there are differences in the idea of entitlements management and implemented authorization standards (Table 3).

Adresy

Krzysztof CZAJKOWSKI: Politechnika Krakowska, Wydział Fizyki, Matematyki i Informatyki, Instytut Teleinformatyki, ul. Warszawska 24, 31-155 Kraków, Polska, kc@pk.edu.pl.
Radosław KORKOSZ: Politechnika Krakowska, Wydział Fizyki, Matematyki i Informatyki, Instytut Teleinformatyki, ul. Warszawska 24, 31-155 Kraków, Polska, radoslaw.korkosz@gmail.com.