Kauno technologijos universitetas

Informatikos fakultetas

Prieigos kontrolės metodas valdiklių valdymo platformoms

Baigiamasis magistro projektas

Klaidas Klimakas

Projekto autorius

Doc. Tomas Adomkus

Vadovas

Kaunas, 2023

Kauno technologijos universitetas

Informatikos fakultetas

Prieigos kontrolės metodas valdiklių valdymo platformoms

Baigiamasis magistro projektas

Informacijos ir informacinių technologijų sauga (6211BX008)

Klaidas Klimakas

Projekto autorius

Doc. Tomas Adomkus

Vadovas

Doc. Audronė Janavičiūtė

Recenzentė

Kaunas, 2023

Kauno technologijos universitetas

Informatikos fakultetas

Klaidas Klimakas

Prieigos kontrolės metodas valdiklių valdymo platformoms

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;

2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąraše;

3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);

4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Klaidas Klimakas

Patvirtinta elektroniniu būdu

Klimakas Klaidas. Prieigos kontrolės metodas valdiklių valdymo platformoms. Magistro baigiamasis projektas / vadovas doc. Tomas Adomkus; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos inžinerija, informatikos mokslai.

Reikšminiai žodžiai: prieigos kontrolė, autorizacija, daiktų internetas, valdiklių valdymo platformos, sprendimų priėmimo komponentas, rizika grįstas, kontekstas, aplinkos požymiai.

Kaunas, 2023. 65 p.

Santrauka

Valdiklių valdymo platformos – tai sparčiai augančios daiktų interneto infrastruktūros dalis. Platformose vykdomas fizinių įrenginių valdymas bei duomenų surinkimas, apdorojimas ir integravimas, todėl jos tampa kibernetiniams nusikaltėliams patraukliu taikiniu. Silpna prieigos kontrolė yra viena iš esminių kibernetinio saugumo grėsmių šioje srityje. Taigi, magistro baigiamajame darbe siekiama sudaryti saugią valdiklių nuotolinio valdymo platformą panaudojant tam pritaikytą prieigos kontrolės metodą.

Analizės dalyje aptariama daiktų interneto ekosistema ir platformų veikimo principai. Nagrinėjamos šioms platformoms būdingos kibernetinio saugumo grėsmės, prieigos kontrolei naudojamos technologijos ir metodai. Pastebima, jog daiktų interneto sprendimai išsiskiria dinamika, ribotais resursais, masiškumu bei jungiamumu, todėl naudojami tradiciniai statiniai prieigos kontrolės metodai nėra tinkami. Teigiama, jog norint užtikrinti mažiausių teisių principą, prieigos kontrolės metodo sprendimų priėmimo komponentas turi pasižymėti detalumu, lankstumu bei paprastumu. Projektavimo dalyje sudaromas metodas, kurio sprendimų priėmimo procese naudojamos subjektų rolės bei rizikomis grįstos prieigos politikos taisyklės. Infrastruktūros situacijos kritiškumui nustatyti bei rizikos balui suskaičiuoti, vertinamas objekto jautrumas, veiksmo poveikis bei kintantys aplinkos konteksto požymiai. Realizacijos dalyje pateikiama metodo įgyvendinimui pasirinkta programinė ir techninė įranga, būdinga tradicinei daiktų interneto infrastruktūrai. Pristatomas realizuojamas standartinės valdiklių valdymo platformos prototipas su valdikliu bei jo naudojamais įvesties ir išvesties įrenginiais.

Tyrimo dalyje siekiama įvertinti pasiūlyto prieigos kontrolės metodo veiksmingumą. Konfigūruojama realizuota platforma pritaikant ją išmaniųjų namų sprendimams. Vykdomi metodo administracinių kaštų, greitaveikos ir saugumo eksperimentai pagal paruoštus scenarijus. Baigiamajame darbe aptariami gauti rezultatai ir pateikiamos išvados.

Klimakas Klaidas. Access Control Method for Controller Management Platforms. Master's Final Degree Project / supervisor assoc. prof. Tomas Adomkus; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Informatics Engineering, Computing.

Keywords: access control, authorization, Internet of Things, controller management platforms, policy decision point, risk-based, context, environment attributes.

Kaunas, 2023. 65.

Summary

Controller management platforms – a part of a fast-growing infrastructure of the IoT (Internet of Things). The management of physical devices as well as data collection, processing and integration is carried out in these platforms, that is why they are becoming an attractive target for cybercriminals. Weak access control is one of the fundamental cybersecurity threats in the field. Thus, the aim of the master's thesis is to create a secure controller remote management platform using an access control method adapted for this purpose.

The analysis part discusses the ecosystem of the IoT and the operating principles of the platforms. Cyber security threats specific to these platforms, technologies and methods used for access control are examined. It is noticed that IoT solutions are distinguished by their dynamics, limited resources, massiveness, and connectivity, so the traditional static access control methods used are not suitable. It is claimed that to ensure the principle of least privilege, the policy decision point of an access control method should be characterized as granular, flexible, and simple. In a project part, the method is formed, whose decision-making process employs subject roles and risk-based access control policies. To determine the criticality of the infrastructure situation and calculate the risk score, the sensitivity of the object, the effect of the action and the altering environmental context are assessed. The realization part describes the software and hardware selected for the implementation of the method, typical of the IoT infrastructure. The implementation of a standard controller management platform with a controller and its input and output devices is presented.

The research aims to evaluate the effectiveness of the proposed access control method. Thus, configuration of the implemented platform is carried out by adapting it to smart home applications. Experiments on administrative costs, speed and security are carried out in accordance with prepared scenarios. Finally, the thesis discusses the obtained results and presents the conclusions.

Turinys

Lentelių sąrašas

Paveikslų sąrašas

Įvadas

Projekto naujumas ir aktualumas

Daiktų internetas (angl. *Internet of Things*) – tai sparčiai auganti IT sritis, kurios sprendimuose sujungiami tiek fiziniai įrenginiai, tiek informacinės sistemos. Jutikliai stebi ir fiksuoja aplinkos parametrus, vykdikliai fizinėje aplinkoje atlieka nurodytus veiksmus, o valdikliai užtikrina šių įrenginių valdymą. Valdikliai taip pat užtikrina duomenų apdorojimą ir komunikaciją su platformomis. Taikomajame sluoksnyje veikiančios platformos atlieka tarpininko vaidmenį: jos saugo ir apdoroja jutiklių duomenis, vartotojams suteikia galimybę nuotoliniu būdu valdyti įrenginius, analizuoti duomenis, užtikrina sąsajas tarp įvairių paslaugų, taip pat rūpinasi saugumu. Tokia daiktų interneto infrastruktūra egzistuoja įvairiose sferose. Šiuo metu populiarėja automatizuotų namų sprendimai, išmanūs miesto ir energetikos pritaikymai, pramonės įmonėse pasitelkiami nuotoliniu būdu valdomi įrenginiai, prie tinklo jungiamos ir kontroliuojamos medicininės sistemos. Daiktų interneto svarba šiuolaikiniam gyvenimui – akivaizdi.

Vis dėlto, šių sprendimų saugai iki šiol skiriama nepakankamai dėmesio, dėl ko skirtinguose daiktų interneto ekosistemos sluoksniuose ryškėja kibernetinio saugumo iššūkiai. Ne išimtis ir valdiklių valdymo platformos, kurios kontroliuoja išmaniuosius įrenginius. Tokios platformos – patrauklus taikinys piktavaliams, o jų sukompromitavimas gali turėti didelės žalos, todėl konfidencialumo, integralumo bei prieinamumo užtikrinimas šiose platformose yra vienas iš prioritetų apsaugant daiktų interneto sprendimus.

Valdiklių valdymų platformos pasižymi silpna prieigos kontrole. Naudojami tradiciniai sprendimai yra statiški, todėl nėra tinkami dinamiškai daiktų interneto aplinkai. Taigi, kyla poreikis ieškoti šiai aplinkai tinkamų autorizacijos metodų.

Darbo tikslas ir uždaviniai

Darbo tikslas – sudaryti valdiklių saugią nuotolinio valdymo platformą panaudojant prieigos kontrolės metodą bei įvertinti šio metodo veiksmingumą.

Baigiamojo darbo tikslui įgyvendinti išsikelti uždaviniai:

- atlikti esamų valdiklių valdymo platformų, kibernetinių grėsmių ir galimų sprendimo būdų analizę;
- sudaryti valdiklių valdymo platformos projektą panaudojant prieigos kontrolės metodą;
- realizuoti eksperimentinę valdiklių valdymo platformą su prieigos kontrolės metodu;
- ištirti realizuotą ir valdiklių valdymo platformoje integruotą prieigos kontrolės metodą bei įvertinti gautus rezultatus.

Dokumento struktūra

Darbe pateikiamos šios dalys:

- Pirmame darbo skyriuje atlikta daiktų interneto platformų bei visos daiktų interneto infrastruktūros analizė. Taip pat aprašytos kibernetinio saugumo grėsmės, kylančios valdiklių valdymo platformoms. Išnagrinėti prieigos kontrolės metodai bei daiktų interneto platformose naudojami prieigos sprendimai.

- Antrajame skyriuje išskirta, kokius rezultatus tikimasi gauti iš sudaryto prieigos kontrolės metodo. Pateikta darbo vizija bei detalizuoti metodo veikimo principai.
- Trečiasis skyrius skirtas realizacijos aprašymui. Jame aprašomas valdiklių valdymo platformos ir prieigos kontrolės metodo projektas, pateikiama techninė ir programinė įranga, naudojama realizacijai, detalizuojamas valdiklių valdymo platformos bei metodo veikimas.
- Ketvirtame skyriuje išdėstyti eksperimentuose tiriami parametrai bei scenarijai, detalizuojama jų eiga. Pateikiami scenarijų metu gauti rezultatai. Taip pat aprašomas valdiklių valdymo platformos paruošimas, naudojamas tyrimui atlikti.

1. Valdiklių valdymo platformų ir prieigos metodų analizė

Šiame skyriuje atliekama probleminės srities analizė. Aiškinamasi, kas yra valdiklių valdymų platformos, kaip veikia, kam jos skirtos ir kokias technologijas naudoja. Taip pat aptariamos tokioms platformoms kylančios kibernetinės grėsmės. Analizuojami saugumo metodai ir priemonės šioms grėsmėms spręsti.

1.1. Daiktų internetas

Valdiklių valdymo platformos – tai sudedamoji daiktų interneto ekosistemos dalis. Daiktų interneto sąvoka naudojama apibrėžti informacinių sistemų infrastruktūrą, susidedančią iš fizinių įrenginių, tarpusavyje sujungtų į tinklą su kitomis technologijomis [1]. Tokios sistemos naudojamos nuotoliniu būdu rinkti duomenis apie aplinkos parametrus, taip pat siekiant fiziškai daryti poveikį šiai aplinkai. Daiktų interneto sistemos taikomos įvairiose gyvenimo srityse: energetikos sektoriuje, pramonėje nuotoliniu būdu valdant pramoninius įrenginius, taip pat išmaniųjų miestų, sveikatos apsaugos ir automatizuotų namų sprendimuose. Pastebima, jog auga daiktų interneto įrenginių, taip pat platformų skaičius. „Cisco Systems“ įmonė prognozuoja, jog iki 2023 metų galo bus pasiekta virš 29,3 mlrd. tinkle sujungtų įrenginių, iš kurių pusė bus nedaug resursų turintys daiktų interneto M2M (*machine to machine*) įrenginiai [2].

1.1.1. Architektūriniai sluoksniai

Daiktų interneto technologijas galima išskaidyti į kelis esminius architektūrinius sluoksnius. Tyrėjai pateikia įvairius skaidymo būdus, tačiau patys populiariausi – tai trijų bei penkių lygių architektūriniai skaidymai [3], kurie matomi 1 pav.

1 pav. Daiktų interneto architektūriniai sluoksniai [3]

Pirmasis sluoksnis – tai suvokimo (angl. *perception*) sluoksnis, jame jutikliai renka duomenis apie aplinką, o vykdikliai vykdo veiksmus aplinkoje. Šiame sluoksnyje taip pat veikia mikrovaldikliai,

naudojami surinkti duomenis iš jutiklių, atlikti pradinį duomenų apdorojimą ir perduoti juos per tinklo sąsajas. Jie užtikrina vykdomųjų komandų išsiuntimą vykdikliams.

Tinklo (angl. *network*) sluoksnyje komunikaciniu kanalu perduodami duomenys tarp įrenginių. Čia veikia tokie protokolai kaip Wi-Fi, Zigbee, Sigfox, 3G, 4G, 5G bei kiti. Taigi, šis sluoksnis leidžia keistis duomenimis tarp atskirtų įrenginių ar sistemų.

Pats aukščiausias architektūrinis lygis – tai taikomasis (angl. *application*). Jame vykdomas duomenų vizualizavimas, veikia programos bei vartotojams skirta grafinė sąsaja. Taikomajame sluoksnyje galima išskirti tarpinės programinės įrangos (angl. *middleware*) sluoksnį, atsakingą už logines operacijas, įrenginių valdymą, duomenų saugojimą ir apdorojimą, duomenų bazių valdymą, debesų kompiuteriją, duomenų analitiką bei saugumą [4].

Ekspertai pateikia dar vieną daiktų interneto ekosistemos suskaidymą, kai architektūra padalinama į 3 lygius [5]. Galinių skaičiavimų (angl. *edge computing*) sluoksnis skiriamas fiziniams įrenginiams, t. y. tiek jutikliams, tiek vykdikliams, taip pat valdikliams. Šiame sluoksnyje lokaliai atliekamas pirminis duomenų apdorojimas. To pavyzdys galėtų būti įvairūs įrenginiai sujungti su „Arduino“, „ESP8266“ ir kitais valdikliais. Toliau duomenys perduodami į rūko (angl. *fog*) mazgus, turinčius žymiai daugiau resursų ir galinčius atlikti sudėtingesnius veiksmus, duomenų analizę. Šio sluoksnio pavyzdžiu galėtų būti „Raspberry Pi“ mikrokompiuteriai. Paskutinis sluoksnis, kuriame saugomi duomenys ir priimami sprendimai – tai debesijos (angl. *cloud*) sluoksnis.

1.1.2. Daiktų interneto platformos

Daiktų interneto platformos veikia taikomajame sluoksnyje ir yra pagrindas visai valdiklių infrastruktūrai. Jos sujungia skirtingus įrenginius, paslaugas ir programas į vieną visumą. Pagrindiniai tokių platformų funkcionalumai [6]:

- fizinių įrenginių valdymas;
- duomenų surinkimas, integravimas ir valdymas;
- duomenų apdorojimas ir vizualizavimas;
- duomenų teikimas programoms ir paslaugoms.

Fizinių įrenginių valdymui priskiriamas komandų vykdymas, įrenginių konfigūravimas, nustatymų keitimas nuotoliniu būdu. Tam atlikti, pirmiausia, reikia užtikrinti komunikaciją tarp įrenginio ir platformos. Šiai komunikacijai paprastai naudojami tokie žinučių siuntimo protokolai kaip MQTT, CoAP, AMQP, DDS, XMPP, HTTP [7]. Pirmajam funkcionalumui reikėtų priskirti ir programinės įrangos atnaujinimus, kurie gali būti atliekami nuotoliniu būdu panaudojus OTA (*Over-the-Air*) metodą.

Duomenims surinkti naudojami anksčiau minėti žinučių siuntimo protokolai. Gauti duomenys išsaugomi duomenų bazėje ir integruojami į platformą. Vartotojams leidžia valdyti duomenis, t. y. juos trinti, keisti, eksportuoti.

Daiktų interneto platformoje surinkti duomenys atvaizduojami vartotojo sąsajoje. Grafinėje sąsajoje pateikiama patogi duomenų vizualizacija. Neretai pateikiamos diagramos, statistiniai duomenys, leidžiantys vartotojui geriau suprasti aplinką, kurioje veikia įrenginiai.

Ketvirtajam funkcionalumui įgyvendinti, naudojami API (aplikacijų programavimo sąsajos). Naudojant šias sąsajas, daiktų interneto platformos paslaugomis gali naudotis kitos sistemos, kurioms teikiami valdiklių infrastruktūroje surinkti ir platformoje apdoroti duomenys.

1.2. Valdiklių valdymo platformoms kylančios saugumo grėsmės

Daiktų interneto taikymai, kaip ir visos IT sistemos, neišvengiamai susiduria su saugumo iššūkiais. „Nokia Corporation“ įmonės duomenimis, 2020 metais infekuotų daiktų interneto įrenginių procentinė dalis pakilo iš 2019 metais fiksuotų 16.17% į 32.72% [8]. Saugumo problemos tampa vis reikšmingesnė, todėl kyla būtinybė ieškoti sprendimų, padėsiančių užtikrinti daiktų interneto sprendimų saugumą.

1.2.1. Išskirtinumas

Daiktų interneto sprendimams, kaip ir visoms sistemoms, būtina užtikrinti pagrindinius informacijos saugumo uždavinius, t. y. konfidencialumą, vientisumą ir prieinamumą. Vis dėlto, šioje srityje kylančios kibernetinio saugumo problemos bei galimi saugos metodai yra kitokie, negu tie, su kuriais susiduria tradiciniai IT sprendimai. Taip yra dėl daiktų interneto infrastruktūrai būdingų savitų savybių [9].

Visų pirma, sprendimuose naudojami įrenginiai yra ribotų resursų. Tai reiškia, jog turi nepakankamai atminties, talpos bei skaičiavimo resursų, daugelis jų veikia naudojant baterijas, todėl tradicinės saugumo technologijos nėra tinkamos.

Jungiamumas taip pat yra būdinga charakteristika. Įrenginiai, naudojami daiktų interneto ekosistemoje, dažnu atveju yra nutolę nuo vartotojo, valdymo platformos bei kitų įrenginių. Tai sunkina jų priežiūrą, o komunikacijai užtikrinti, būtinas sujungimas į tinklą. Dažnu atveju duomenų keitimasis vykdomas belaidžiais tinklais, todėl kyla papildomų grėsmių.

