Зборник радова Факултета техничких наука, Нови Сад

УДК: 007:004.056
ДОИ: https://doi.org/10.24867/20BE26Blagojevic

ИМПЛЕМЕНТАЦИЈА КОНТРОЛЕ ПРИСТУПА КОРИШЋЕЊЕМ *APACHEDS* И *APACHE FORTRESS*

IMPLEMENTING ACCESS CONTROL USING APACHEDS AND APACHE FORTRESS

Ивана Благојевић, *Факултет техничких наука, Нови Сад*

Област – РАЧУНАРСТВО И АУТОМАТИКА

Кратак садржај – *Рад приказује намену ApacheDS сервера и описује имплементацију RBAC контроле приступа коришћењем Apache Fortress на примеру електронског дневника. Наведене су теоријске основе за поменуте технологије. Приказан је модел имплементираног решења. Објашњене су функционалности.*

Кључне речи: *LDAP сервер, контрола приступа, RBAC, Apache Fortress*

Abstract – *This paper explains the purpose of the ApacheDS server and describes the implementation of RBAC access control using Apache Fortress. The theoretical foundations for the mentioned technologies are stated. A model of the implemented solution is presented. The functionalities are explained.*

Keywords: *LDAP server, access control, RBAC, Apache Fortress*

1. УВОД

Комплексност архитектуре информационих система константно расте, док је развој решења временски ограничен. Изоловањем компоненти система сваки сервис представља одвојену функционалну целину. Централизованом контролом приступа кроз *LDAP* сервер постиже се бржи развој, не понављајући имплементацију за сваки сервис појединачно, тиме редукујући могућност неисправне аутентификације и ауторизације.

2. *LDAP* СЕРВЕРИ

LDAP (*Lightweight Directory Access Protocol*) [1] је софтверски протокол за чување и уређивање података тако да они буду ефикасно претраживи. *LDAP* протокол се користи за комуникацију сервера са директоријумима. Подаци се чувају у хијерархијском информационом стаблу *DIT* (*Directory Information Tree*), који податке смешта у гране стабла, омогућавајући администраторима да се лакше крећу кроз директоријуме ради проналажења података [2].

LDAP представља стандардни апликативни протокол који дефинише интерфејс између клијентских апликација које могу да комуницирају са директоријумским сервисима попут *Active Directory*-а или *OpenLDAP*-а [3].

Термин *LDAP* у пракси се често односи осим на сам протокол и на директоријум, односно сервер. Главна улога *LDAP* протокола је опслуживање централног чвора за ауторизацију и аутентификацију, чувајући информације о корисницима, корисничке креденцијале, податке о групама, корисничким улогама и слично. *LDAP* директоријуми су оптимизовани за претрагу и операцију читања у односу на перформансе писања. Подаци који изистуку потребу за честим изменама више одговарају релационим базама података. *LDAP* је намењен подацима који су централизовани и нису често подложни изменама, стога се може категоризовати као сервис - пиши једном, читај више пута [2].

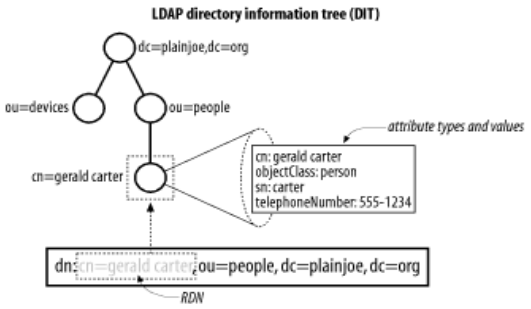
Модели *LDAP* сервера:

1) Информациони модел (*Information Model*)

дефинише структуру и тип података који могу бити сачувани и приказани у информационом стаблу *LDAP* сервера. Ћелија система је инстанца ентитета из реалног света (сервер, уређај, корисник), која је описана кроз атрибуте и приказана као елемент *DIT* стабла [2].

2) Модел именовања (*Naming Model*)

дефинише начин креирања идентификатора за ћелију стабла, додељујући јој *DN* (*Domain name*), као и организацију у оквиру *DIT* хијерархије [2].



