Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007:004.056
DOI: https://doi.org/10.24867/20BE21Rankovic

КОНТРОЛА ПРИСТУПА ПЛАТФОРМЕ ЗА ИВИЧНО РАЧУНАРСТВО

ACCESS CONTROL OF AN EDGE COMPUTING PLATFORM

Тамара Ранковић, *Факултет техничких наука, Нови Сад*

Област – РАЧУНАРСТВО И АУТОМАТИКА

Кратак садржај – *Ивично рачунарство јавља се као парадигма која треба да превазиђе недостатке централизоване организације рачунарских ресурса традиционалног рачунарства у облаку. Приликом развоја сервиса ивичног рачунарства, потребно је решити и проблем контроле приступа. Кроз рад је описана политика и механизам контроле приступа којима се проширује постојећа платформа која нуди услуге ивичног рачунарства. Политика контроле приступа оптимизована је за рад са хијерархијском организацијом ресурса. На основу дефинисане политике, имплементиран је систем за централизовану ауторизацију.*

Кључне речи: *контрола приступа, ауторизација, ивично рачунарство*

Abstract – *Edge computing emerges as a paradigm aiming to overcome problems caused by centralized organization of the traditional cloud. One aspect to take into consideration when developing edge computing services is access control. Main purpose of this paper is to define and implement access control policy and mechanism for an existing edge computing platform. Described access control policy is optimized for hierarchical organization of resources. A centralized authorization system is developed, based on the access control policy.*

Keywords: *access control, authorization, edge computing*

1. УВОД

Идеја иза парадигме ивичног рачунарства (енгл. *edge computing*) јесте децентрализација облака кроз измештање рачунарских ресурса из огромних центара података (енгл. *data centers*), а све са циљем физичког приближавања тих ресурса крајњим корисницима [1]. Један од проблема који треба решити приликом развоја нових платформи које ће нудити услуге ивичног рачунарства је контрола приступа. Контрола приступа представља битан алат који очувава приватност корисника и осигурава безбедност читавог система. Један од подпроблема контроле приступа у контексту ивичног рачунарства је ауторизација захтева за приступ подацима и сервисима.

НАПОМЕНА:

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

Платформа Constellations (c12s) [2] имплементира протоколе за динамичко формирање дистрибуираних микрооблака (енгл. *micro-clouds*) и нуди услуге ивичног рачунарства као било који други сервис рачунарства у облаку. Кроз овај рад потребно је описати политику контроле приступа и имплементацију механизма контроле приступа којом ће платформа бити проширена. За развијено решење потребно је приказати његову употребљивост за различите сценарије коришћења.

2. ПРЕГЛЕД ОБЛАСТИ

Један од важних задатака које информациони систем треба да подржи је заштита ресурса и података тако да се одржи њихова поверљивост, интегритет и доступност. Сваки приступ систему мора се контролисати и осигурати да се само ауторизованим лицима обезбеди приступ. Тај процес назива се контрола приступа.

Скуп правила која одређују које активности су дозвољене у систему над којим ресурсима у контексту контроле приступа називају се безбедносне политике [3]. Политике контроле приступа могу се поделити у три групе [3]: Discretionary Access Control (DAC), Mandatory Access Control (MAC) и Role-Based Access Control (RBAC).

Једна или више безбедносних политика могу се описати помоћу модела контроле приступа који дефинишу формализме за спецификацију и имплементацију безбедносних политика и омогућавају анализу безбедносних политика [3]. Имплементација безбедносних политика врши се помоћу метода или алата који се називају безбедносни механизми [3].

Поред основних модела наведених група политика, постоје и бројна проширења како би се надоместили неки од недостатака које оригинални модели поседују. Најмлађа група међу значајним групама политика јесте Attribute-Based Access Control (ABAC) чији циљ је постизање фине контроле приступа коју RBAC није могао да обезбеди.