Daiktų interneto infrastruktūra pasižymi masiškumu ir nevienalytiškumu. Naudojama daug įrenginių, kurie komunikuoja tiek tarpusavyje, tiek su platformomis. Taigi, būdingi dideli informacijos srautai, sudėtingas įrenginių valdymas. Įrenginiai dažnu atveju yra skirtingi, t. y. naudoja skirtingas technines ir programines priemones. Visa tai išplečia galimą atakos paviršių.

Dinamiškumas – tai dar viena išskirtinė savybė, pastebima daiktų interneto sprendimuose. Įrenginiai, kaip ir jų savybės ar būsenos, nuolat kinta. Jie gali būti įjungi, atjungti, miego būsenoje, gali kisti jų priklausomybės ir ryšiai su vartotojais. Ypatingai dinamiška aplinka, kurioje veikia daiktų interneto įrenginiai: gali kisti lokacija, greitis, temperatūra ir t. t.

Taigi, daiktų interneto ekosistemos saugumas yra sudėtingas uždavinys, kadangi infrastruktūrai būdingos netradicinės savybės. Tai reikalauja atskiros saugumo iššūkių analizės.

1.2.2. Daiktų interneto infrastruktūros saugumo grėsmės

Kibernetinio saugumo grėsmės kyla visai daiktų interneto infrastruktūrai. Remiantis OWASP (*Open Web Application Security Project*) 2018 metais sudarytu sąrašu, pateikiama dešimt problematiškų iššūkių daiktų interneto srityje [10]:

1. Silpni, lengvai atspėjami, įkoduoti (angl. *hardcoded*) slaptažodžiai, kurių negalima pakeisti ir kurie naudojami galinių durų prieigai tiek įrenginio programinei įrangai, tiek klientinėms programoms.

2. Nesaugios tinklo paslaugos, kompromituojančios konfidencialumą, vientisumą, prieinamumą, veikiančios įrenginiuose bei matomos viešai internete.
3. Nesaugios sąsajos (pvz., žiniatinklio svetainių ar debesijos paslaugų API). Joms trūksta autentifikacijos, autorizacijos, taip pat turi silpną šifravimą bei įvesties ir išvesties filtravimą.
4. Nesaugus įrenginių atnaujinimams. Įrenginių programinė įranga nėra tikrinama, persiuntimo metu trūksta šifravimo, nėra galimybės atstatyti sistemą į buvusią būseną.
5. Nesaugių bei pasenusių komponentų, bibliotekų naudojimas.
6. Nepakankama privatumo apsauga tiek įrenginiuose, tiek visoje ekosistemoje.
7. Nesaugus duomenų perdavimas ir saugojimas nesirūpinant nei šifravimu, nei prieigos kontrolės mechanizmais.
8. Įrenginių valdymo trūkumas neturint mechanizmų užtikrinti saugią jų priežiūrą ir valdymą, naudojamų įrenginių kontrolę, programinės įrangos atnaujinimus, eksploatacijos nutraukimą (angl. *decommissioning*), sistemų stebėjimą.
9. Nesaugūs gamintojų numatyti nustatymai, trūkumas mechanizmų, kuriais vartotojai būtų apriboti pakeisti įrenginių konfigūracijas.
10. Fizinė apsaugos (angl. *physical hardening*) trūkumas, įskaitant įrenginius su įjungtais derinimo (angl. *debug*) prievadais ar jautrios informacijos saugojimą išimamose atminties kortelėse.

Remiantis aptartu sąrašu, 1 lent. pateikti architektūriniai sluoksniai su jiems būdingomis saugumo problemomis.

1 lentelė. Daiktų interneto sprendimų saugumo iššūkiai skirtinguose architektūriniuose sluoksniuose

Saugumo iššūkis	**Taikomasis sluoksnis**	**Tinklo sluoksnis**	**Suvokimo sluoksnis**
Silpni arba įkoduoti slaptažodžiai		✓	✓
Nesaugios tinklo paslaugos		✓	✓
Nesaugios sąsajos	✓		
Nesaugūs įrenginių atnaujinimai	✓		✓
Nesaugių komponentų naudojimas	✓		✓
Nepakankama privatumo apsauga	✓	✓	✓
Nesaugus duomenų perdavimas ir saugojimas	✓	✓	✓
Įrenginių valdymo trūkumas	✓		
Nesaugūs gamintojų numatyti nustatymai		✓	✓
Fizinės apsaugos trūkumas		✓	✓

Taigi, visuose daiktų interneto ekosistemos sluoksniuose kyla kibernetinio saugumo problemų. Dalis jų panašios, dalis jų skiriasi. Vis dėlto, nemaža dalis grėsmių glūdi taikomajame sluoksnyje, kur veikia daiktų interneto platformos.

1.2.3. Daiktų interneto platformų saugumo grėsmės

Šiame skyrelyje plačiau nagrinėjama kiekviena taikomojo sluoksnio saugumo grėsmė ir jos priežastys.

Nesaugios sąsajos. Į šią kategoriją būtų galima įtraukti visas žiniatinklio paslaugoms būdingas saugumo problemas, kurias aprašo atskiras „OWASP Top 10“ sąrašas [11]. Vienos esminių

problemų – tai prasta prieigos kontrolė, kai nesugebama užtikrinti, jog vartotojai negali peržengti savo prieigos teisių ribų ir taip atlikti neautorizuotus veiksmus, taip pat SQL injekcijų galimybė, kai nėra patikrinama vartotojo įvestis. Kiti saugumo iššūkiai: nesaugus dizainas, netinkami saugumo nustatymai, netinkami arba nepakankami kriptografiniai sprendimai, prasta identifikacija bei autentifikacija, programinės įrangos vientisumo patikrinimų nebuvimas, įvykių neregistravimas ir nestebėjimas.

Silpna prieigos kontrolė ir autentifikacija. Autentifikacija – tai procesas, kurio metu identifikuojamas vartotojas ir jam suteikiama prieiga prie sistemos. Daiktų interneto sprendimuose jis reikalingas norint užtikrinti, jog tik patikrinti vartotojai bei įrenginiai galėtų siųsti bei gauti duomenis, taip užtikrinant sistemos patikimumą. Prieigos kontrolė (angl. *access control*) reikalinga užtikrinti autorizacijos procesą tam, kad resursus pasiektų tik tie, kuriems yra leidžiama juos pasiekti nepažeidžiant saugumo. Tradiciniai prieigos kontrolės bei autentifikacijos metodai daiktų interneto platformose yra silpni, t. y. neatsižvelgia į vartotojų ir įrenginių įvairovę, jų specifiką ir dinamišką aplinką [12]. Naudojami metodai yra statiniai, nevertina aplinkos, todėl negali užtikrinti mažiausių teisių principo.

Kriptografijos problemos. Tiek duomenų vientisumą, tiek duomenų konfidencialumą galima apsaugoti kriptografiniais metodais. Pavyzdžiui, saugiam atnaujinimų procesui užtikrinti, naudojama PKI (viešojo rakto infrastruktūra), tačiau dauguma daiktų interneto įrenginių turi nepakankamai skaičiavimo resursų, kad galėtų efektyviai įgyvendinti šios infrastruktūros algoritmus. Taigi, dažnai daiktų interneto sprendimuose duomenų saugumui užtikrinti nėra naudojami kriptografiniai metodai arba naudojami pasenę ir silpni. Dėl šios priežasties duomenys (pvz., programinės įrangos atnaujinimai) nėra saugiai perduodami ir leidžia piktavaliui sugadinti ar pakeisti žinučių paketus ir taip sukompromituoti sistemas [13].

Nesaugi komunikacija. Norint saugiai perduoti duomenis, žinutės privalo būti tinkamai šifruojamos tarp galinių mazgų ir taikomojo sluoksnio sprendimų [14]. Galiniai taškai neretai naudoja silpnus šifravimo algoritmus bei protokolus, taip pat netinkamus numatytuosius šifravimo nustatymus, kurių naudojimas kelia šniukštinėjimo (angl. *sniffing*) atakos grėsmę. Atskiras dėmesys turėtų būti skiriamas žinučių siuntimo protokolams (pvz.: MQTT, CoAP, AMQP), kurių veikimas labiausiai priklauso nuo trečiojo architektūrinio sluoksnio. Kiekvienas protokolas turi jam būdingų saugumo silpnybių [15]. Tokių protokolų naudojimas kelia problemų dėl prenumeravimo (angl. *subscribtion*) bei siuntimo (angl. *publication*), kuris gali būti neautorizuotas ar neautentifikuotas, jeigu naudojami pagal nutylėjimą brokerio parinkti nustatymai. Galima žmogus-viduje ataka, jeigu nėra taikomi kriptografiniai sprendimai ir žinutės siunčiamos atviru tekstu. Taip pat susiduriama su galimybe klastoti adresus.

Nesaugūs komponentai. Nesaugūs trečiųjų šalių komponentai gali leisti įsilaužėliui prieiti prie jautrių duomenų, įvykdyti norimas komandas. Šio pažeidžiamumo pavyzdžiu galėtų būti pasenusių bibliotekų naudojimas programinėje įrangoje, taip pat nesaugių operacinių sistemų eksploatacija ar kitokių komponentų įtraukimas iš sukompromituotų tiekimo grandinių.

Privatumo problemos. Įrenginiai renka didelius kiekius duomenų iš įvairių įrenginių. Tokia informacija gali teikti jautrią informaciją, pavyzdžiui, informaciją apie asmens arba jo nuosavybės buvimo vietą, žmogaus atvaizdą, jo fizines savybes, namų įrenginių būseną, taip pat gali pažeisti tokius reikalavimus kaip BDAR (Bendrasis duomenų apsaugos reglamentas) [16]. Daiktų interneto

platformoje išanalizuoti duomenys gali teikti informaciją apie žmogaus įpročius, jo asmeninį gyvenimą, taip pat identifikuoti patį žmogų. Netinkamose rankose atsidūrusi informacija gali būti panaudota piktavališkais tikslais. Taigi, būtina ieškoti būdų, kaip užtikrinti, jog duomenys tiek perdavimo, tiek saugojimo metu išliktų privatūs ir atitiktų reikalavimus bei jų valdymas ir panaudojimas būtų kontroliuojamas.

Įrenginių valdymo trūkumas. Daiktų interneto sprendimuose paprastai vyrauja daugybė skirtingų įrenginių. Norint užtikrinti jų saugumą, būtina imtis ne tik techninių priemonių apsaugoti duomenis, bet ir užtikrinti saugų įrenginių valdymo ir priežiūros procesą. Administratoriai, turėdami ribotas galimybes valdyti ir prižiūrėti įrenginius negali efektyviai reaguoti į atsirandančius saugumo iššūkius (pvz., numatytųjų silpnų konfigūracijų naudojimas įrenginiuose), pažeidžiamumus. Tokiu pavyzdžiu galėtų būti paslaugų sutrikdymo (angl. *service interruption*) atakos [17]. Apibendrinus, daiktų interneto sprendimams reikia turėti centralizuotą galimybę saugiai konfigūruoti įrenginius, parinkti saugius nustatymus, stebėti įrenginių įvykių įrašus bei sistemos būseną, gauti įspėjimus, atnaujinti įrenginius, nutraukti jų naudojimą.

Detaliai išnagrinėjus daiktų platformoms kylančias saugumo grėsme, pastebėta, kad nemenka dalis problemų susijusios su silpna prieigos kontrole. Apibendrinus, reikalinga papildoma prieigos metodų ir sprendimų, naudojamų daiktų interneto platformose, analizė.

1.3. Valdiklių valdymo platformų prieigos kontrolės metodai

Prieigos kontrolę galima apibūdinti kaip mechanizmą, užtikrinantį, kad tik patikimi subjektai turi galimybę prieiti prie resursų [18]. Šis mechanizmas nusako, kaip vartotojai ir sistemos gali sąveikauti, atsižvelgiant į politikas (angl. *policies*). Bendruoju atveju, kompiuterinėse sistemose prieigos kontrolė sprendžia, ar subjektas (pvz.: procesas, įrenginys, žmogus, paslauga, programa) gali atlikti tam tikrą veiksmą (pvz.: skaityti, rašyti, atnaujinti, vykdyti) su objektu (pvz.: duomenų bazė, failas, servisas, įrenginys) pagal nustatytas sistemos prieigos politikas. Politikos – tai nustatytos sistemoje saugomos sąlygos, kurioms esant suteikiama prieiga prie objektų. Apibendrintas prieigos kontrolės veikimas pateiktas 2 pav.

2 pav. Pagrindiniai prieigos kontrolės komponentai [19]

Prieigos valdymo mechanizmas užtikrina CIA (ang. *Confidentiality, Integrity, Availability*) triados modelį [20]:

- **Konfidencialumas.** Užtikrinama, jog informaciją galės pasiekti tik autorizuoti vartotojai.
- **Integralumas.** Užtikrinama, jog tik autorizuoti vartotojai galės daryti informacijos pakeitimus.
- **Prieinamumas.** Užtikrinama, jog informacija visuomet bus prieinama gavus prieigos užklausas.

Toliau pateikiami populiariausi prieigos kontrolės mechanizmai [19].

1.3.1. DAC metodo analizė

DAC (angl. *Discretionary Access Control*) – tai diskrecinis prieigos valdymo metodas, kuris buvo sukurtas operacinėms sistemoms ir vėliau pradėtas taikyti daiktų interneto sprendimams [21]. Jį naudojant įrenginių savininkas nurodo, kuris vartotojas turi teisę gauti prieigą prie įrenginio, tam apibrėžia taisykles, kuriose nurodomos galimos operacijos kiekvienam vartotojui. Įgyvendinant šį prieigos metodą, naudojamos prieigos matricos, autorizacijos lentelės bei prieigos kontrolės sąrašai (angl. *ACL – access control lists*).

Privalumai:

- visiška įrenginio kontrolė nurodant, kas gali gauti prieigą ir kokiomis sąlygomis galima atlikti nurodytas operacijas.

Minusai:

- jeigu vartotojas yra daugybės įrenginių savininkas, toks metodas apsunkina auditavimą ir administravimą, kadangi neturint centralizuoto metodo, vartotojas privalo kiekvienam įrenginiui atskirai nurodyti prieigos teises;
- metodas yra statiškas ir nekintantis.

1.3.2. MAC metodo analizė

MAC (angl. *Mandatory Access Control*) – tai imperatyvinis prieigos valdymo mechanizmas, besiremiantis visų subjektų ir objektų klasifikacija, t. y. visi gyvi ir negyvi vartotojai bei paslaugos turi jiems priskirtas saugumo žymes (angl. *label*), kurios apibrėžia, kokio jautrumo informaciją (pvz.: slapta, visiškai slapta, konfidenciali) galima prieiti arba sukurti [22]. Tam, kad šis modelis veiktų tinkamai, reikalingi žymių apribojimai, kurie leistų tik ribotai grupei žmonių atlikti subjektų žymių pakeitimus.

Privalumai:

- saugesnis nei DAC;
- spartesnis prieigos teisių nustatymas, kai yra daugiau subjektų;

Minusai:

- sudėtingas bei brangus įgyvendinimas, priežiūra. Tai ypač galioja dinamiškose aplinkose, kur reikalingas lankstumas bei greitis (pvz., pacientų sveikatos užtikrinimo procese);
- savininkas neturi galimybės pats kontroliuoti išteklių, kadangi naudojamas centralizuotas prieigos teisių mechanizmas;
- neatsižvelgiama į konkrečią situaciją ir detales, kadangi viską reguliuoja sistema.

1.3.3. RBAC metodo analizė

RBAC (angl. *Role-Based Access Control*) – tai sistema, kurios pagrindą sudaro vartotojai, rolės bei teisės [23]. Kiekvienam vartotojui suteikiama rolė bei tai rolei priskirtos teisės atlikti tam tikrus veiksmus, nustatytus pagal politiką (angl. *policy*). Pati rolė apibrėžia funkcionalumą hierarchinėje struktūroje. Pasikeitus rolei, pasikeičia ir vartotojo teisės atlikti veiksmus.

Privalumai:

- užtikrinamas paprastumas didelėje organizacijoje, kadangi vartotojo rolė gali būti pakeista pagal jos atsakomybes organizacijoje, nepakeičiant priskirtų teisių.

Minusai:

- rolės yra statinės ir turi būti iš anksto apibrėžtos centrinio administratoriaus;
- rolių skaičius gali žymiai išaugti, priklausomai nuo taikymo sudėtingumo, ir taip padaryti visą sistemą gremėzdiška.

1.3.4. ABAC metodo analizė

ABAC (angl. *Attribute-Based Access Control*) – tai panašus į prieš tai nagrinėtą metodą, tačiau vietoj rolių naudojami politikos nustatyti požymiai (angl. *attribute*) [24]. Šie požymiai gali būti įvairios savybės, nusakančios subjektus (pvz.: vardas, amžius), objektus, aplinką (pvz.: laikas, vieta), sąlygas. Politikos sukuriamos tam, kad jomis naudojantis būtų galima suteikti prieigos teises pagal politikoje aprašytus požymius.

Privalumai:

- žymiai didesnis lankstumas negu RBAC, kadangi prieigos teisės sukuriamos konkrečiam vartotojui, o ne rolei;
- užtikrinamas saugumas, kadangi suteikiamos tik tos teisės, kurių reikia.

Minusai:

- sudėtingas konfigūravimas ir administravimas, priklausomai nuo aplinkos, administratoriui gali tekti sukurti daug politikų, atsižvelgiančių į skirtingus aplinkos požymius.