Слика 1. Модел именовања [4]

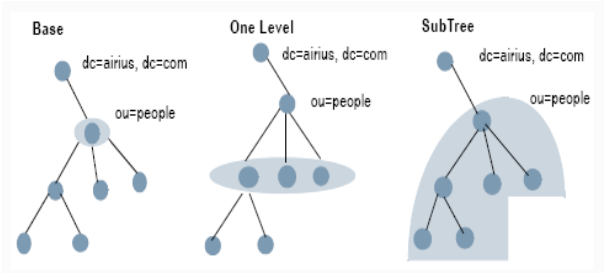
Односно описује како су подаци у стаблу јединствено идентификовани у односу на остале *entry*-је истог родитеља. *RDN* (*Relative distinguished name*) је атибут за идентификацију *entry*-ја, који улази у састав *DN*-а (слика 1).

3) Функционални модел (*Functional Mode*)

дефинише функционалности сервера подељене у три категорије: упити, ажурирање, и аутентификација. Оптимална претрага једна је од главних предности

НАПОМЕНА:
Овај рад проистекао је из мастер рада чији ментор је био др Горан Сладић, ред. проф.

LDAP сервера, а предуслови су дефинисање основног чвора (*Base DN*) и опсега претраге (слика 2). Опсег претраге специфицира ниво претраге у упиту и тиме ограничава област претраживања [4].



Слика 2. Функционални модел [4]

4) Безбедносни модел (*Security Model*)

одређује степен приступа за аутентификованог клијента на основу политике права приступа сервера.

3. КОНТРОЛА ПРИСТУПА И *APACHE FORTRESS*

Контрола приступа ограничава управљање ресурсима на основу исхода аутентификације и ауторизације корисника. Ресурс је објекат који садржи и пружа информације. Заштићен ресурс у зависности од система над којим је контрола приступа имплементирана може бити документ, директоријум оперативног система, табела или пројекција у оквиру система за управљање базама података [5].

Постоје три основна типа контроле приступа:

1) Дискрециона контрола приступа

(*Discretionary Access Control - DAC*) је систем који додељује права приступа на основу правила која су специфицирана од стране корисника. Принцип на ком се овај тип заснива је да субјекат одређује ко може приступити његовом објекту [6].

2) Контрола приступа базирана на улогама

(*Role Based Access Control - RBAC*) је недискрециони тип, где право није одређено од стране субјекта, већ од стране администратора система.

3) Обавезујући тип контроле приступа

(*Mandatory Access Control - MAC*) ослања се на хијерархијску структуру добављања ресурса. Приступ ресурсима контролисан је на основу подешавања дефинисаних од стране администратора система. Додељујући степен класификације и тип категорије, *MAC* одређује услове приступа ресурсима [6].

3.1. *Role Based Access Control (RBAC)*

RBAC модел садржи четири компоненте:

1) Основни модел (*Core RBAC*) дефинише

базични сет *RBAC* елемената: корисници, улоге, пермисије, операције, објекти, као и релације између њих [5].

2) Хијерархијски модел (*Hierarchical RBAC*)

односи се на структурирање улога чија је намена рефлектовање одговорности и области управљања у оквиру организације [5].

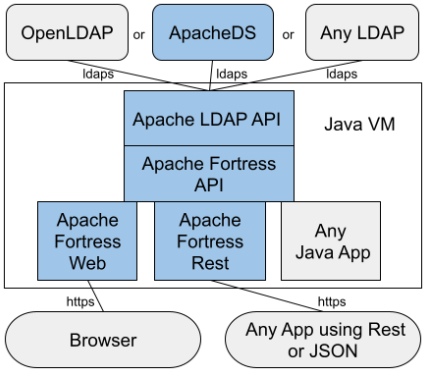
3) Статичко раздвајање улога (*Static Separation of Duty Relations*) дефинише међусобно дисјунктне корисничке задатке у односу на скуп улога [5].

4) Динамичко раздвајање улога (*Dynamic Separation of Duty Relations*) дефинише својства која ограничавају доступност пермисија кориснику, додељујући ограничења улогама које могу бити активиране кроз корисничке сесије. Обезбедити да корисник има додељену пермисију само у оквиру времена када му је потребна је суштина динамичког додељивања улога [7].