2.1. Системи за ауторизацију

Open Policy Agent (OPA) [4] је софтверска компонента која складишти и евалуира политике како би донела одлуку о томе да ли је одређена политика важећа или не, узимајући у обзир тренутно стање система. Једна од врста политика које OPA може да складишти су политике контроле приступа. OPA омогућава раздвајање (енгл. *decoupling*) политика контроле приступа од пословне логике система. Неке од предности оваквог приступа су могућност дељења

политика између апликација, централизовано управљање политикама, могућност тестирања политика у изолованом окружењу и њихово олакшано разумевање. Подаци који су неопходни за евалуацију политика могу се проследити као део тела захтева или унапред реплицирати из оригиналног извора. Мане које OPA поседује везане су за то што у себе не укључује слој за трајно складиштење података. У случају да се подаци унапред учитавају, сви се морају налазити у меморији и не постоји гаранција њихове трајности. Са друге стране, уколико се подаци сваки пут шаљу у оквиру захтева, може доћи до знатних кашњења узрокованих преносом велике количине података преко мреже.

Zanzibar [5] је глобални систем за ауторизацију који омогућава складиштење и евалуацију листи контроле приступа. Подаци се складиште као торке које се састоје из субјекта, релације и објекта. Операцијама алгебре скупова дефинишу се потенцијално сложене политике које описују односе између субјеката и објеката. Апликативни програмски интерфејс (енгл. *Application Programming Interface, API*) сервиса нуди операције за читање, писање, евалуацију листи и асинхроно праћење измена над њима. Субјекти могу представљати кориснике или групе корисника. Група се састоји из скупа корисника и/или других група. Овакав модел доводи до потенцијално дубоког угњеждавања корисника и њихове припадности групама, која се мора утврдити приликом евалуације политика. Проблем је решен формирањем индекса из ког се директно може проверити којим групама корисник припада. Захтев за евалуацију одговара на питање да ли дати корисник поседује неку релацију над одабраним објектом.

3. ОПИС ПЛАТФОРМЕ

Основни задатак платформе је динамичко формирање микрооблака, организованих у хијерархијску структуру. На најнижем нивоу налазе се груписани чворови – кластери (енгл. *cluster*) који су сачињени из скупа чворова [2]. Један или више кластера формира регион, а један или више региона организовани су у топологију [2]. Микрооблаци се формирају према захтеву корисника и од тренутно слободних чворова који задовољавају задате услове. Ресурси над којима систем подржава операције су топологије, региони, кластери, простори имена, конфигурације, тајне и логови.

Политика контроле приступа платформе треба да се ослони на постојеће моделе ради лакшег прихватања од стране корисника. Како различити корисници могу имати потребу за различитим нивоима грануларности приликом управљања привилегијама, политика треба да подржи што је могуће финију контролу приступа, али и да подржи једноставне сценарије употребе. Многи ресурси у систему поседују хијерархијску организацију, као што је сама организација микрооблака. Из тог разлога одабрани модел треба да подржи наслеђивања привилегија, како не би било потребно експлицитно наводити дозволе за ниже нивое хијерархије, ако их корисник већ поседује на неком од виших нивоа.

Поред наслеђивања дозвола над ресурсима над којима се врше акције, пожељно је сличан механизам применити и приликом организације корисника који представљају субјекте у систему. Управљање дозволама сваког појединачног корисника у великим организацијама захтеван је посао и подложан грешкама. Стога, механизам груписања корисника може бити од велике помоћи. Корисници своје дозволе затим могу наследити на основу група којима припадају.

Провера права приступа је саставни део сваког захтева ка систему и стога је потребно имплементирати је тако да буде што ефикаснија, односно да за врло мало време може да пружи одговор, како не би негативно утицала на перформансе система. Такође, модул задужен за проверу дозвола треба да буде високо доступан јер представља компоненту од које зависе све остале.

4. ПОЛИТИКА КОНТРОЛЕ ПРИСТУПА

Кроз ову секцију биће описани елементи формиране политике контроле приступа и операције које се могу обављати над њима.