1.3.5. Cap-BAC metodo analizė

Cap-BAC (angl. *Capability-Based Access Control*) – tai netradicinis prieigos valdymo modelis, kuris paremtas nesuklastojamais žetonais (angl. *tokens*), nusakančiais vartotojui suteiktas teises. Šis metodas orientuotas į prieigos sprendimų pateikimą ir įgyvendinimą, o ne sprendimų priėmimą. Autentifikacijos metu žetonai yra sukuriami IAM (angl. *Identity and Access Management*) sistemos ir išsiunčiami vartotojui [25]. Gauti žetonai nusako, kokias paslaugas vartotojui yra leidžiama pasiekti bei kokius veiksmus galima atlikti. Norint atlikti tam tikrą operaciją, vartotojas privalo pateikti jam suteiktą žetoną.

Privalumai:

- lankstus metodas, todėl tinka naudojant dėvimus įrenginius, taip pat pramogoms ar namams, kur taikymas yra spontaniškas ir be aiškių ryšių.

Minusai:

- sudėtingas administravimas, kadangi reikia sukurti žetonus ir juos saugoti.

1.3.6. Prieigos kontrolės metodų klasifikacija

Visus naudojamus prieigos kontrolės metodus galima suklasifikuoti į tiesioginius (angl. *direct*) ir netiesioginius (angl. *indirect*) [19]. Pirmuoju atveju vartotojas pats atlieka veiksmus pagal jam suteiktas prieigos teises, antruoju – prieigos teisės deleguojamos kitam subjektui. Taip pat galima klasifikacija pagal architektūrą:

Centralizuota. Ši architektūra pasižymi tuo, kad prieigos kontrolės mechanizmas realizuojamas centriniame vienete (pvz., centriniame serveryje), kur priimami sprendimai pagal turimas politikas. Šis vienetas gali būti realizuotas ir kaip šliuzas (angl. *gateway*), supaprastinantis komunikaciją, taip pat kaip atskira nutolusi sistema. Galiniai įrenginiai veikia kaip pasyvūs informacijos tiekėjai. Vartotojai, norėdami gauti prieigą prie įrenginių ar kitokių resursų, privalo jungtis per API. Tokia architektūra yra paprasta administravimo požiūriu, ją lengva prižiūrėti ir audituoti, prieigos sprendimams priimti nenaudojami ir taip riboti galinių įrenginių resursai. Vis dėlto, ši architektūra kenčia nuo tradicinių centralizuotos architektūros trūkumų, t. y. sugedus šiam prieigos kontrolės vienetui, nustoja veikti visa sistema. Kita problema – tai negebėjimas panaudoti konteksto (angl. *contextual*) informacijos.

Centralizuota ir kontekstinė. Tokia architektūra panaudoja konteksto informaciją (pvz.: vieta, laikas, aplinkos statusas), kuri susiejama su galiniais įrenginiais. Taigi, įrenginiai tampa aktyvūs informacijos tiekėjai. Šis požiūris taip pat paveldi centriniai architektūrai būdingus trūkumus, taip pat reikalauja papildomos komunikacijos su įrenginiais, teikiančiais kontekstinę informaciją. Komunikacija nuo įrenginio iki sistemos negali būti visiškai apsaugota [26].

Paskirstyta (angl. *Distributed*). Šioje architektūroje nėra centrinio vieneto, o prieigos kontrolė realizuojama galiniuose įrenginiuose. Įrenginiai visiškai kontroliuoja jų informaciją, išmetus tarpinį vienetą, galima užtikrinti saugią komunikaciją nuo vieno galo iki kito. Didžiausia problema yra tai, kad daiktų interneto įrenginiai turi ribotus resursus, kurie papildomai užimami apdorojant prieigos kontrolės logiką bei saugojant sprendimus.

1.3.7. Reikalavimai renkantis prieigos kontrolės metodą

Rekomenduojama apsvarstyti šiuos bruožus renkantis prieigos kontrolės mechanizmą daiktų interneto sprendimams [27]:

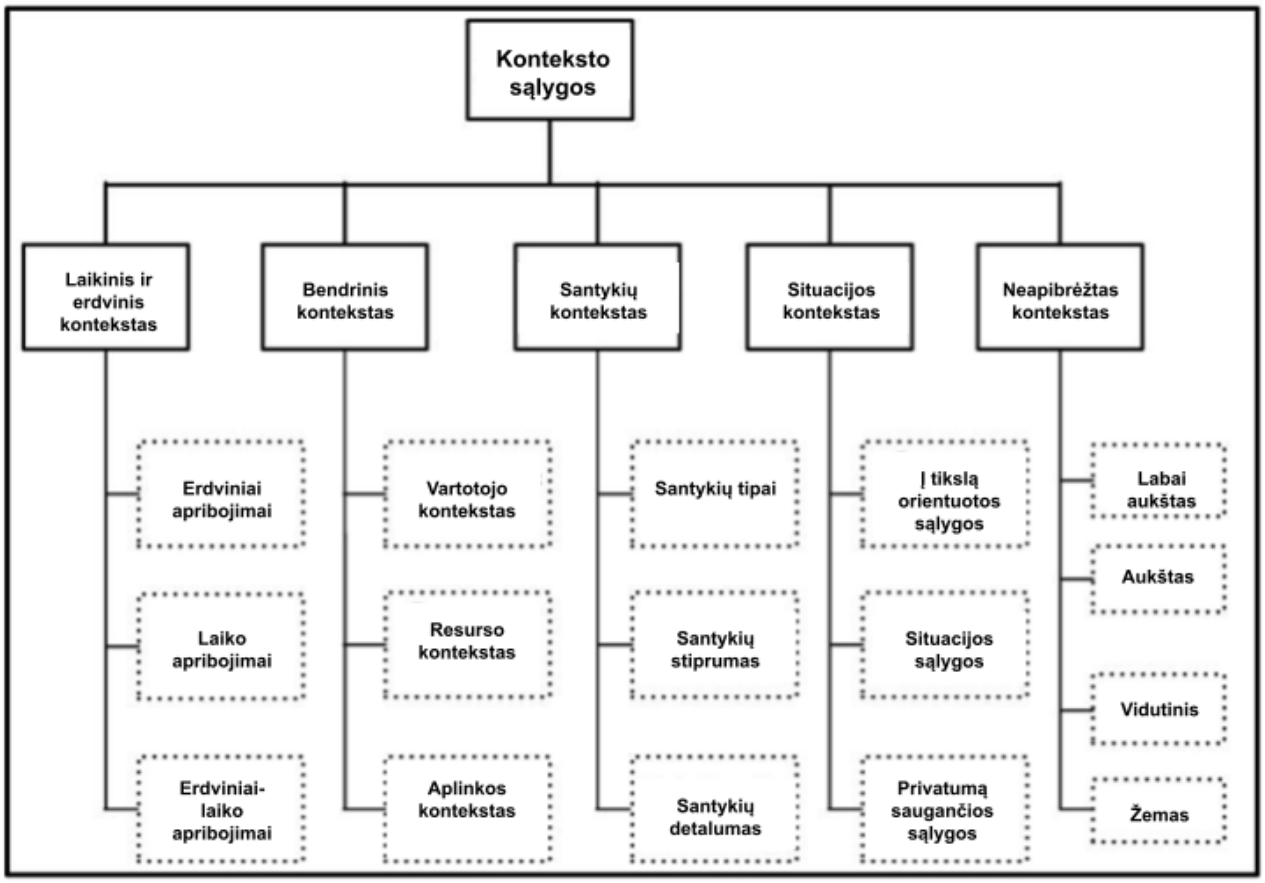
- paprastas administravimas ir konfigūravimas;
- plečiamumas turėtų leisti lengvai pridėti naujus vartotojus bei objektus, neapsunkinant prieigos kontrolės mechanizmo administravimo;
- lankstumas ir galimybė atsižvelgti į kontekstą priimant sprendimus, t. y. turėtų būti atsižvelgiama ne tik į statines teises, tačiau ir į aplinką, kintančias situacijas;
- detalumas turėtų leisti ekspresyviai aprašyti politikos prieigos taisykles tam, kad būtų atsižvelgiama į skirtingus bruožus;
- dinaminių požymių palaikymas, taip leidžiant prisitaikyti prie individualių scenarijų;
- galimybė integruotis su trečiosiomis šalimis;
- lengvasvoriškumas, t. y. galiniai įrenginiai neturėtų būti užkraunami sudėtingais skaičiavimais.

1.3.8. Kontekstas

Konteksto įvertinimas prieigos kontrolės mechanizmuose yra viena esminių savybių, norint pagerinti daiktų interneto platformų saugą. Vertinant kontekstą, galima išskirti tris grupes, į kurias skirstomas kontekstas [28]:

- **Į vartotoją orientuotas kontekstas**. Tokia informacija apibūdina vartotoją, t. y. tiek resurso prašantį vartotoją, tiek jo savininką ar bet kokį kitą aplinkos asmenį.
- **Į resursą orientuotas kontekstas.** Tai informacija apie naudojamus duomenis ir duomenų šaltinius (pvz.: įrenginio profilį, elgesį, gamintoją, komponentus).
- **Į aplinką orientuotas kontekstas**. Ši informacija pristato resursus bei vartotojus supančią aplinką (pvz.: geografinę lokaciją, laiką).

Platesnis konteksto grupavimas pateikiamas 3 pav.



3 pav. Konteksto informacijos grupės [28]

Priklausomai nuo sprendimų pritaikymo, svarbu apsvarstyti dinamišką informaciją, kuri yra nuolatos kintanti laike. Pavyzdžiui, naudojant išmaniuosius dėvimus įrenginius, jų buvimo vieta gali nuolatos keistis. Taip pat reikėtų atkreipti dėmesį į priklausomybes, kadangi laikui bėgant gali keistis įrenginių savininkai. Taigi, svarbu apsvarstyti dabartinius ir buvusius įrenginio savininkus, bei kitokius ryšius. Dar viena konteksto grupė – tai netikslių ir nevienalyčių duomenų reikšmių pavertimas į aiškiai apibrėžtą ir suprantamą reikšmę. To pavyzdys galėtų būti rizikos lygių nustatymas, pagal kuriuos būtų atsižvelgiama autorizuojant vartotojus.

1.4. Valdiklių valdymo platformų prieigos kontrolės sprendimai

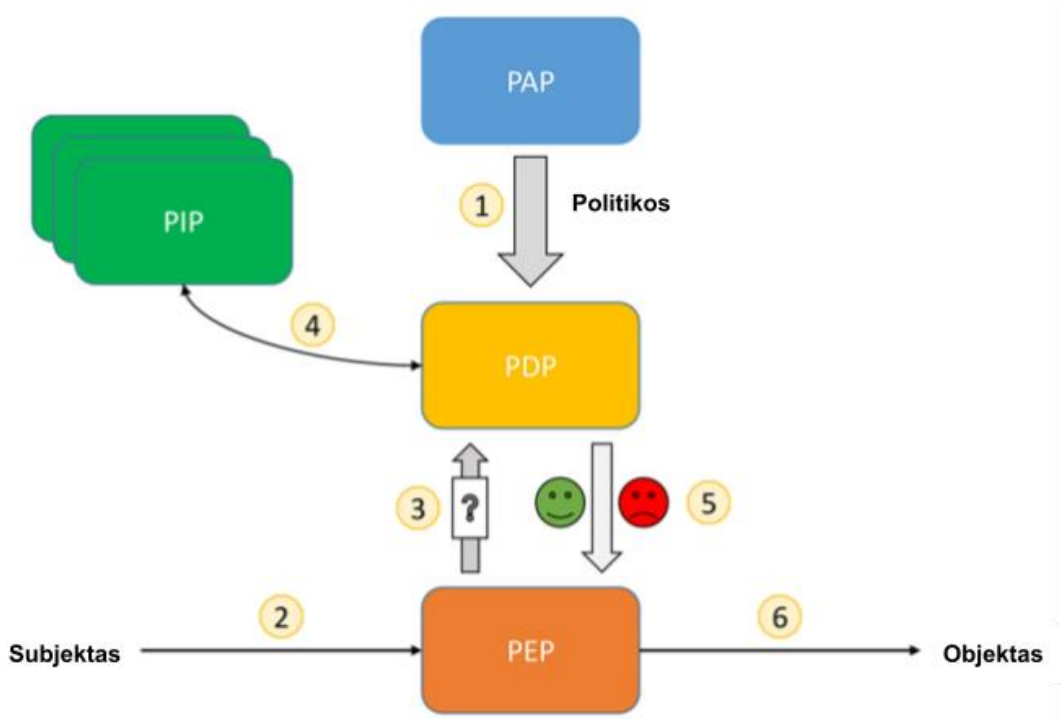
Kuriant prieigos kontrolės metodą valdiklių valdymo platformoms, svarbu išmanyti egzistuojančias technologijas. Šioje dalyje analizuojami sprendimai, naudojami prieigos kontrolės valdymo mechanizmams realizuoti, taip pat prieigos kontrolės standartai, lyginamos skirtingos prieigos valdymo realizacijos.

1.4.1. Architektūriniai sprendimai

Visi prieigos kontrolės sprendimai atlieka tą pačią funkciją bei yra sudaryti iš panašių loginių blokų, kurių funkcionalumą svarbu suprasti [29]:

- **PDP** (angl. *Policy Decision Point*) – tai blokas, naudojamas priimti sprendimus pagal turimas politikos taisykles.
- **PEP** (angl. *Policy Enforcment Point*) – tai blokas, naudojamas priimtiems sprendimams įvykdyti.
- **PAP** (angl. *Policy Administration Point*) – tai blokas, skirtas politikos taisyklėms sudarinėti.
- **PIP** (angl. *Policy Information Point*) tarnauja kaip šaltinis, iš kurio galima gauti tam tikrus aplinkai, objektams ar subjektams būdingus požymius. Taigi, PDP blokas naudojasi PIP sukaupta informacija tam, kad priimtų sudėtingus ir kontekstą įvertinančius sprendimus.

4 pav. pateikiama supaprastinta nagrinėtų architektūrinių blokų tarpusavio sąveika.



4 pav. Prieigos kontrolės sprendimų standartiniai architektūriniai blokai [29]

1.4.2. Standartai

OAuth 2.0 – tai vienas populiariausių autorizacijai skirtų karkasų (angl. *framework*), leidžiantis trečiųjų šalių aplikacijoms gauti prieigą prie vartotojo kontroliuojamų bei apribotų resursų [30].

Architektūra įtraukia 4 sistemos aktorius, t. y. *trečiąją šalį*, norinčią gauti prieigą prie resursų, *resursų savininką*, galintį suteikti bei atimti prieigą prie apsaugotų resursų, *resursų serverį*, valdantį resursus, bei *autorizacijos serverį*, kuris kontroliuoja autorizacijos procesą. Visi šie aktoriai užtikrina komunikaciją TLS (angl. Transport Layer Security) kanalu. Bendravimas tarp aktorių apima šiuos žingsnius:

1. klientas pradeda komunikaciją su resursų savininku;
2. resursų savininkas suteikia prieigą, atsiųsdamas autorizacijos kodą;
3. klientas pateikia gautą autorizacijos kodą autorizacijos serveriui;
4. autorizacijos serveris patikrina kodą bei suteikia žetoną su įvairiomis leidimo detalėmis (pvz.: laiko limitą, prieigos mastą, kt.);
5. klientas persiunčia gautą žetoną resursų serveriui;
6. resursų serveris patikrina žetono validumą ir suteikia prieigą prie resursų.

Toks metodas nėra tinkamas daiktų interneto ekosistemai dėl įrenginių resursų ribotumo, todėl naudojamas *Device Flow* plėtinys, įgalinantis ribotus resursus turinčius įrenginius gauti prieigos žetonus. Svarbu pabrėžti, jog prieiga suteikiama to pačio savininko servisui. Apibendrinus, šis standartas neatsižvelgia į daiktų interneto ribotumą ir heterogeniškumą, dinamišką aplinką, nesprendžia sprendimų priėmimo proceso problemų.

ACE (angl. *Authentication and Authorization in Constrained Environments*) darbo grupė išplečia anksčiau minėtą standartą, pritaikydama jį daiktų interneto aplinkai [31]. Šis sprendimas naudojasi CoAP žinutėmis, CWT (angl. *CBOR Web Token*) žetonais bei COSE (angl. *CBOR Object Signing and Encryption*) ir DTLS šifravimu. ACE naudoja PoP (angl. *Proof of Possession*) žetonus. Apibendrinus, visa tai reikalauja žymiai mažiau resursų bei atsižvelgia į tai, jog įrenginiai ne visada turės nuolatinę prieigą prie interneto, tačiau pats standartas sprendžia tik autorizacijos sprendimo dalijimosi ir vykdymo problematiką, o efektyvaus ir saugaus sprendimo priėmimo problema nėra sprendžiama.

Kantara Initiative darbo grupė sukūrė UMA (angl. *User-Managed Access*) autorizacijos specifikaciją, besiremiančią OAuth karkasu [31]. Šią autorizaciją stengiamasi pritaikyti prie nevienalytės ekosistemos. UMA leidžia deleguoti prieigą trečiajai šaliai, taigi, priešingai negu OAuth atveju, kai buvo suteikiama teisė vartotojo resursams pasiekti kitus jo resursus, UMA leidžia deleguoti prieigą kitam vartotojui. Kitas skirtumas – tai šios specifikacijos galimybės valdyti prieigos kontrolę centralizuotai ir pagal iš anksto apibrėžtas politikos taisykles.

1.4.3. Žetonai

Prieigos žetonai – tai tekstinės eilutės, nusakančios subjektui suteikiamos prieigos mastą, gyvavimo laikotarpį bei kitas savybes. Vienas populiariausių ir dažniausiai naudojamų standartų yra JWT (angl. *JSON Web Tokens*) – tai save apibūdinantį (angl. *self-descriptive*) formatą turintys žetonai, kurie saugiai perduoda informaciją JSON (angl. *JavaScript Object Notation*) formatu [32]. Tokia informacija yra patikima, kadangi ją galima pasirašyti HMAC (angl. Hash-Based Message Authentication Code) autentifikavimo kodu arba panaudojus RSA (Rivest–Shamir–Adleman) ar ECDSA (angl. *Elliptic Curve Digital Signature Algorithm*) privataus bei viešojo rakto porą.