3.2. *Apache Fortress*

Apache Fortress је стандардизован систем за ауторизацију, који пружа контролу приступа на основу корисничких улога, делегирану администрацију укључујући политику корисничких лозинки коришћењем *LDAP* сервера. *Fortress* користи директоријумски сервер за чување информација о корисницима, њиховим улогама, пермисијама и слично. *ApacheDS и OpenLDAP* су нативно подржани сервери, међутим могуће је користити било који *LDAPv3* компатабилан систем [8].

На слици 3. приказана је архитектура *Apache Fortress* система. Комуникација почиње слањем захтева, који је иницираан од стране претраживача или било које друге клијентске апликације. У позадини *Apache Fortress* користи *Java API* намењен за комуникацију са неким од *LDAP* сервера.



Слика 3. Архитектура *Apache Fortress* система [9]

Apache Fortress садржи четири компоненте:

1) *Apache Fortress Core* [10] је централна

компонента коју користе остали делови система *Apache Fortress*-а омогућавајући комуникацију са *Apache Directory Server*-ом преко *SSL* протокола [11].

2) *Apache Fortress Realm* компонента уграђивањем

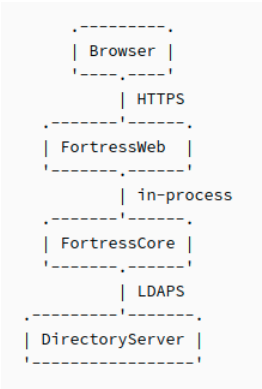
у *Apache Tomcat* [12] обезбеђује безбедносну подршку коришћењем *Java EE*, која мапира кориснике и њихове улоге назад ка повезаном *LDAP* серверу.

3) *Apache Fortress Rest* компонента представља

HTTP сервисни интерфејс који обмотава *API*-је приликом интеракције. Ток развоја приказан је на слици 4. где су приказана три нивоа [13]:

- Клијент је било који *HTTP* интерфејс који подржава поруке формата намењеног за *Apache Fortress.*

- Сервлетски контејнер намењен за дистрибуцију веб апликација.
- Директоријумски сервер складишти полису и извршава *LDAPv3* протоколе.



Слика 3. Приказ нивоа развоја [13]

Предуслови да би *Apache Fortress Rest* био спреман за аутентификацију су подешени следећи параметри:

- Координате *LDAP hostname*-а
- Тип *LDAP* сервера
- Креденцијали административног налога
- Дефинисање путање конфигурационог чвора у стаблу
- Информације о информационом стаблу *LDAP* сервера

4) *Apache Fortress Web* обухвата *HTML* странице и радни оквир.

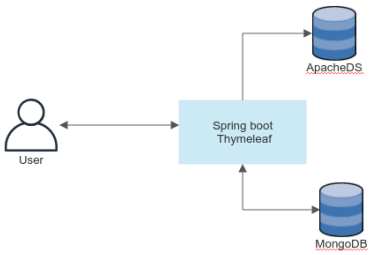
Ревизија акција (*Auditing*) омогућава бележење интеракција са *LDAP* сервером. Акције додавања, ажурирања, брисања ентитета спадају у интеракције које се бележе и размена података о истим је могућа преко *API*-ја. Претрага акције ауторизације и аутентификације је пример намене система за ревизију акција [2].

Вишебројни тенанти (*Multitenancy*) односи се на софтвер чија је једна инстанца присутна на серверу, а опслужује више тенанта. Софтвер је дизајниран тако да сваком тенанту пружа само одређене податке у оквиру инстанце укључујући његове податке, конфигурацију, функционалности и подешавања тенанта [15]. Дизајн архитектуре за потребе различитих тенанта подразумева још један ниво хијерархије у оквиру *LDAP* сервера.

4. МОДЕЛИ СЕРВЕРА

Пројекат електронски дневник је реализован као мултитенант апликација, чија је намена евиденција активности ђака у образовном систему. Препознато је пет улога у систему, где је за сваког корисника улоге дефинисан лист стабла са основним сетом података у оквиру *Apache Directory Servera*.