4.1 Елементи

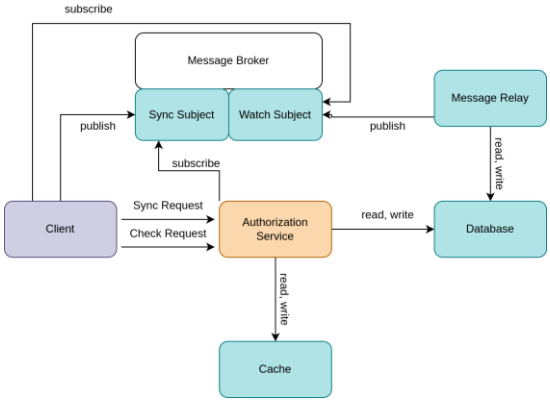
Ресурсима су представљени сви пасивни и активни ентитети у систему. Активни ентитети су они који могу да врше операције. Операције се врше над пасивним ентитетима. Свака инстанца ресурса припада одређеној врсти ресурса.

Атрибути се могу поделити на оне који описују ресурсе и на оне који представљају стање окружења у тренутку провере дозволе. Обе врсте поседују назив и вредност произвољног типа. Улога атрибута је да ближе опишу ресурс и омогуће финију контролу приступа. Атрибути околине слични су атрибутима ресурса по томе што пружају додатне информације на основу којих се могу доносити одлуке боље поткрепљене контекстом. Разлика између две врсте је та да су атрибути ресурса при провери дозволе унапред познати, док се вредности атрибута околине региструју у тренутку доношења одлуке.

Дозволе контролишу које акције се могу вршити над којим ресурсима и под којим условима. Свака дозвола повезује два ресурса, где је један активан, а други пасиван ентитет. Дозвола поседује назив који треба да представља операцију која се врши. Поред назива поседује и врсту, која дефинише да ли се одређена активност дозвољава или забрањује. Дозволе могу важити независно од контекста, а могу им се доделити и логички услови на основу којих се узима у обзир да ли дозвола важи или не. Услов представља функцију чије су улазне вредности атрибути субјекта, објекта и околине, а повратна вредност је Буловог типа и дефинише да ли су услови за важење дозволе испуњени или не.

Између два ресурса може се формирати ***хијерархијска веза*** тако да један ресурс представља родитеља, а други дете. Сматра се да је дете егзистенцијално зависно од родитеља. Субјекат, поред директно формираних дозвола над неким ресурсом, наслеђује и све дозволе својих родитеља. Дозвола која је директно

везана за неки објекат пропагира се и до његове деце. Ако субјект има одређену дозволу над родитељом, има је и над дететом.



Слика 1. *Токови комуникације система за ауторизацију*

4.2. Операције

Подржане операције укључују додавање хијерархијске везе међу ресурсима и њено уклањање, додавање и брисање атрибута ресурса, креирање и уклањање дозволе и проверу дозволе.

У ситуацији када се брише хијерархијска веза, поред њеног брисања, потребно је уклонити и све ресурсе чија је егзистенцијална зависност нарушена тим брисањем. Један од начина за проналазак свих ресурса које треба уклонити је рекурзивни пролазак кроз стабло у чијем корену се налази дете обрисане хијерархијске везе. Почевши од њега, треба проверити да ли ресурс има барем једног родитеља и ако нема обрисати га и поступак поновити за свако његово дете. Уколико има једног или више родитеља, тренутну процедуру треба терминирати без брисања ресурса.

Провера дозволе треба да одговори на питање да ли дати субјект над датим објектом може да изврши наведену акцију у тренутним условима околине. Поступак доношења одлуке укључује следеће кораке:

1. *Проналазак дозвола*: Било који субјекат над било којим објектом може имати више дозвола истог назива. Скуп дозвола истог назива које субјекат има над објектом представља унију скупова дозвола које су директно том субјекту додељене над тим објектом, дозвола које су додељене том субјекту над неким претком објекта, дозвола које су претку субјекта додељене над објектом и оних које су му додељене над претком објекта.

2. *Додела приоритета*: Из скупа прикупљених дозвола потребно је одредити које дозволе имају већу, а које мању тежину. Дозволе које су наслеђене имају мањи приоритет од оних које су директно додељене ресурсу.