1.4.4. Kalbos

Realizuojant prieigos kontrolės mechanizmus, viena iš problemų, su kuria susiduriama – tai politikos taisyklių aprašymas. Šiai problemai spręsti paprastai naudojamos prieigos kontrolės politikos taisyklių aprašymo kalbos, turinčios joms būdingą sintaksę bei operatorius, kurie specifikuoja subjektus, objektus bei veiksmus. Toliau aprašomos vienos labiausiai paplitusių kalbų [19].

XACML (angl. *eXtensible Access Control Markup Language*) – tai atviras, standartizuotas ir vienas populiariausių prieigos kontrolės aprašymo standartų, paremtas XML bei skirtas aprašyti žiniatinklio paslaugų politikos taisyklėms. Šis standartas dažnai naudojamas ABAC modelio realizacijai. Pagrindiniai XACML komponentai [33]:

- *Rule* – šis elementas yra *Policy* tipo komponentų pagrindas ir privalo egzistuoti kartu su pastaruoju, kadangi savarankiškai negali būti perduodamas kitam aktoriui. Jis apibrėžia sąlygos išraišką, taigi yra įvertinamas, priklausomai nuo savo turinio.
- *Policy* – šis elementas naudojamas sprendimo priėmimo komponente ir yra sudarytas iš daugybės *Rule* elementų.
- *PolicySet* – šis elementas veikia kaip konteineris, kuriame talpinami kiti *Policy* arba *PolicySet* elementai.

Be pagrindinių XACML komponentų yra ir kitų svarbių komponentų:

- *Target* – šis elementas nusako pageidavimus subjektui, resursui bei veiksmams.
- *Effect* – šis elementas nusako taisyklės *Rule* kūrėjo norimą efektą, kai tikrinama sąlyga yra teisinga, t. y. „*True*“. Efektui nusakyti galimos „*Permit*“ ir „*Deny*“ reikšmės.
- *Condition* – šis elementas nusako sąlygos išraišką.
- *Attribute Designator* – šiame elemente nurodomas požymio, kurį norima įvertinti, pavadinimas.
- *Attribute Value* – šiame elemente nurodoma požymio reikšmė.

JACPol – tai JSON formatu pagrįsta lengvai plečiama bei paprasta politikos aprašymo kalba. Tam naudojama smulkmeniška bei hierarchiškai išdėliota struktūra, taip primindama XACML standartą. JACPol vartotoją verčia naudoti tradicinį ABAC modelį.

PTaCL (anlg. *Policy Target and Composition Language*) – tai ekspresyvi politikos taisyklėmis grįsta prieigos kontrolės aprašymo kalba. Naudojant šią aprašymo kalbą, sudaromos požymiais paremtos ABAC autorizacijos taisyklės.

XrML (angl. *eXtensible rights Markup Language*) – tai politikos taisyklėms aprašyti skirta kalba, paremta XML (angl. *eXtensible Markup Language*) standartu. Šia kalba apibūdinamos prieigos teisės bei būsenos.

Paminėtos kalbos yra pakankamai patogios ir ekspresyvios, tačiau tuo pačiu ir sudėtingos. Neretai paprastesniems atvejams naudojamas JSON formatas, kadangi yra lengvai suprantamas bei nedaug resursų reikalaujantis. Duomenų hierarchija pritaikoma pagal konkretų prieigos kontrolės mechanizmą.

1.4.5. Prieigos kontrolė daiktų interneto platformose

Daiktų interneto platformų yra įvairių: vienos yra bendro pobūdžio ir nepritaikytos konkrečiai sričiai, kitos – specializuotos ir sukurtos konkrečiai sričiai, pavyzdžiui išmaniųjų namų automatizacijai. Žemiau nagrinėjamas prieigos kontrolės valdymas populiariausiose daiktų interneto platformose.

IoT Hub – tai *Microsoft Azure* platformos dalis, sujungianti įrenginius, užtikrinanti jų priežiūrą ir valdymą. Ši platforma naudoja AAD (angl. *Azure Active Directory*) autentifikaciją ir palaiko RBAC prieigos valdymo modelį [34]. Vis dėlto, daugiausia atsižvelgiama tik į roles, neužtikrinamas daiktų interneto ekosistemai reikalingas konteksto ir požymių įvertinimas vertinant skirtingas situacijas.

Cloud IoT Core – tai *Google Cloud* platformos teikiama daiktų interneto paslauga, kurią naudojantys vartotojai gali sujungti įrenginius, surinkti ir apdoroti duomenis. Naudojanti prieigos kontrolės mechanizmais galima suteikti prieigą tik projekto lygmenyje [35]. Tam naudojamos rolės, taigi taikomas RBAC modelis.

AWS IoT Core – tai *AWS* (angl. *Amazon Web Services*) platformos daiktų interneto sprendimas, skirtas sujungti didelį kiekį daiktų interneto įrenginių ir centralizuotai juos valdyti. Ši platforma naudoja prieigos kontrolei skirtą IAM (angl. *Identity and Access Management*) AWS servisą, kurį naudojant realizuojamos politikos taisyklės [36]. Jos priskiriamos tam tikrai tapatybei (angl. *identity*) arba resursui. Tokiose taisyklėse įtraukiamos ir sąlygos, kurioms esant galima prieiga. Tačiau, kaip ir prieš tai minėtose platformose, toks prieigos valdymo metodas nepakankamai atsižvelgia į daiktų internetui būdingą aplinkos heterogeniškumą ir dinamiškumą.

Verta apžvelgti ir į specifinę sritį nutaikytas daiktų interneto platformas. Išmanieji namai – tai viena populiariausių ir nuolatos augančių daiktų interneto sričių, todėl galima nesunkiai surasti ir paanalizuoti tam skirtas platformas.

openHAB (angl. *Open Home Automation Bus*) – tai viena žinomiausių atviro kodo namų automatizacijos programinių įrangų. Tačiau analizuojant šioje platformoje taikomas saugos priemones, autorizacijai taikomas tik rolėmis grįstas metodas su vos keliomis rolėmis prisijungiant prie vartotojams skirtos grafinės sistemos [37]. Daugiau prieigos kontrolei dėmesio nėra skiriama.

Home Assistant – taip pat atviro kodo programinė priemonė, skirta valdyti išmaniuosius namų įrenginius. Kaip ir prieš tai nagrinėta platforma, šioje taip pat yra tik dvi rolės, skirtos prieigos kontrolei [38]. Taigi, autorizacijai skiriama dėmesio minimaliai.

Apibendrinus, aptartos platformos neturi efektyvių autorizacijos mechanizmų, pritaikytų daiktų interneto aplinkoms. Galima pastebėti, jog prieigos kontrolės problemų kyla tiek bendrinėse daiktų interneto platformose, tiek apibrėžtoje srityje veikiančiose platformose. Dažniausia problema ta, jog nėra atsižvelgiama į smulkesnius sistemų komponentus, vyraujančią konteksto informaciją, objektų, subjektų ar aplinkos požymius, naudojamos tik rolės.

1.5. Analizės dalies išvados

Analizės dalyje išnagrinėta, kaip veikia valdiklių valdymo platformos, kokie yra daiktų interneto ekosistemos veikimo principai bei kokios saugumo problemos joje egzistuoja. Atlikus probleminės srities analizę, galima padaryti kelias išvadas:

1. Nustatyta, jog valdiklių valdymo platformos veikia taikomajame daiktų interneto architektūriniame sluoksnyje. Jos rūpinasi įrenginių sujungimu į vieną visumą, duomenų saugojimu, apdorojimu ir vizualizavimu, taip pat įrenginių valdymu ir integravimu. Surinkti duomenys teikiami vartotojams bei kitiems servisams. Daiktų interneto ekosistemoje vyrauja nedaug skaičiavimo resursų ir energijos turintys įrenginiai, komunikuojantys bevieliais kanalais.
2. Analizuotos platformoms kylančios saugumo grėsmės, iš kurių svarbiausios: nesaugios sąsajos, silpna prieigos kontrolė ir autentifikacija, kriptografijos problemos, nesaugi komunikacija, nesaugūs komponentai, privatumo problemos, įrenginių valdymo trūkumas. Šiame darbe orientuojamasi į silpną ir netinkamą autorizaciją daiktų interneto platformose bei ieškomas metodas, kuris galėtų pagerinti prieigos kontrolės sprendimų priėmimą tokiuose sprendimuose.
3. Pastebėta, jog siekiant išspręsti prieigos kontrolės problemas daiktų interneto platformose, tradiciniai metodai nėra tinkami. Naudojant DAC metodą, resurso savininkui reikia nurodyti kiekvienam vartotojui taisykles, todėl yra sudėtingas tolesnis sistemos plečiamumas, o pats metodas yra statinis ir neatsižvelgia į dinamiką. MAC metodas taip pat nėra tinkamas, kadangi objektų savininkas nekontroliuoja prieigos, o pats metodas neatsižvelgia į detales. Pasirinkus RBAC metodą, reikia iš anksto apibrėžti, kokius veiksmus kiekviena rolė gali atlikti, tačiau didesnėje organizacijoje, stengiantis išlaikyti mažiausių privilegijų principą, rolių skaičius gali žymiai išaugti ir pasidaryti sudėtingas administravimas. Šis metodas taip pat yra statinis kaip ir pirmieji du. Analizuojant egzistuojančias platformas, pastebima, jog prieigos kontrolė įgyvendinama minimaliai ir priimant prieigos sprendimus nėra vertinamas aplinkos sąlygų kontekstas, objekto ar subjekto požymiai.
4. Darbo metu bus siekiama sukurti prieigos kontrolės metodą daiktų interneto platformai, veikiančiai žiniatinklio serveryje. Toks metodo panaudojimas administratoriui suteiks galimybę apibrėžti prieigos politikos taisykles, nusakančias subjektams leidžiamus veiksmus su valdikliais ar jų duomenimis. Metodo sprendimų priėmimo logika atsižvelgs tiek į vartotojų roles, tiek į statinius ar dinaminius požymius ir sąlygas, kurioje veiks sistema.

2. Prieigos kontrolės metodo valdiklių valdymo platformoms projektas

Projektui skirtame skyriuje aprašomi siekiami rezultatai kuriamam prieigos kontrolės metodui. Vėliau pagal tai sudaroma metodo vizija, aprašoma metodo taikymo sritis. Toliau detalizuojamas metodo sprendimo priėmimo procesas, rizikos skaičiavimo faktoriai, metodo naudojamos prieigos kontrolės taisyklės. Taip pat pateikiama detali metodo veiklos diagrama.

2.1. Siekiami rezultatai

Analizės dalyje nustatyta, jog daiktų interneto taikymo srityje vyrauja dinamiška bei nuolat kintanti aplinka su nedaug resursų turinčiais įrenginiais. Taip pat padaryta išvada, jog valdiklių valdymo platformos susiduria su autorizacijos saugumo problema, kuomet naudojami tradiciniai statiniai prieigos kontrolės metodai neatsižvelgia į daiktų interneto aplinkos dinamiškumą ir taip nesugeba užtikrinti mažiausių teisių principo. Taigi, šiame darbe sprendžiama prieigos kontrolės sprendimų priėmimo efektyvumo ir saugumo problema. Sudarant prieigos kontrolės metodą, siekiama išgauti šiuos rezultatus:

- **Detalumas.** Metodas turėtų užtikrinti galimybę detaliai įvertinti prieigos galimybes, atsižvelgdamas į subjektų, resursų ar aplinkos požymius. Galimybė vertinti skirtingus požymius neapsibrėžiant iš anksto nustatytais veiksmais užtikrina, kad metodas įvertins skirtingus prieigos aspektus ir taip leis užtikrinti mažiausių teisių principą. Detalumui įgyvendinti naudinga ABAC prieigos kontrolės metodams būdinga logika. Šio tipo metodai naudoja prieigos politikos taisykles, kuriomis remiantis yra priimami prieigos kontrolės sprendimai. Architektūrą paprastai sudaro esminės dalys: administravimo komponentas, politikos vykdymo komponentas, sprendimų priėmimo komponentas. Tokio tipo metodai leidžia detaliau nusakyti, kokius požymius turint arba kokioms aplinkos sąlygoms esant galimas prieigos suteikimas.
- **Lankstumas.** Metodas turėtų vertinti ne tik statinius subjektų ar objektų požymius, bet ir stebėti nuolatos kintančius aplinkos parametrus. Kintančių konteksto savybių vertinimas leidžia atsižvelgti į skirtingas ir kartais skubių veiksmų reikalaujančias prieigos situacijas, lanksčiai priimant prieigos kontrolės sprendimus. Lankstumu taip pat pasižymi ABAC tipo metodai, tačiau vien to valdiklių valdymo platformai nepakanka, kadangi reikalingas papildomas mechanizmas, kuris aktyviai stebėtų aplinkos konteksto sąlygas. Be to, sprendimų priėmimo procese reikalingos išimtys iš numatytų prieigos politikos taisyklių, kai susiduriama su rizikinga, neatidėliotina situacija. To pavyzdys galėtų būti grėsmė žmogaus gyvybei, tokiu atveju žmogaus saugumas turėtų būti svarbesnis negu informacijos saugumas.
- **Paprastumas**. Stengiantis užtikrinti metodo lankstumą ir detalumą, metodas taip pat turėtų išlaikyti kiek įmanoma paprastumą ir efektyvumą. Tai reiškia, jog konfigūravimas nėra sudėtingas, o plečiantis sistemai (pvz., atsiradus didesniam subjektų kiekiui), metodo prieigos taisyklių skaičius neturėtų reikalauti daugiau taisyklių ir administravimo kaštų negu to reikalauja statiniai prieigos metodai (pvz.: MAC, DAC, RBAC). Paprastumui didinti, pritaikomas hibridinis ABAC bei RBAC prieigos kontrolės metodų sprendimas. Tai reiškia, jog priimant prieigos sprendimą, vertinami aplinkos požymiai bei resurso požymiai, taip pat naudojamas rizikos balų skaičiavimas, įvertinant šiuos požymiu ir pagal juos išgaunant rizikingumo balą. RBAC logika išlaikoma, panaudojant administratoriaus subjektams priskirtas roles, pagal kurias apdorojamos sukurtos prieigos politikos taisyklės. Taip

bandoma išnaudoti abiejų privalumus – ABAC detalumą ir lankstumą bei RBAC metodo konfigūracijų paprastumą.

2.2. Prieigos kontrolės metodo vizija

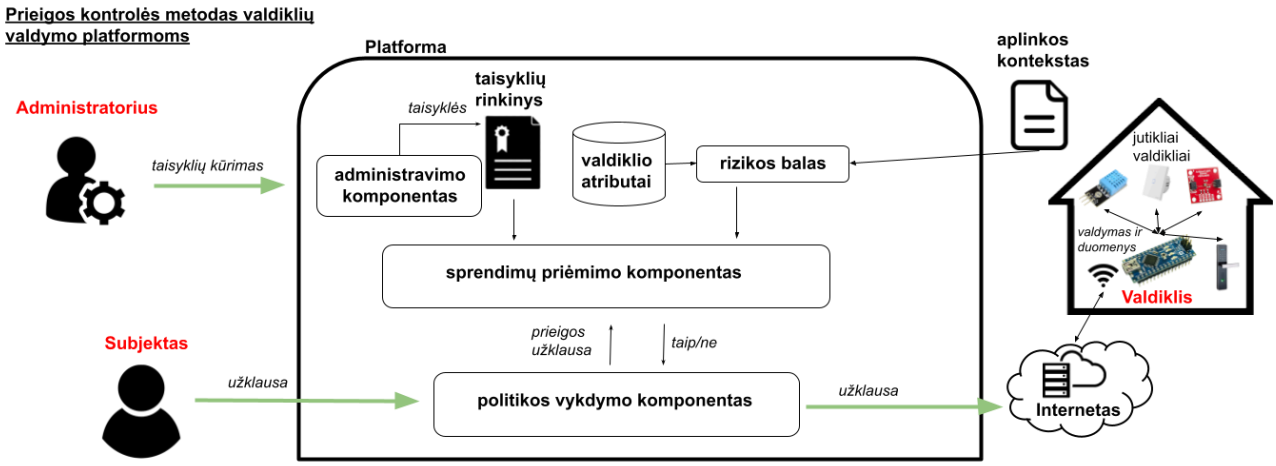
Ankstesniame poskyryje išsikelti siekiami rezultatai kuriamam prieigos kontrolės metodui. Nuspręsta, jog metodo detalumui bei lankstumui įgyvendinti, reikia remtis ABAC metodų tipui būdingais standartiniais architektūriniais blokais. Šiame poskyryje aprašoma metodo supaprastinta veikimo vizija bei aplinka, kurioje metodas veiktų.

Įprastai ABAC tipo metodų veikimo procese dalyvauja 3 pagrindiniai aktoriai: administratorius, subjektas ir objektas. Administratoriaus pagrindinis uždavinys – sukurti prieigos politikos taisykles, subjekto, arba vartotojo – teikti prieigos užklausas prieigos metodui, siekiant atlikti tam tikras operacijas su objektu. Valdiklis atlieka objekto, arba resurso, vaidmenį. Kadangi prieigos kontrolės metodas kuriamas valdiklių valdymo platformoms, tai tiek subjektas, tiek administratorius veiksmus atlieka naudodamiesi platformos teikiama sąsaja. Prieigos kontrolės metodas užklausas įvertina ir pateikia rezultatą: užklausos arba atmetamos, arba patvirtinamos.