Приказ архитектуре система електронског дневника налази се на слици 5. Корисник уносом креденцијала иницира пријаву на систем, шаље захтев ка бекенд апликацији која комуницира са *ApacheDS*-ом и врши верификацију унетих података. Након успешног приступа систему, ширем скупу података могуће је приступити преко конекције ка нерелационој бази података, *MongoDB*.



Слика 4. Ток података

4.1. Функционалности система

RBAC је механизам за ограничавање скупа функционалности доступних кориснику на основу додељене улоге. Улоге су додељене у централном чворишту за аутентификацију и ауторизацију, *ApacheDS*.

Администратор система (*ADMIN*) има улогу која управља ентитетом школске године и њеним статусом. Главна функционалност корисника са улогом *ADMIN* је управљање тенант администраторима (*TENANT_ADMIN*), односно директорима школа. Улога администратора тенанта је креирање наставника у оквиру школе, у којој је администратор, креирање разреда и додељивање резредних старешина. Наставник може имати школски разред ком је ментор, док разред има тачно једног разредног старешину.

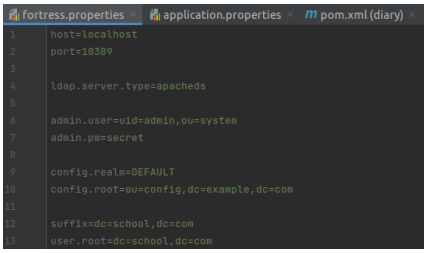
Централне активности корисника са улогом наставника (*TEACHER*) су одржавање часова, унос оцена и изостанака. Бирајући разред, наставник започиње час који је повезан са ентитетом оцене и изостанка. Сваки школски час чува временску одредницу почетка и краја часа, податак о наставнику који је час одржао, као и предмет који је предаван. Оцене и изостанци садрже тачно један час и ђака, чиме је одређено када и за кога је запис унет. Корисници са улогама родитеља (*PARENT*) и ђака (*STUDENT*) имају могућност прегледа изостанака и оцена. Ђаци чувају податак о тренутном школском разреду који похађају, као и листу претходних. Листа претходних разреда служи за евиденцију успеха ђака током школовања. Ентитет ђака повезан је са ентитетом родитеља, имајући тачно једног док родитељ има листу идентификатора који одређују децу.

5. ИМПЛЕМЕНТАЦИЈА

Конфигурација *ApacheDS*-а извршена је кроз окружење *Apache Directory Studio*. *Apache Directory Studio* је алат за рад са директоријумима, првенствено са *ApacheDS*-ом, а неке од функционалности су приступ и претраживање *LDAP* информационог стабла, уграђивање *LDIF* шеме преко *Schema Editor*-а, конфигурација сервера и слично. Стандардне шеме које су иницијално садржане на самом серверу не подржавају атрибуте који су неопходни за рад са *Apache Fortress*-ом. Након успешног увоза *Apache Fortress LDIF* текстуалног документа, у подстаблу конфигурационог чвора (*ou=schema*) бива приказан садржај претходно одабране шеме.

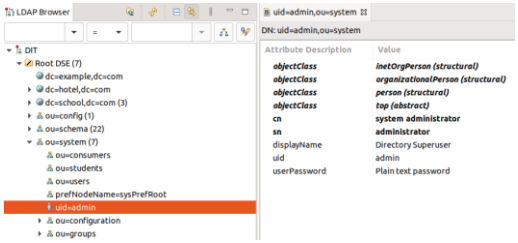
Параметри неопходни за комуникацију са *LDAP* сервером постављени су у *fortress.properties*

документу (слика 6). Координате *LDAP hostname*-а постављене су у прве две линије документа, што указује на остваривање конекције са *LDAP* сервером преко *localhost*-а кроз порт 10389.



Слика 6. Конекција са *ApacheDS*

Следи подешавање типа сервера, затим креденцијала административног налога, односно где је постављена његова позиција из структуре директоријумског информационог стабла (слика 7).