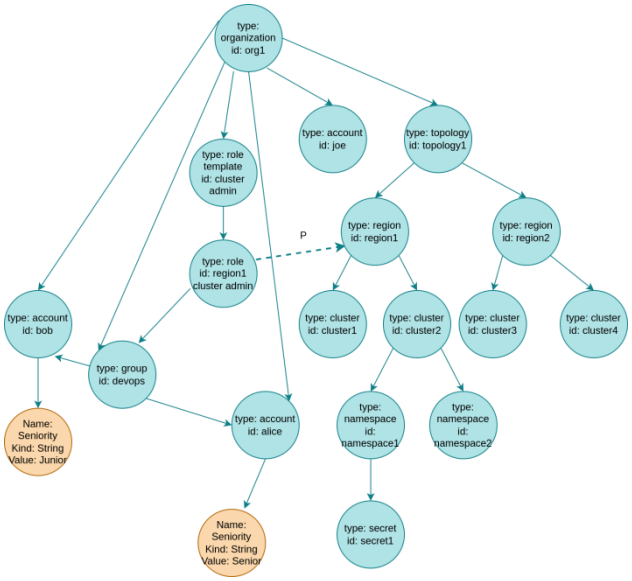
3. Одређивање важећих дозвола и доношење одлуке: Корак доношења одлуке спроводи се евалуацијом дозвола редом по њиховим приоритетима док се не дође до одговора или не истроше све дозволе преузете у првом кораку. Дозволе које уз себе имају придружен неки услов узимају се у обзир приликом доношења одлуке само уколико је тај услов задовољен. У супротном, или се прелази на дозволе нижег приоритета које имају испуњен услов или га уопште не поседују или се приступ забрањује уколико не постоји више дозвола које се могу евалуирати.

5. СИСТЕМ ЗА АУТОРИЗАЦИЈУ

Сервис је имплементиран у програмском језику Go. Синхрони API имплементиран је употребом gRPC радног оквира и Protocol Buffers (Protobuf) механизма за серијализацију структурираних података. Асинхрони API базиран је на механизму објављивања и претплате на поруке (енгл. *publish-subscribe*). Као брокер за размену порука користи се NATS . Сви подаци трајно се складиште у Neo4j граф бази, док Redis кешира податке у меморији као парове кључ-вредност.

Прву групу операције чине операције за ажурирање, односно синхронизацију података. Наредну групу чине операције за проверу дозвола. Клијент може да провери да ли одређени субјекат има дозволу задатог назива над датим објектом. Сервис се приликом провере ослања на податке којима је кеш попуњен приликом претходних провера. Уколико подаци ту нису доступни, добавља их из базе која их трајно перзистира.

Приликом синхронизације података клијенту је често неопходно да очува конзистентност. У таквим ситуацијама може се ослонити на трећи тип операција, а то су операције за праћење промена података. Сервис на одговарајућу тему (енгл. *subject*) емитује догађаје које у себи садрже информације о свакој измени у подацима. Како би ово урадио, користи механизам трансакционог слања порука. Клијент ће пријавом на исту ту тему почети да прима поруке и применом механизама као што је сага успешно моћи извршити дистрибуирану трансакцију.



Слика 2. *Пример хијерархије ресурса*

На слици 1 приказана је архитектура система са током захтева који се обављају. Захтеви за синхронизацију стижу директно преко gRPC API-ја или преко теме за захтеве за синхронизацију, којом управља брокер за размену порука. Сваки захтев садржи јединствени идентификатор који се затим шаље и као део

одговора, како би клијент знао који одговори су намењени њему. Преносилац порука (енгл. *message relay*) региструје промене које су се десиле и објављује их на тему намењену за слање одговора. Захтеви за проверу дозвола су такође gRPC захтеви који стижу директно сервису за ауторизацију. Он затим комуницира са кеш компонентом и са базом података, а затим одговор враћа синхроно.