Platformoje veikiančio prieigos kontrolės metodo blokų tarpusavio veikimą galima išskaidyti į kelis esminius etapus:

1. Administratorius prideda rizikomis grįstas prieigos politikos taisykles į platformos saugomų taisyklių rinkinį. Visa tai atliekama naudojantis *administravimo komponentu*.
2. Subjektas siunčia užklausą į valdiklių valdymų platforma atlikti tam tikrą operaciją su valdikliu. Ši užklausa nukeliauja į prieigos kontrolės metodą.
3. Gauta užklausa perduodama *politikos vykdymo komponentui*. Šis atlieka pirminį užklausos apdorojimą ir siunčia ją į *sprendimų priėmimo komponentą*.
4. *Sprendimų priėmimo komponentas* gautą užklausą apdoroja remdamasis atitinkama rizikomis grįsta prieigos politikos taisykle iš taisyklių rinkinio. Taisyklė apdorojama, atsižvelgiant į taisyklėje nurodytus sprendimo priėmimo veiksnius, t. y. nurodytą rizikingumo balą. Pastarasis apskaičiuotas remiantis valdiklio požymiais bei aplinkos konteksto sąlygomis. Pagal turimas taisykles *sprendimų priėmimo komponentas* priima galutinį sprendimą ir siunčia atsakymą atgal politikos vykdymo komponentui leisti užklausą arba ją atmesti.
5. *Politikos vykdymo komponentas* gautą atsakymą naudoja tolimesniems veiksmas, t. y. subjektui grąžinamas pranešimas apie draudžiamą veiksmą arba užklausa praleidžiama toliau į sistemą atlikti norimą operaciją su valdikliu ar jo duomenimis.

Aprašyti prieigos kontrolės metodo komponentai ir bendras metodo veikimas vizualizuotas 5 pav.



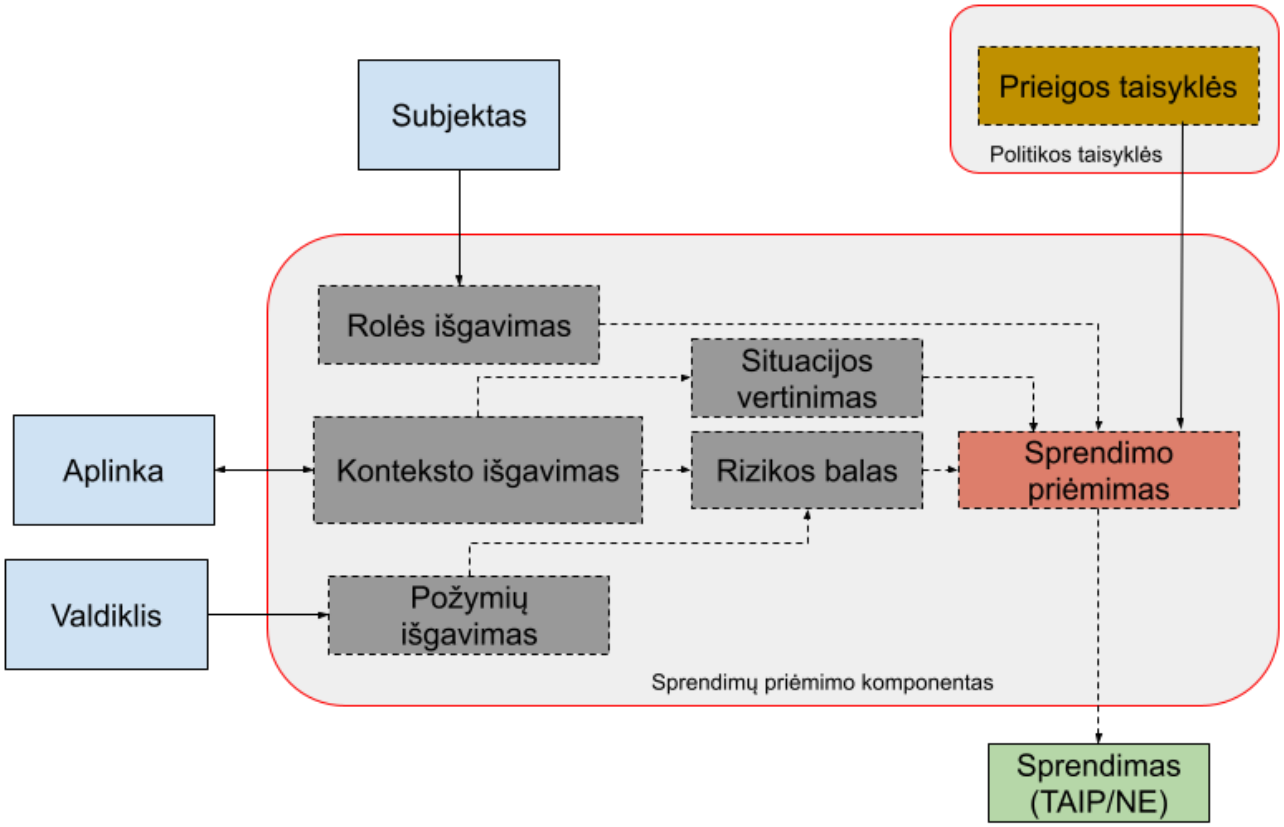
5 pav. Prieigos kontrolės metodo valdiklių valdymo platformoms vizija

Taigi, gavus atsakymą iš prieigos kontrolės metodo, užklausa gali būti įgyvendinama arba atmetama. Platformos realizacija ir valdiklių tinklas komunikuoja nutolusiu būdu, t. y. per internetą. Metodo taikomojoje aplinkoje gali veikti įvairių tipų įvesties bei išvesties įrenginiai (pvz.: valdikliai, kontroliuojantys apšvietimo, durų atidarymo įrenginius, taip pat surenkantys duomenis iš drėgmės, temperatūros, šviesos, artumo bei kitų jutiklių). Valdiklis valdomas iš platformos, o duomenys persiunčiami į platformą naudojantis valdiklio turimais komunikacijos moduliais, kuriais pasiekiamas tinklas. Apibendrinus, aptarta, kaip metodas veikia valdiklių valdymo platformoje bei kokia yra platformą supanti aplinka.

2.3. Sprendimų priėmimo komponento principinis veikimas

Praeitame poskyryje aptarta apibendrinta metodo veikimo architektūra ir aplinka. Vis dėlto, svarbiausias metodo elementas – tai sprendimų priėmimo komponentas, kuriame realizuojama sprendimų priėmimo logika. Vadovaujantis šia logika, sprendžiama ar subjekto išsiųsta užklausa galima, ar ne. Žemiau aptariamos sprendimo priėmimo komponento dalys, dalyvaujančios prieigos sprendimų priėmimo procese.

Metodas naudoja rizikomis grįstas politikos taisykles. Jų vertinimas primena ABAC metodų tipui būdingą politikos taisyklių apdorojimą, kai priimant sprendimą, atsižvelgiama į skirtingus subjekto, objekto ar aplinkos požymius. Kuriamam metodui naudojamos tiek vartotojams priskirtos aplinkos rolės, tiek prieigos politikos taisyklės, nusakančios subjekto rolei leidžiamą maksimalų rizikingumo balą. Vienas esminių sprendimo priėmimo etapų – vartotojo aplinkos rolės išgavimas, arba nuskaitymas, vėliau naudojamas apdorojant politikos taisykles. Šias subjektų roles sistemos administratorius priskiria sistemos konfigūracijos metu. Taip pat priimant sprendimus reikalingi aplinkos požymiai, kitaip sakant, konteksto sąlygos, bei resurso požymiai. Šie požymi reikalingi įvertinant situaciją bei apskaičiuojant rizikingumo balą. Taigi, šioje vietoje bandoma suderinti RBAC bei ABAC prieigos kontrolės metodų privalumus, taip realizuojant kiek įmanoma paprastesnį, lankstesnį bei detalesnį metodą. Pavaizduotas procesas matomas 6 pav.



6 pav. Prieigos kontrolės metodo principinis veikimas

Sprendimo priėmimo procese svarbus aplinkai būdingo dinamiškai kintančio konteksto įvertinimas. Tam kuriamas prieigos kontrolės metodas naudoja aplinkos būsenų išgavimą. Pastarosios priklauso nuo administratoriaus pridėtų konteksto sąlygų, kurias stebi valdiklių valdymo platforma. Aplinkos būsenos pavyzdžiu galėtų būti įvairių jutiklių duomenys apie aplinką. Taigi, toks vertinimas leidžia užtikrinti, jog bus atsižvelgta į skirtingas situacijas.

Vis dėlto 6 pav. matoma, jog aplinkos būsenos nėra tiesiogiai vertinamos priimant sprendimą, tam naudojamas tiek rizikos balas, tiek situacijos vertinimas. Rizikos balas apskaičiuojamas įvertinant aplinkos kontekstą bei objekto požymius. Tam naudojamas nuskaitytas valdiklio jautrumas bei norimos atlikti operacijos kritiškumas. Rizikos vertinimas leidžia paprastai įvertinti skirtingus požymius ir pateikti bendrą rizikingumo balą. Detalus rizikos apskaičiavimo procesas aprašomas kitame poskyryje.

Apibendrinus, prieigos kontrolės sprendimų priėmimas vykdomas pagal sudarytas prieigos politikos taisykles kiekvienai subjektų rolei. Nurodytas maksimalus leistinas rizikingumo balas rolei lyginamas su apskaičiuotu rizikingumu subjekto užklausai, atsižvelgiant į aplinkos bei resurso požymius.

2.4. Rizikos balo skaičiavimas

Rizikos balui apskaičiuoti reikia apsibrėžti, kurie rizikos faktoriai bus naudojami skaičiavimams. Kuriamo metodo atveju naudojami šie požymiai:

- **Resurso jautrumas.** Šis požymis apibūdina resurso, arba valdiklio, prie kurio bando prieiti subjektas, jautrumo lygį. Kuo didesnis jautrumo lygis, tuo blogesnės tikėtinos pasekmės gali įvykti neautorizuotos arba netinkamai autorizuotos operacijos atveju. Taigi, esant didesniam resurso jautrumui, bus didesnis ir bendras rizikos balas.
- **Veiksmo poveikis.** Skirtingi veiksmai, arba operacijos, gali turėti skirtingas pasekmes, todėl kiekvienai operacijai priskiriamas tam tikras poveikis, nusakantis neigiamas pasekmes. Kuo didesnis poveikis, tuos didesnis bendras rizikos balas.
- **Užklausos konteksto rizikingumas**. Tam tikros užklausos aplinkos konteksto sąlygos (pvz., užklausa iš tam tikro IP adreso ar tam tikro įrenginio tipo) gali turėti neigiamų pasekmių, todėl kelia riziką. Šiai rizikai įvertinti, aplinkos konteksto sąlygoms pagal sistemos konfigūracijas gali būti priskiriamas aukštesnis rizikingumo balas.

Nustačius esminius veiksnius, sudarančius galutinį rizikos įvertinimą, svarbu pasirinkti ir rizikos apskaičiavimo metodą. Tokių metodų yra įvairių [39]: neraiški logika (angl. *fuzzy logic*), mašininis mokymasis, žaidimo teorija, standartinis rizikos įvertinimo metodas ar kitos matematinės lygtys. Dėl savo paprastumo kuriamam prieigos kontrolės metodui naudojamas standartinis rizikos apskaičiavimas:

$$Standartinis\ rizikos\ įvertis = Tikimybė \times Poveikis.$$

Kuriamo metodo atveju tikimybę, jog įvyks neigiamos pasekmės, nusako subjekto ketinamo atlikti veiksmo poveikis. Tai reiškia, jog skirtinga operacija su resursu (valdikliu) gali turėti skirtingą tikimybę, jog įvyks neigiamos pasekmės. Tuo tarpu neigiamą poveikį, arba pasekmes, įvykus įvykiui, nusako resurso ir jo duomenų jautrumas. Taigi, bendruoju metodo atveju bazinis rizikos įvertinimas apskaičiuojamas naudojantis šia formule:

$$Bazinis\ metodos\ rizikos\ įvertis = Veiksmo\ poveikis \times Resurso\ jautrumas.$$

Taip pat svarbu įvertinti užklausos konteksto rizikingumą, kadangi esant skirtingam konteksto rizikingumui, metodas turėtų į tai atsižvelgti. Aplinkos konteksto sąlygoms įvertinti, standartinėje rizikos apskaičiavimo formulėje įvedamas papildomas kintamasis, nusakantis konteksto rizikingumą, todėl apibendrinta metodo rizikos įvertinimo formulė skaičiuojama taip:

$$Rizikos\ balas = Bazinis\ rizikos\ įvertis \times Konteksto\ rizikingumas.$$

Aplinkos rizikingumą gali sudaryti keletas aplinkos konteksto sąlygų. Taigi, bendras aplinkos sąlygų rizikingumas yra suma visų galimų aplinkos konteksto sąlygų padalinus iš tokių sąlygų skaičiaus:

$$Konteksto\ rizikingumas = \frac{\sum_i Konteksto\ sąlygos\ \boldsymbol{i}\ rizikingumas}{Konteksto\ sąlygų\ skaičius}.$$

Pateiktas rizikos balo skaičiavimas atliekamas tik įprastomis aplinkos sąlygų situacijomis. Aplinkos sąlygų įvertinimus galima suskaidyti į dvi slenkstines ribas:

- **Kritinė situacija.** Tai reiškia, jog esant šiai situacijai sprendimo priėmimo komponentas nekreipia dėmesio į apskaičiuotą bazinį rizikos įvertį. Kritinė situacija apibrėžia atvejus, kai susiduriama su neatidėliotina situacija, kai žmogaus sveikata, jo turtas tampa svarbesnis, negu saugoma informacija. Tokiu atveju metodas geba į tai atsižvelgti ir sprendimų priėmimo mechanizme nėra vertinamas rizikingumo balas, bet iškart subjektui suteikiama prieiga.
- **Įprasta situacija**. Tokia situacija yra įprasta ir stabili, todėl nekelia jokios grėsmės subjektui. Informacijos saugumas šiuo atveju yra esminis, todėl jokių išimčių rizikos atžvilgiu prieigos kontrolės metodas nedaro.

Šioms situacijoms įvertinti, reikia nustatyti vertinamiems aplinkos požymiams slenkstines ribas, kurias peržengus, situacija tampa kritinė. Tą atlieka valdiklių valdymo platformos administratorius, konfigūruodamas prieigos metodą. Apibendrinus, aptarta sprendimų priėmimo logiką leidžia atsižvelgti į aplinkos sąlygų kritiškumą, taip užtikrinant prieigos kontrolės metodo lankstumo savybę.

Toliau analizuojama, kaip bus vertinamas kiekvienas rizikos veiksnys. Veiksmai – tai operacijos, kurias gali atlikti subjektas su konkrečiu resursu (valdikliu bei prie jo prijungtais jutikliais ar vykdikliais). Kiekvienam galimam veiksmui su valdikliu administratorius turi priskirti veiksmo poveikio įvertinimą. Tai yra įvertinimas, kokias pasekmes informacijos konfidencialumo, integralumo bei prieinamumo atžvilgiu turėtų operacija, jeigu ji būtų panaudota nekorektiškai. Toks įvertinimas gali turėti šias neigiamo poveikio reikšmes: mažas poveikis, vidutiniškas poveikis, aukštas poveikis. Žemiau pateikiamos paminėtų poveikių skaitinės reikšmės, kurios bus naudojamos skaičiuojant rizikos balą:

- mažas poveikis – 1;
- vidutiniškas poveikis – 2;
- didelis poveikis – 3.

Resurso jautrumas suprantamas kaip poveikio įvertinimas, jei resursas bus panaudotas piktavališkais tikslais. Jautrumas gali turėti atitinkamas reikšmes: nejautrus, jautrus, labai jautrus. Didelis jautrumo lygis reiškia, jog resursas turi ypač svarbios informacijos, kuri turėtų būti griežtai saugoma, tuo tarpu žemas jautrumo lygis rodo, kad resursas ir su juo asocijuojami duomenys nėra tokie svarbūs. Jautrumo reikšmės turi skaitinius atitikmenis:

- nejautrus resursas – 1;
- jautrus resursas – 2;
- labai jautrus resursas – 3.

Užklausos konteksto sąlygų rizikingumas nusako, ar aplinka yra pavojinga. Tam įvertinti, naudojamos dvi reikšmės: didelis rizikingumas ir normalus rizikingas. Pastarosioms suteikiamos atitinkamos skaitinės reikšmės:

- normalus rizikingumas – 1;
- didelis rizikingumas – 2.

Sudėjus visas konteksto sąlygų rizikingumo reikšmes ir padalinus iš jų skaičiaus, gaunama bendra reikšmė tarp 1 ir 2. Mažesnis įvertis reiškia mažą rizikingumą, didesnis – didelį. Taigi, kuo didesnis konteksto rizikingumas, tuo jis turės didesnį poveikį galutiniam rizikos balo dydžiui, t. y. rizikingumui.

Galutinis rizikos balas apskaičiuojamas naudojant anksčiau minėtas skaitines reikšmes ir pasitelkiant žemiau pateiktą rizikos skaičiavimo formulę:

$$Rizikos\ balas = (Veiksmo\ poveikis\ \times Resurso\ jautrumas) \times Konteksto\ rizikingumas.$$

Gautas rizikos balas yra tiesiogiai naudojamas sprendimo priėmimo procese lyginant jį su rizikos politikos taisyklėse nurodytu maksimaliu rizikingumo balu. Rizikos reikšmių intervalas suskaidomas į 5 rizikos lygius:

- nuo 0 iki 3.6 – tai reiškia, jog rizikingumas yra labai mažas, t. y. nereikšmingas;
- nuo 3.6 iki 7.2 – tai žemo rizikingumo reikšmės;
- nuo 7.2 iki 10.8 – tai vidutinio rizikingumo reikšmės;
- nuo 10.8 iki 14.4 – tai aukšto rizikingumo reikšmės;
- nuo 14.4 iki 18 – tai reiškia, jog informacijos saugumo prasme rizikingumas yra labai didelis, t. y. kritinis;

Atsižvelgiant į paminėtus intervalus ir jų paaiškinimus, valdiklių valdymo platformos administratorius sukuria prieigos politikos taisykles, nurodydamas vartotojų roles ir joms priskiriamus maksimalius rizikingumo balus.