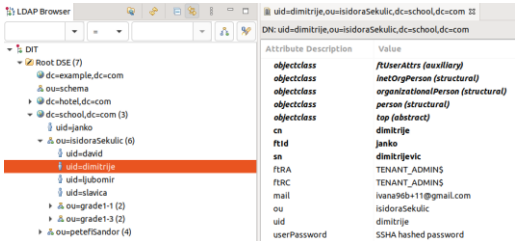


Слика 5. Администратор сервера

5.1. Корисници система

Апликација електронски дневник препознаје ентитет школе, разреда и корисника са одређеном улогом. Постизање система са више тенанта имплементирано је креирајући више чворова са објектом класе *organizationalUnit*.

Администратор тенанта, односно директор школе приказан је на слици 8. где видимо доменско име *entry*-ја и њихове атрибуте. *FtUserAttrs* је класа објекта *LDIF* шеме *Apache Fortress*-а, која садржи атрибуте о улогама корисника у систему, *ftRA* и *ftRC*.



Слика 6. - Тенант администратор

Процес пријаве започет је уношењем корисничког имена и лозинке, затим следи апликативно одређивање *dn*-а. Тек након одређивања *dn*-а као критеријума претраге, остварује се конекција са *LDAP* сервером. Након успешно добављеног корисника, следи аутентификација где истоимена метода проверава једнакост лозинке која је унета и вредност атрибута *userPassword entry*-ја са *LDAP* сервера.

Следи постављање аутентификованог корисника у сесију и даље се управо из ње вуку сви неопходни подаци приликом ауторизације. Ауторизација приступа корисника *endpoint*-има извршена је анотацијом *PreAuthorize* којом је означено са којом улогом корисник има право приступа одређеној функционалности.

6. ЗАКЉУЧАК

Кроз пројекат за евиденцију активности у школском систему, приказан је развој апликације раздвајајући осетљиве податке од функционалних информација. Осетљиви подаци, централизовани су у чвор *ApacheDS* сервера чији су бенефити приказани кроз функционалности и перформансе. Простор за проширење апликације пружен је кроз моделовање динамички раздвојених улога, чиме би се обезбедио највиши степен заштите података.

7. ЛИТЕРАТУРА

[1] https://directory.apache.org/apacheds/basic-ug/1.2-some-background.html (приступљено у јулу 2022.)

[2] https://www.zytrax.com/books/ldap/ (приступљено у јулу 2022.)

[3] https://www.lepide.com/blog/what-is-active-directory-and-how-does-it-work (приступљено у јулу 2022.)

[4] https://docs.informatica.com/data-integration/powerexchange-adapters-for-powercenter/10-2/powerexchange-for-ldap-user-guide-for-powercenter/understanding-powerexchange-for-ldap/ldap-models.html (приступљено у јулу 2022.)

[5] Information Technology Industry Council, American National Standard for Information Technology – Role Based Access Control. 2004.

[6] https://westoahu.hawaii.edu/cyber/best-practices/best-practices-weekly-summaries/access-control/ (приступљено у августу 2022.)

[7] https://dl.acm.org/doi/epdf/10.1145/300830.300834 (приступљено у августу 2022.)

[8] https://directory.apache.org/fortress/overview.html (приступљено у јуну 2022.)

[9] https://github.com/apache/directory-fortress-enmasse (приступљено у јуну 2022.)

[10] https://directory.apache.org/apacheds/basic-ug/1.4.3-adding-partition.html#what-are-partitions (приступљено у јулу 2022.)

[11] https://www.ibm.com/docs/en/ibm-http-server/9.0.5?topic=communications-secure-sockets-layer-ssl-protocol (приступљено у августу 2022.)

[12] https://tomcat.apache.org/ (приступљено у августу 2022.)

[13] https://apache.googlesource.com/directory-fortress-commander/+/HEAD/README-SECURITY-MODEL.md (приступљено у августу 2022.)

Кратка биографија:

Ивана Благојевић рођена је у Новом Саду 1996. године. Основне академске студије завршила је 2019. године на Факултету техничких наука. Мастер рад из области Електротехнике и рачунарства - Рачунарство и аутоматика је одбранила 2022. године.
контакт: ivana96b@gmail.com