5.1. Складиштење података

Граф базе представљају подврсту база података које се могу описати као нерелационе (енгл. *Not only SQL, NoSQL*) базе оптимизоване за складиштење, претрагу и ажурирање података који имају структуру графа. Neo4j је представник граф база, који користи модел графа за складиштење на најнижем нивоу. Neo4j не захтева употребу шеме података што омогућава једноставну еволуцију модела података кроз време. Упитни језик који Neo4j користи назива се Cypher. Neo4j информације организује у чворове, везе и својства.

Одабир граф базе за складиштење ауторизационих података потиче од захтева за ефикасну проверу дозвола. Формирање скупа дозвола које субјект има над објектом укључује пролазак кроз потенцијално дубоку хијерархију ресурса. Уколико се овакви упити формирају над релационом базом података, јавља се потреба за великим бројем операција спајања табела (енгл. *joins*), које су временски изузетно скупе.

6. ИНТЕГРАЦИЈА

Пример хијерархије ресурса на нивоу једне организације приказан је на слици 2. Плави чворови представљају ресурсе, наранџасти атрибуте, док су пуним линијама приказане хијерархијске везе, а испрекиданим линијама дозволе.

```
{
  "permissionName": "namespace.create",
  "principal": {
    "id": "alice",
    "kind": "account"
  },
  "resource": {
    "id": "cluster1",
    "kind": "cluster"
  },
  "envAttributes": [
    {
      "name": "ipaddress",
      "kind": "string",
      "value": ... /* byte array representing the serialized string attribute
with value 1.2.3.4 */
    }
  ]
}
```

Листинг 1. *Пример захтева за проверу дозволе*

Улога администратора кластера додељена над објектом региона гарантује потребне дозволе над свим кластерима у оквиру тог региона. Међу дозволама из скупа P налази се дозвола која омогућава креирање простора имена. Услов дозволе захтева да корисник који обавља акцију има сениоритет *Senior* и да је IP адреса са које се захтев шаље 1.2.3.4. Пример тела захтева за проверу ове дозволе који би био успешан приказан је у листингу 1.

7. ЗАКЉУЧАК

Кроз овај рад приказано је једно од могућих решења за спровођење контроле приступа у оквиру платформе ивичног рачунарства. Дефинисана је политика контроле приступа и имплементирана је кроз систем за ауторизацију који се интегрише са остатком система.

Правци даљег рада укључују формализацију дефинисане политике кроз модел контроле приступа и њено проширење. Други ток развоја односи се на унапређење комуникације са системом за ауторизацију развојем клијентске библиотеке. Поред тога, потребно је решити проблем недостатка ограничења над типовима ресурса и везама међу њима. Може се дефинисати механизам за формирање скупа ограничења над којим ће се валидирати сваки пристигли захтев.

8. ЛИТЕРАТУРА

[1] Ana Juan Ferrer, Joan Manuel Marquès, and Josep Jorba. *Towards the decentralised cloud: Survey on approaches and challenges for mobile, ad hoc, and edge computing*. ACM Computing Surveys (CSUR), 51(6):1–36, 2019.

[2] Miloš Simić. *Dynamic formation of the distributed micro clouds*. PhD dissertation, University of Novi Sad, 2022.

[3] Messaoud Benantar. *Access control systems: security, identity management and trust models*. Springer Science & Business Media, 2005.

[4] https://www.openpolicyagent.org/docs/latest/ (Приступљено у августу 2022)

[5] Ruoming Pang, Ramon Caceres, Mike Burrows, Zhifeng Chen, Pratik Dave, Nathan Germer, Alexander Golynski, Kevin Graney, Nina Kang, Lea Kissner, Jeffrey L. Korn, Abhishek Parmar, Christina D. Richards, and Mengzhi Wang. *Zanzibar: Google's consistent, global authorization system*. In 2019 USENIX Annual Technical Conference, Renton, WA, 2019.

Кратка биографија:

Тамара Ранковић рођена је 29.11.1998. године у Шапцу. Школске 2017/2018. године уписује основне академске студије на Факултету техничких наука у Новом Саду, смер Рачунарство и аутоматика. Школске 2021/2022. године уписује мастер академске студије на Факултету техничких наука у Новом Саду, смер Рачунарство и аутоматика. Исте школске године полаже све испите и студије завршава у року.