2.5. Politikos taisyklių interpretavimas

Ankstesniuose poskyriuose trumpai aptarta, kaip veikia valdiklių valdymo platformos prieigos kontrolės metodas, kokie duomenys naudojami sprendimo priėmimo procese. Vis dėlto, kuriamo metodo veikimui užtikrinti, svarbu apibrėžti, kaip sudaromos subjekto rolėms skirtos prieigos politikos taisyklės.

Prieigos politikos taisyklės sudaromos prieigos kontrolės metodo administravimo komponente ir saugomos duomenų bazėje JSON duomenų formatu. Šis formatas pasirinktas, kadangi yra paprastas naudojant JavaScript aplikacijose, lengvai suprantamas bei nedaug resursų užimantis. Taisyklės sudarytos iš 3 esminių laukelių:

- *_id* – tai identifikacinis numeris, skirtas unikaliai identifikuoti prieigos politikos taisykles duomenų bazėje;
- *envRole* – tai vartotojams priskiriama rolė pagal jų aplinkoje atliekamą vaidmenį;
- *riskScore* – tai maksimalus leistinas rizikingumo balas.

Taigi, apdorojant subjektų užklausas, pirmiausia nustatoma užklausą pateikusio vartotojo aplinkos rolė. Tuomet duomenų bazėje surandama taisyklė, atitinkanti vartotojo rolę. Surastos taisyklės maksimalus rizikingumo balas lyginamas su apskaičiuotu subjekto užklausos rizikingumo balu ir pagal tai grąžinamas atsakymas.

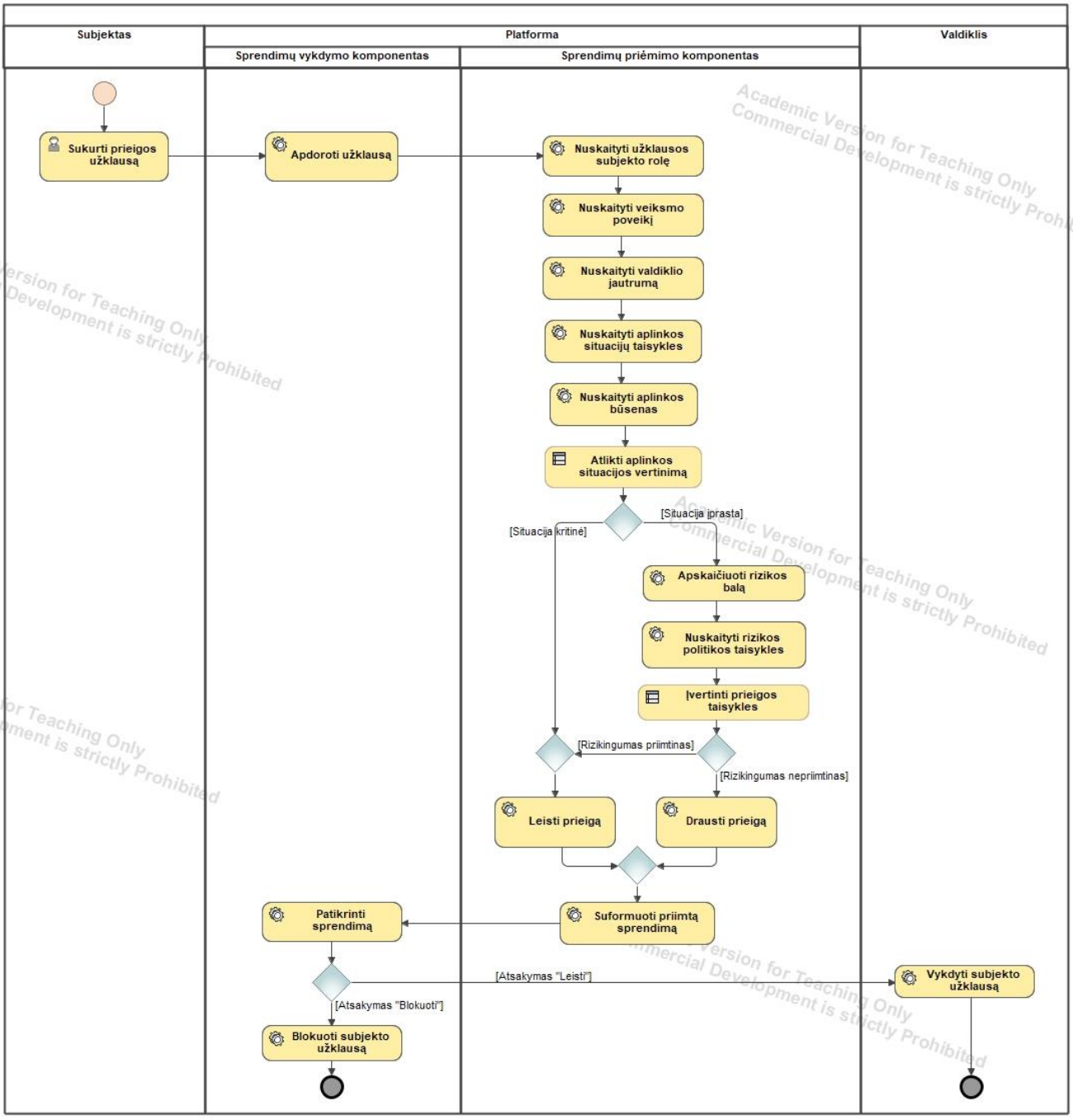
2.6. Detalus prieigos metodo sprendimo priėmimo procesas

Šiame poskyryje pateikiamas detalus sudaromo prieigos kontrolės metodo sprendimo priėmimo procesas. Vis dėlto, tinkamam metodo veikimui būtinas pradinis administratoriaus įsikišimas paruošiant sistemą.

Visų pirma, metodo administravimo komponente būtina turėti sukurtas prieigos politikos taisykles, nusakančias kiekvienai vartotojų aplinkos rolei leistiną maksimalų rizikos balą. Šiomis taisyklėmis vadovaujamasi sprendimo priėmimo procese. Būtinas rolės priskyrimas subjektui, taip pat subjekto,

resurso, operacijų ir aplinkos būsenų numatymas tam, kad visa tai būtų galima vertinti, priimant prieigos kontrolės sprendimus. Reikalingas išankstinis resurso jautrumo, veiksmo poveikio priskyrimas bei aplinkos konteksto sąlygų pridėjimas. Apsibrėžus visus šiuos duomenis, galimas sklandus metodo veikimas.

Esminė prieigos užklausų apdorojimo proceso dalis – tai sprendimų priėmimo komponentas, kuriame priimamas galutinis prieigos kontrolės sprendimas. Detalus proceso veikimas matomas 7 pav.



7 pav. Prieigos kontrolės metodo veiklos diagrama

Prieigos kontrolės metodas pradeda veikti gavus prieigos užklausą iš subjekto. Pirminį šios užklausos apdorojimą atlieka sprendimų vykdymo komponentas. Ši metodo dalis atlieka minimalų

užklausos apdorojimą, duomenų nuskaitymą ir toliau visa tai perduoda sprendimų priėmimo komponentui.

Sprendimų priėmimo komponentas nuskaito subjekto rolę. Taip pat nuskaitomas užklausoje ketinamos atlikti operacijos poveikis, objekto (resurso), prie kurio subjektas ketina gauti prieigą, jautrumas. Prieigos kontrolės metodas nuskaito aplinkos konteksto sąlygų taisykles bei reikšmes, pasiruošdamas situacijos įvertinimui.

Atlikus pradinius žingsnius, sprendimo priėmimo komponentas vykdo vertinimą ir nustato ar esanti aplinkos situacija yra įprasta, ar kritinė. Galimi du variantai.

- Jeigu situacija yra kritinė, metodas, neatsižvelgdamas į rizikingumą, suteikia subjektui prieigą, kadangi vertinama, jog žmonių bei turto saugumas yra svarbesnis už informacijos saugumą.
- Jeigu situacija yra įprasta, metodas toliau skaičiuoja pateiktos užklausos rizikos balą. Balas vertina resurso jautrumą, operacijos poveikį bei įvertina užklausos konteksto rizikingumą.

Kontekstas gali nusakyti užklausos laiką, įrenginio tipą, IP adresą ar kitą faktorių, kuris vertinant bendrą rizikingumą yra labai svarbus, kadangi padeda įvertinti aplinką. Atlikus bendro rizikingumo balo apskaičiavimą, nuskaitomos rizikos politikos taisyklės pagal nustatytą subjekto rolę ir palyginama, ar apskaičiuotas balas yra mažesnis negu rizikos politikoje nurodytas balas. Jeigu mažesnis – tuomet traktuojama, jog rizikingumas yra tinkamas ir suteikiama prieiga, jeigu didesnis – rizikingumas netinkamas, todėl prieiga draudžiama.

Toliau suformuotas atsakymas perduodamas atgal į sprendimų vykdymo komponentą, kuris patikrina priimtą sprendimą ir, priklausomai nuo to ar užklausa leidžiama, ar draudžiama, perduoda subjekto užklausą įvykdyti arba blokuoja užklausos vykdymą ir išveda informacinį pranešimą apie negalimą atlikti užklausą.

2.7. Projekto dalies išvados

Šiame skyriuje pateikta valdiklių valdymo platformoms skirto prieigos kontrolės metodo vizija, sudarytas projektas. Pagal tai galima padaryti kelias esmines išvadas:

1. Kuriamas prieigos kontrolės metodas orientuojamas spręsti prieigos kontrolės sprendimų priėmimo problematiką. Stengiamasi užtikrinti mažiausių teisių užtikrinimo principą dinamiškoje daiktų interneto platformoje. Iškelti rezultatai – metodo detalumas, lankstumas ir paprastumas. Siekiant detalumo numatyta galimybė įvertinti skirtingus požymius neapsibrėžiant vien statiniais nustatymais. Tikimasi, kad metodas gebės įvertini įvairius prieigos aspektus ir taip užtikrins mažiausių teisių principą. Lankstumas – tai galimybė reaguoti į besikeičiančias situacijas, vertinti dinaminius požymius adaptuojant prieigos kontrolės sprendimus pagal konkrečias situacijas. Palaikant detalumą ir lankstumą, metodas turėtų užtikrinti ir paprastumą, t. y. konfigūravimams neturėtų būti sudėtingas, o prieigos politikos taisyklių skaičius plečiant sistemą neturėtų reikalauti daugiau taisyklių ir administravimo kaštų negu to reikalauja tradiciniai metodai.
2. Išsikeltiems rezultatas pasiekti pasitelkta ABAC tipo metodų architektūra bei veikimo principai. Metodui naudojamas rizikos vertinimas su aplinkos dinaminių savybių ir situacijų vertinimo mechanizmu. Taigi, sudarytas hibridinis ABAC ir RBAC tipų metodas su rizikomis grįstomis politikos taisyklėmis išlaikant tiek roles, tiek požymius.

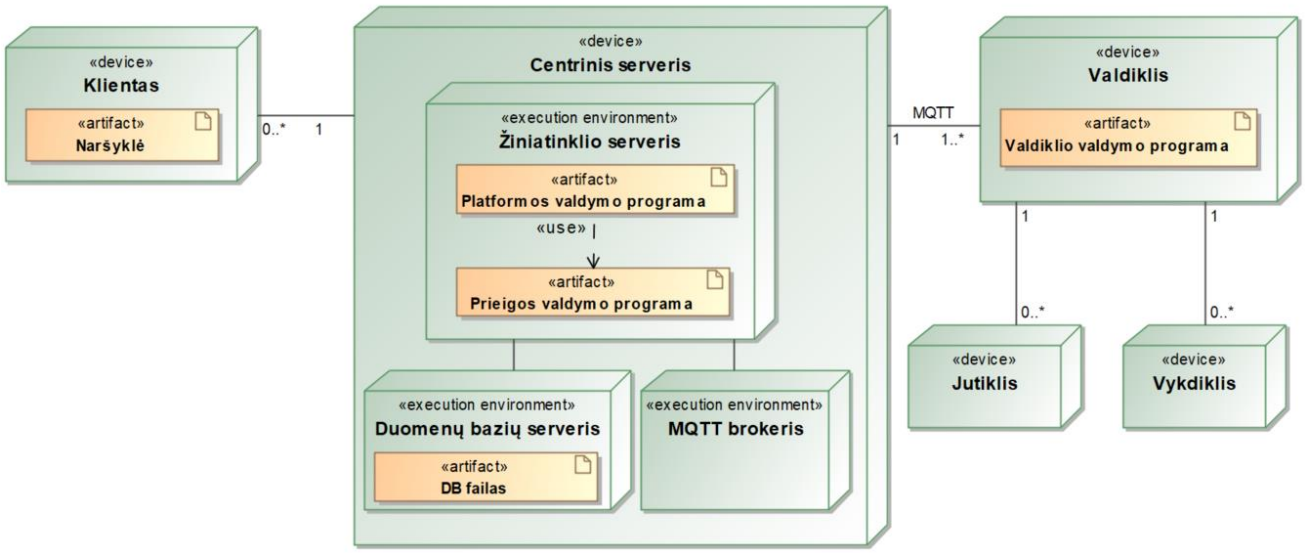
3. Apibendrinus, valdiklių valdymo platformoms skirtas prieigos kontrolės metodas turėtų būti detalesnis bei lankstesnis už tradicinius statinius prieigos metodus (pvz.: DAC, MAC, RBAC). Tikimasi, jog jis palaikys paprastesnį plečiamumą lyginant su anksčiau minėtais metodais. Taip pat sudarytas metodas turėtų pasižymėti konfigūravimo paprastumu ir efektyvumu bei nenusileisti kitiems ABAC tipo metodams.

3. Prieigos kontrolės metodo valdiklių valdymo platformoms realizacija

Realizacijai skirtame skyriuje aprašomi prototipo architektūriniai komponentai, detalizuojama prototipui naudojama programinė bei techninė įranga.

3.1. Sistemos diegimo diagrama

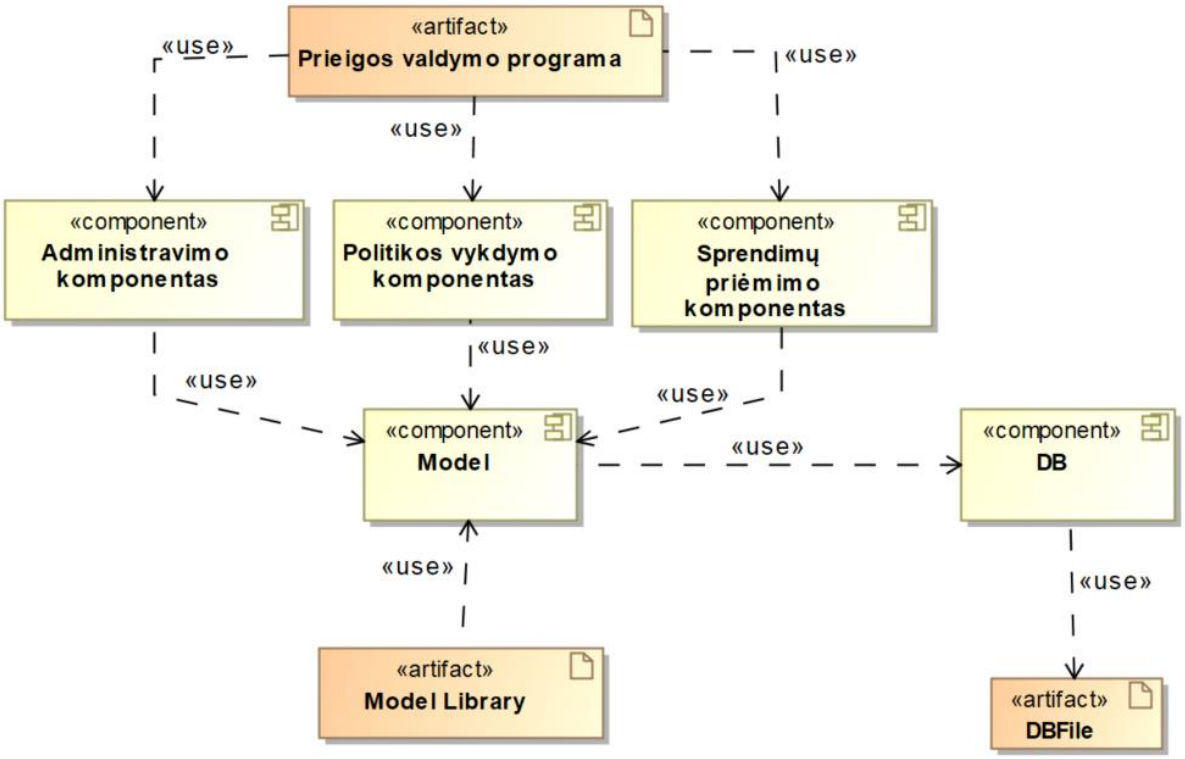
Prieigos kontrolės metodo prototipo sudėtis vaizduojama 8 pav. Sprendimo architektūra susideda iš trijų esminių dalių, t. y. iš kliento įrenginio, centrinio serverio bei valdiklio.



8 pav. Valdiklių valdymo platformos diegimo diagrama

Klientą atspindi bet kuris valdiklių valdymo platformos naudotojas, įskaitant tiek platformos administratorių, tiek subjektą. Naudojant asmeniniame įrenginyje veikiančias naršykles, pasiekiamas viešame tinkle veikiantis centrinis serveris, kuriame žiniatinklio serveris užtikrina platformos vartotojo sąsajos atvaizdavimą, taip pat jame vykdomos visos operacijos.

Žiniatinklio serveryje veikia pagrindinė programa, skirta valdyti platformą, tačiau ji naudoja pagalbinę prieigos valdymo programą, skirtą užtikrinti platformos valdiklių valdymo saugumą. Pastarojoje realizuotas esminis šio darbo sprendimas – prieigos kontrolės metodas valdiklių valdymo platformai. Prieigos valdymo programą sudaro 9 pav. matomi komponentai.



9 pav. Valdiklių valdymo platformos komponentų diagrama

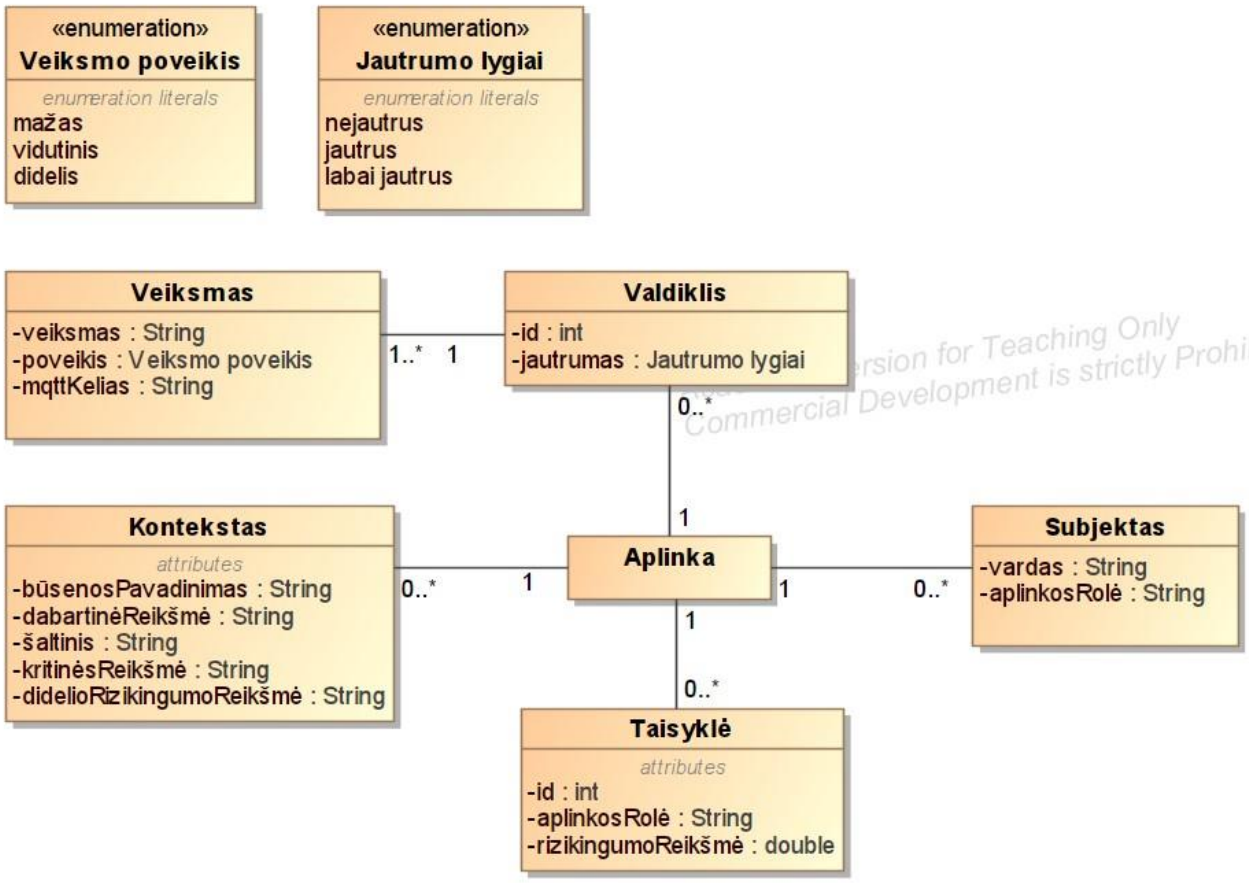
Administravimo komponentas skirtas metodo administracinėms užduotims realizuoti, t. y. pridėti naujas prieigos valdymo taisykles. Politikos vykdymo komponente realizuojama komunikacinė sąsaja tarp subjekto bei valdiklių valdymo platformos prieigos valdymo metodo. Sprendimų priėmimo komponente remiantis saugomomis taisyklėmis nusprendžiama, ar subjekto užklausą galima patvirtinti. Tam naudojamos duomenų bazėje saugomos taisyklės bei kiti reikalingi duomenys.

Žiniatinklio serveryje sukurtiems duomenims išsaugoti pasitelkiama duomenų bazių serveryje laikoma duomenų bazė. Centrinis serveris komunikuoja su valdikliais naudojant MQTT komunikavimo protokolą. Šiame serveryje veikia MQTT brokeris, užtikrinantis komunikaciją šiuo protokolu.

Valdikliai veikia nuotolyje, t. y. naudojantis belaidžio tinklo Wi-Fi protokolu bendraujama su centriniu serveriu. Valdiklio valdymo programa skirta valdyti valdiklį, nuskaityti duomenis iš jutiklių, kontroliuoti vykdiklius bei užtikrinti komunikaciją tinkle. Priklausomai nuo naudojamų išvesties ir įvesties įrenginių, valdiklis gali komunikuoti tiek analoginiu signalu, tiek I2C, UART ar kitomis duomenų magistralėmis, todėl 8 pav. tai nėra akcentuojama.

3.2. Prieigos kontrolės metodo duomenų modelis

Efektyviam prieigos kontrolės metodo veikimui užtikrinti, reikia saugoti tam tikrus duomenis apie subjektą, valdiklį bei aplinką. Prieigos kontrolės metodui reikalingas duomenų modelis pateikiamas 10 pav.



10 pav. Prieigos valdymo metodo valdiklių valdymo platformoms duomenų modelis

Reikėtų pastebėti, jog tam tikrą valdiklių infrastruktūros aplinką gali sudaryti keli subjektai ir keli valdikliai. Aplinka taip pat gali turėti ją apibūdinančias konteksto sąlygas bei reikalingas prieigos kontrolės politikos taisykles.

Subjekto duomenų lentelėje saugoma informacija tik apie subjektą unikaliai identifikuojantį vardą bei subjektui priskirtą aplinkos rolę, t. y. vaidmenį platformos taikymo srityje. Naudojant šią rolę identifikuojama subjektui taikytina prieigos politikos taisyklė.

Valdiklio duomenų lentelėje pateikiama tik valdiklį identifikuojanti informacija bei informacija, nusakanti jo jautrumą pagal anksčiau nagrinėtus jautrumo lygius. Valdiklis gali turėti kelis veiksmus, kuriuos galima atlikti su valdikliu. Kiekvienas tokių veiksmų turi administratoriaus priskirtą poveikio įvertinimą bei MQTT komunikacijai naudojamą kelią, skirtą valdiklio veiksmui inicijuoti.

Konteksto duomenų lentelėje saugoma informacija apie konteksto būsenos pavadinimą, dabartinę aplinkos konteksto reikšmę, konteksto duomenų šaltinį, iš kurio gaunamos reikšmės. Taip pat saugomos kritinės konteksto reikšmės, kurioms esant situacija yra vertinama kaip kritinė, bei didelio rizikingumo reikšmės, kurioms esant skiriamas didelio rizikingumo balas.

Taisyklių lentelėje nurodomas taisyklės identifikacinis numeris bei saugoma aplinkos rolė, pagal kurią bus taikomos taisyklės subjekto užklausoms. Taip pat šioje lentelėje apibrėžiama maksimalaus subjektui leistino rizikingumo reikšmė.

Sistemoje, priklausomai nuo saugos administratoriaus atliktų konfigūracijų, gali būti naudojami duomenys apie užklausos atlikimo laiką, vartotojo IP adresą ar lokaciją. Ši informacija duomenų bazėje nėra saugoma, kadangi paminėti duomenys dinamiškai kinta, priklausomai nuo sesijos.

3.3. Prototipo techninė įranga

Realizuojant prieigos kontrolės metodo prototipą, naudojama ši techninė įranga:
1. „Raspberry Pi 4 model B“ mikrokompiuteris.
2. „ESP8266 NodeMCU“ mikrovaldiklis.
3. Jutikliai ir išvesties įrenginiai.

Parinkta techninė įranga būdinga įprastoms daiktų interneto infrastruktūroms. Tokiu pasirinkimu stengiamasi priartinti valdiklių valdymo platformos ir metodo veikimą prie realios daiktų interneto infrastruktūros.

3.3.1. Mikrokompiuterio „Raspberry Pi 4 model B“ specifikacija

„Raspberry Pi 4 model B“ mikrokompiuteris, dar vadinamas mini kompiuteriu, yra nedidelis SBC (angl. *single-board computer*) tipo kompiuteris [40]. Toks įrenginys puikiai tinka daiktų interneto projektams ir tyrimams. Įrenginys turi 64 bitų keturių branduolių ARM Cortex-A72 procesorių. Mikrokompiuteris palaiko 802.11 ac Wi-Fi standartą, taip pat gali perduoti duomenis Ethernet jungtimi. Konkretus pasirinktas modelis turi 4 GB RAM atmintį bei 32 GB SD atminties kortelę. Toks sprendimas yra tinkamas daiktų interneto taikymams namų aplinkoje, kadangi gali atstoti nedidelių matmenų, bet pakankamai galingą serverį, kuriame veiktų vartotojui reikiama programinė įranga.

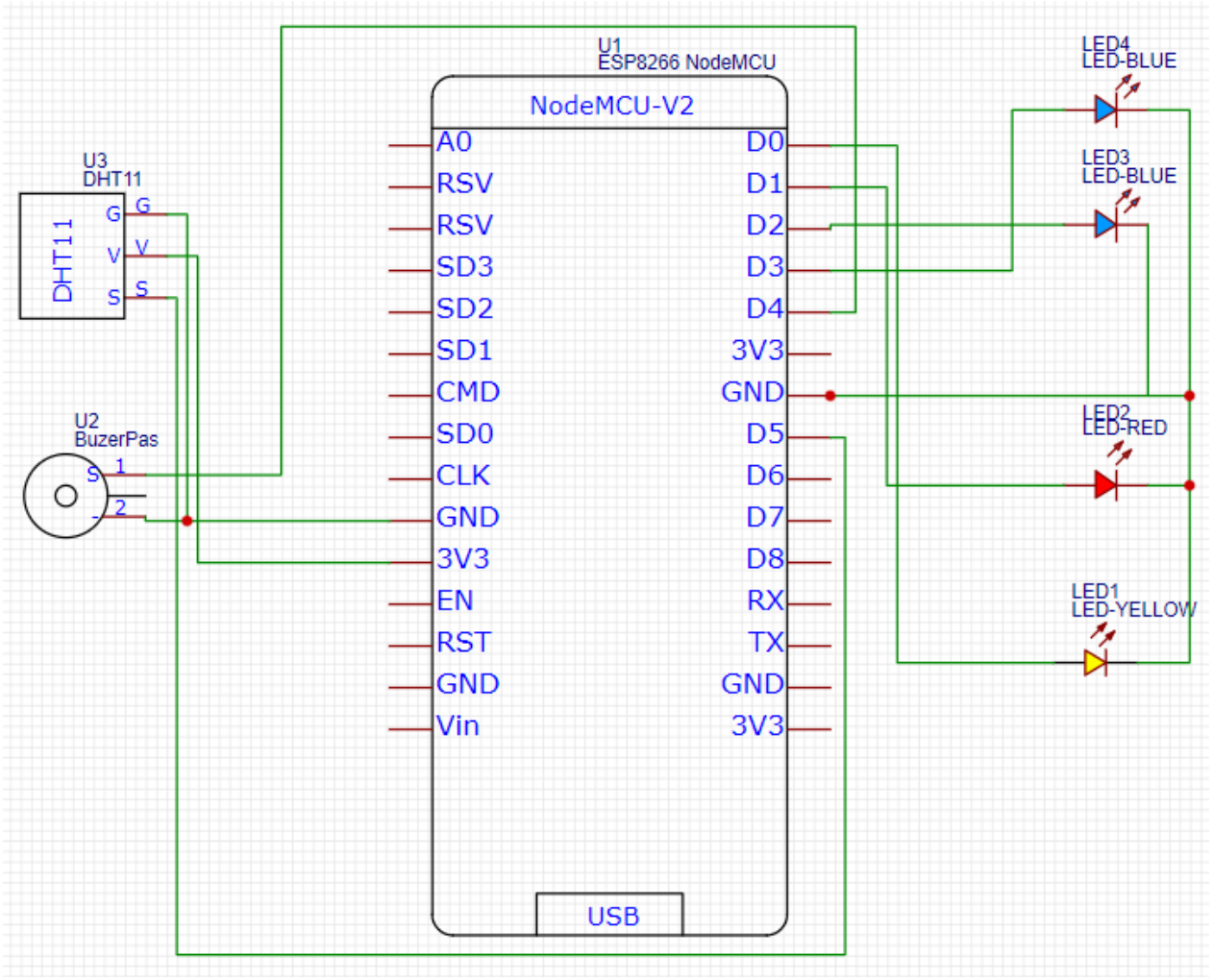
3.3.2. Mikrovaldiklio „ESP8266 NodeMCU“ specifikacija

Projekto prototipo realizacijai naudojamas mikrovaldiklis „ESP8266 NodeMCU V2“. Įrenginys specialiai pritaikytas daiktų interneto realizacijoms, yra pigus, pasižymi mažu energijos suvartojimu ir turi integruotą Wi-Fi komunikacijos modulį, todėl yra dažnai naudojamas projektų ir tyrimų. Šis valdiklis turi 32 bitų 80 MHz procesorių, 4 MB atmintinę (angl. *flash memory*), 64 kB SRAM atmintį. Wi-Fi modulis suderinamas su 802.11 b/g/n standartais ir palaiko iki 72.2 Mbps duomenų perdavimo greitį. ESP8266 turi SPI, UART, I2C jungtis, 11 skaitmeninių įvesčių bei 1 analoginę.

3.3.3. Jutikliai ir išvesties įrenginiai

Siekiant atlikti išsamius ir iliustratyvius sukurtos valdiklių valdymo platformos testavimo atvejus, reikalingi papildomi mikrovaldiklio valdomi įvesties bei išvesties įrenginiai, kuriuos naudojant bus imituojami ir ištestuojami platformoje leistini veiksmai. Taigi, tam tikslui prie mikrovaldiklio „ESP8266 NodeMCU“ prijungti šie įrenginiai:
- mėlynos LED šviesel ės (2 vnt.);
- raudona LED šviesel ė;
- geltona LED šviesel ė;
- DHT11 temperatūros ir drėgmės jutiklis;
- pasyvus švilpukas (angl. *buzzer*).

11 pav. pateikiama projekto mikrovaldiklio įrenginių principinė schema.



11 pav. Mikrovaldiklio įrenginių principinė schema

Pagal aukščiau pateiktą schemą matoma, jog geltonos, raudonos ir mėlynos LED šviesėlių anodų dalys yra atitinkamai sujungtos su mikrovaldiklio jungtimis D0, D1, D2, D3. Katodų dalys sujungtus su mikrovaldiklio GND jungtimi. Pasyvaus švilpuko neigiama jungtis sujungtas su mikrovaldiklio GND, o teigiama – su D4. DHT11 jutiklio neigiama dalis sujungta su mikrovaldiklio GND, VCC – su 3.3 V išvestimi, o duomenų jungtis – su D5.

3.4. Prototipo programinė įranga

Realizuojant prieigos kontrolės metodą, naudojama programinė įranga tiek valdiklyje, tiek mikrokompiuteryje.

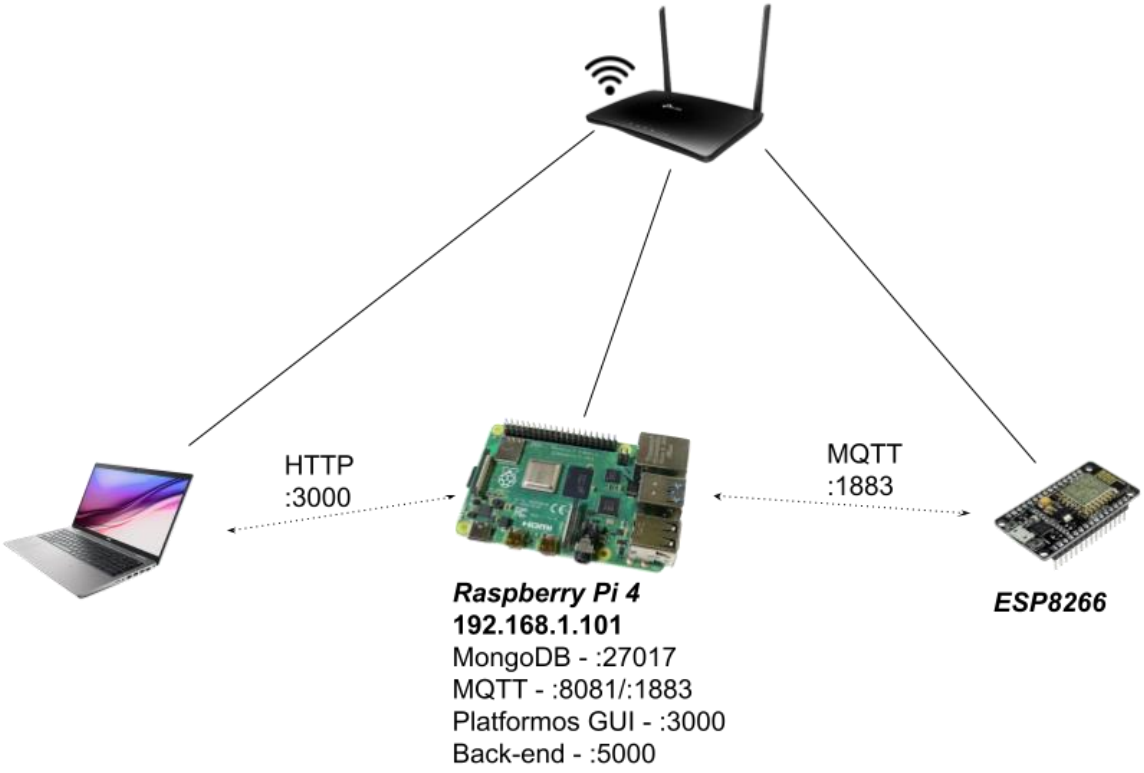
Mikrokompiuteryje veikianti programinė įranga:

- „Raspberry Pi OS Lite“ 32-bit operacinė sistema.
- „Node.js v15.7.0“ vykdymo aplinka. Ji skirta valdiklių valdymo platformos veikimui užtikrinti. Platformos grafinė sąsaja veikia HTTP protokolu 3000 prievadu, vidinė platformos dalis – 5000 prievadu. Komunikacijai tarp skirtingų platformos dalių naudojamas JSON duomenų apsikeitimo formatas.

– „MongoDB“ duomenų bazių valdymo sistema. Ji skirta platformos prieigos kontrolės metodo duomenims ir politikos taisyklėms saugoti. Duomenų bazė pasiekiama 27017 prievadu.
– „Mosquitto MQTT v3.1.1.1“ brokeris. Jis atlieka tarpininko vaidmenį komunikuojant MQTT protokolu. Komunikacija su valdikliu užtikrinama per 1883 prievadą, o sąsaja su žiniatinklio programa vykdoma per 8081 prievadą. Prisijungiant prie brokerio naudojama autentifikacija.
– „MQTT.js“ biblioteka. Ji skirta užtikrinti komunikacijai tarp valdiklių valdymo platformos ir valdiklio MQTT protokolu.

Mikrokompiuteryje veikianti programinė įranga:
– „ESP8266WiFi“ biblioteka. Ji skirta užtikrinti komunikacijai su serveriu Wi-Fi protokolu.
– „PubSubClient“ biblioteka. Ji skirta užtikrinti komunikacijai su serveriu MQTT protokolu.

12 pav. matoma diagrama, apibendrinanti išvardintą prototipo programinę, taip pat ir techninę įrangą.



12 pav. Aparatūrinės posistemės schema

Tokia programinė įranga dažnai naudojama daiktų interneto sprendimuose bei valdiklių valdymo platformų realizacijose. Ji užtikrina greitą, paprastą ir efektyvią komunikaciją su valdikliais.

3.5. Valdiklių valdymo platforma

Siekiant patikrinti, kaip veikia prieigos kontrolės metodas, realizuotas valdiklių valdymo platformos prototipas uždarame tinkle. Klientas platformą gali pasiekti adresu http://192.168.1.101/:3000. Grafinė sąsaja sistemos administratoriaus rolę turinčiam vartotojui leidžia atlikti tokioms

platformoms įprastus veiksmus: pridėti naujus vartotojus, valdiklius ir galimus atlikti veiksmus su šiais valdikliais. Vartotojas, atlikęs pradines konfigūracijas, gali nuotoliu naudodamasis šia platforma valdyti įrenginius, gauti iš jų informaciją.

Platformos grafinė sąsaja bendrauja su vidine platformos dalimis 5000 prievadu. Vidinėje dalyje realizuota visa platformos logika. Valdiklių funkcijos inicijuojamos išsiuntus MQTT pranešimą tema *homeDeviceControl*. Pranešimas siunčiamas su atitinkama žinute nurodant MQTT funkcijos kelią pagal anksčiau administratoriaus atliktas valdiklių konfigūracijas. Atsakymai gaunami MQTT tema *homeDeviceControlAnswers*.

Siekiant užtikrinti valdiklių valdymo platformos saugumą ir prieigos kontrolės metodo veikimą, sistemoje pridėta galimybė administratoriui pasirinkti valdiklių jautrumą bei veiksmų su tais valdikliais poveikį.

Taip pat administratorius turi galimybę pridėti konteksto sąlygas ir joms nurodyti didelio rizikingumo ar kritinės situacijos ribines reikšmes. Konteksto sąlygų stebėjimas vykdomas MQTT protokolu, temoje *homeDeviceContext* pagal pridėtus funkcijų kelius. Gauti duomenys fiksuojami duomenų bazėje. Taip pat konteksto sąlygos gali būti tikrinamos pagal sistemoje realizuotas funkcijas. Realizuotos 3 tokios funkcijos:

- *Daytime*. Ši funkcija tikrina, ar užklausa daroma dienos metu. Nakties metu atliktos užklausos laikomos rizikingesnėmis.
- *Network*. Ši funkcija tikrina, ar subjekto IP adresas yra vidinio tinklo. Iš išorinio tinklo atkeliavusi užklausa laikoma rizikingesne.
- *Location*. Ši funkcija tikrina, ar subjekto sukurta užklausa atkeliauja iš Lietuvos. Jeigu užklausa atkeliauja iš kitur, laikoma, kad tai kelią riziką, todėl rizikingumo balas bus didesnis.

Administratorius konfigūruoja prieigos kontrolės logiką, pridėdamas prieigos politikos taisykles *politikos administravimo komponente*.

3.6. Prieigos kontrolės metodo veikimas prototipe

Atlikus platformos saugos konfigūracijas, prieigos kontrolės metodas gali tinkamai apdoroti subjektų užklausas. Subjektas sprendimų vykdymo komponentui pateikia 13 pav. matomus JSON formato duomenis.

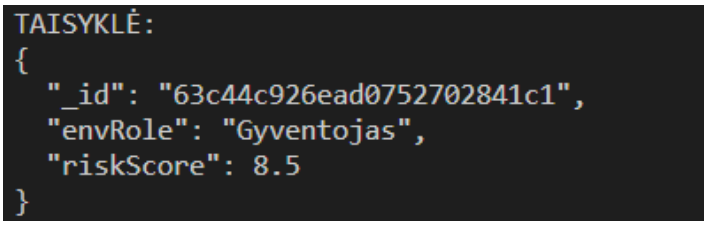
```
{
  "device": 100002,
  "mqttpath": "/fireplace/on"
}
```

13 pav. Užklausos į sprendimų vykdymo komponentą pavyzdys

Užklausoje pateikiamas valdiklio identifikacinis numeris bei MQTT funkcijos kelias, kurį gavęs valdiklis, vėliau atliks tam tikrus veiksmus. Pirmiausia, *sprendimų vykdymo komponentas* nustato užklausą pateikusio subjekto aplinkos rolę, ištraukia užklausos antraštes, kurios bus naudojamos

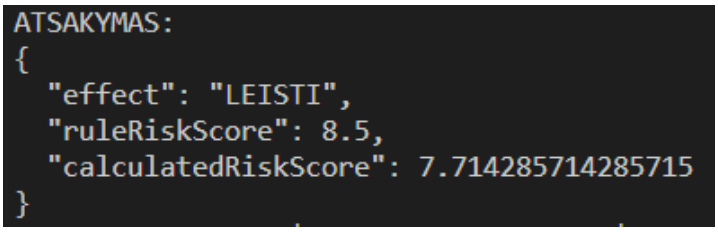
kaip kontekstinė informacija. Gauta informacija siunčiama į *sprendimų priėmimo komponentą*, papildomai pateikiant ir valdiklio jautrumą bei veiksmo poveikio lygį.

Toliau sprendimų priėmimo komponente įvertinama situacija ir, jeigu ji nėra kritinė, atliekamas rizikos balo apskaičiavimas apskaičiuojant riziką pagal projektavimo dalyje pateiktą rizikos skaičiavimo formulę. Komponente apdorojama taisyklė, kurios pavyzdys matomas 14 pav.



14 pav. Apdorojamos prieigos politikos taisyklės pavyzdys

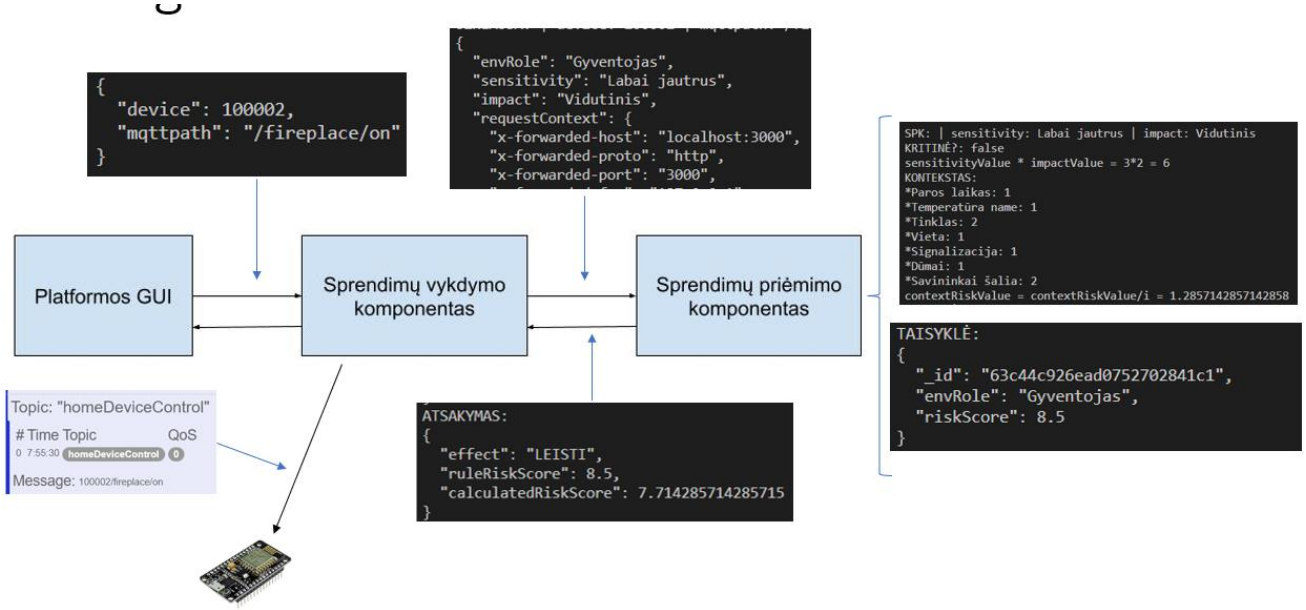
Toliau suformuojamas atsakymas sprendimų vykdymo komponentui. Pavyzdys matomas 15 pav.



15 pav. Sprendimų priėmimo komponento atsakymo pavyzdys

Matoma, jog pagal atliktą rizikos skaičiavimą ir nuskaitytą taisyklę, atliekamas palyginimas ir grąžinamas atsakymas *effect* atgal į *sprendimų vykdymo komponentą*.

Apibendrintą loginę prieigos užklausų apdorojimo seką matome 16 pav.



16 pav. Prieigos užklausų apdorojimo seka

Sprendimų vykdymo komponentas, gavęs atsakymą iš *sprendimų priėmimo komponento*, atitinkamai blokuoja užklausą arba išsiunčia MQTT žinutę valdikliui subjekto pasirinktiems veiksmams įgyvendinti.

3.7. Prieigos kontrolės metodo prototipo realizacijos išvados

Realizacijos skyriuje pateikta valdiklių valdymo platformoms skirto prieigos kontrolės metodo įgyvendinimo detalizacija. Iš viso to galima padaryti šias išvadas:

1. Prototipo architektūrą sudaro platformos klientas, centrinis serveris su MQTT brokeriu bei duomenų baze, taip pat valdikliai. Komunikacija tarp serverio ir valdiklių vykdoma MQTT protokolu. Serveryje veikia valdiklių valdymo platforma, turinti prieigos valdymo programą, sudarytą iš administravimo komponento, sprendimų vykdymo komponento bei sprendimų priėmimo komponento. Pastarajame veikia esminis algoritmas, priimantis prieigos sprendimus pagal sudarytas prieigos politikos taisykles.
2. Realizuojant prototipą, pasirinktas daiktų interneto taikymuose dažnai naudojamas „Raspberry Pi 4 model B“ mikrokompiuteris su „Raspberry Pi OS Lite“ operacine sistema. Jame veikia „Node.js“ vykdymo aplinka, „MongoDB“ duomenų bazių valdymo sistema, „Mosquitto“ MQTT brokeris. Taip pat prototipe naudojamas „ESP8266 NodeMCU“ mikrovaldiklis. Abu įrenginiai sujungti viename Wi-Fi tinkle. Mikrovaldiklis Wi-Fi komunikacijai užtikrinti naudoja „ESP8266WiFi“ biblioteką. Siekiant palaikyti MQTT protokolą, naudojama „PubSubClient“ biblioteka, serverio pusėje – „MQTT.js“ biblioteka.
3. Valdiklių valdymo platformos grafinė sąsaja pasiekiama 3000 prievadu, vidinė dalis veikia 5000 prievadu. Duomenys tarp platformos komponentų siunčiami JSON formatu. Valdiklis su MQTT brokeriu bendrauja 1883 prievadu, žiniatinklio programa – 8081 prievadu. Platforma su duomenų baze bendrauja 27017 prievadu.
4. Pagal administratoriaus atliktas pradines konfigūracijas platformoje nustatomas valdiklių jautrumas, veiksmų su jais poveikio lygis, taip pat konteksto sąlygos ir jų kritiškumas ar rizikingumas. Subjekto užklausos pirmiausia gaunamos į sprendimų vykdymo komponentą. Toliau užklausa pagal administratoriaus sukurtas JSON formato rizikomis grįstas politikos taisykles apdorojama sprendimų vykdymo komponente. Jame lyginamas apskaičiuotas bendras rizikingumo balas su maksimaliu leistinu rizikingumo balu vartotojo rolei pagal nustatytą politikos taisyklėje. Gautas atsakymas pateikiamas JSON formatu atgal į sprendimų vykdymo komponentą, kuris, priklausomai nuo sprendimo, užklausą užblokuoja arba išsiunčia reikiamą MQTT žinutę valdikliui.

4. Prieigos kontrolės metodo valdiklių valdymo platformoms tyrimas ir rezultatai

Šiame skyriuje atliekamas realizuoto prototipo tyrimas. Pirmiausia, detalizuojami tiriami parametrai, toliau aprašomi valdiklių valdymo platformos aplinkos paruošimo žingsniai. Skyriaus gale pateikiami tyrimo scenarijų metu gauti rezultatai bei visa tai apibendrinančios išvados.

4.1. Parametrai

Siekiant atlikti realizuoto metodo kokybinį ir kiekybinį eksperimentą, apibrėžiami esminiai parametrai, skirti įvertinti tyrimo scenarijus. Žemiau pateikiami metodo efektyvumą padėsiantys įvertinti parametrai:

- **Administravimo kaštai.** Šis parametras leis įvertinti metodo sudėtingumą ir politikų skaičių.
- **Greitaveika**. Šis parametras leis įvertinti metodo sprendimų priėmimo laiką.
- **Saugumas**. Šis parametras skirtas palyginti tikėtiną sprendimą su realiai metodo priimtu sprendimu.

Taigi, atliekant tyrimo scenarijus bus fiksuojamos išvardintos parametrų reikšmės kaip eksperimento rezultatai.

4.2. Aplinkos paruošimas

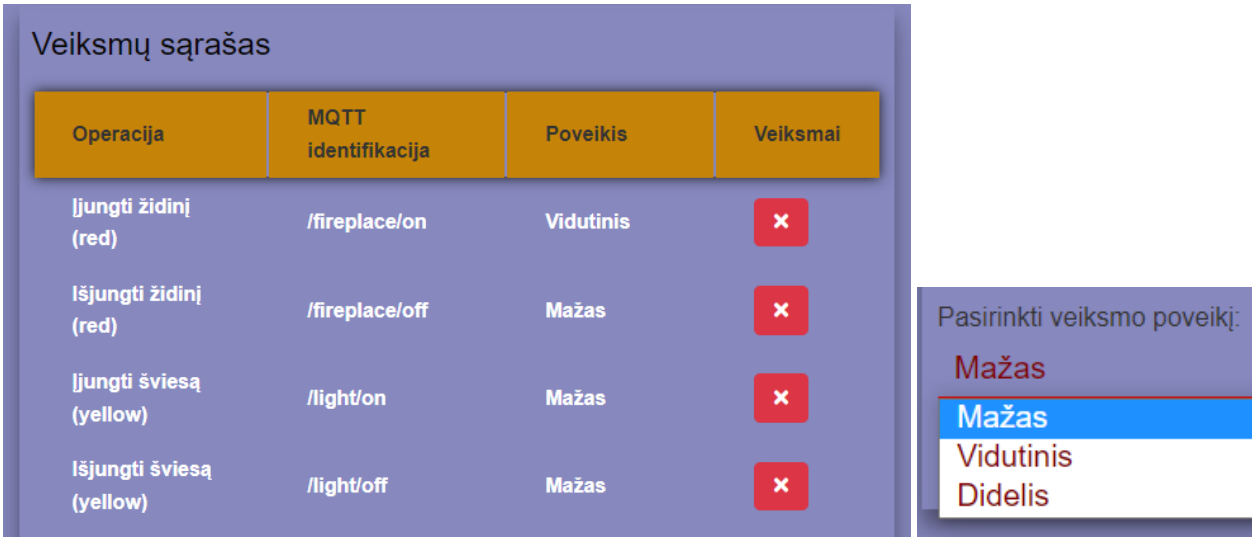
Siekiant lengviau atlikti tyrimą bei geriau suprasti jo metu gautus rezultatus, sukurta valdiklių valdymo platforma sukonfigūruota taip, kad atitiktų išmaniųjų namų aplinką.

Visų pirma, platformoje pridedami vartotojai, kuriems suteikiamos šiai aplinkai būdingos rolės: savininkas, gyventojas, vaikas, auklė, svečias, nuomininkas. 17 pav. pateikiamas tokios konfigūracijos pavyzdys.

17 pav. Vartotojų konfigūracijos pavyzdys

Toliau sukonfigūruojami valdikliai, pasirenkamas jų jautrumas, taip pat pridedami veiksmai, kuriuos galima atlikti su šiais įrenginiais, bei tų veiksmų poveikio lygis. Pavyzdžiai matomi 18 ir 19 pav.

18 pav. Valdiklių konfigūracijos pavyzdys



19 pav. Veiksmų konfigūracijos pavyzdys

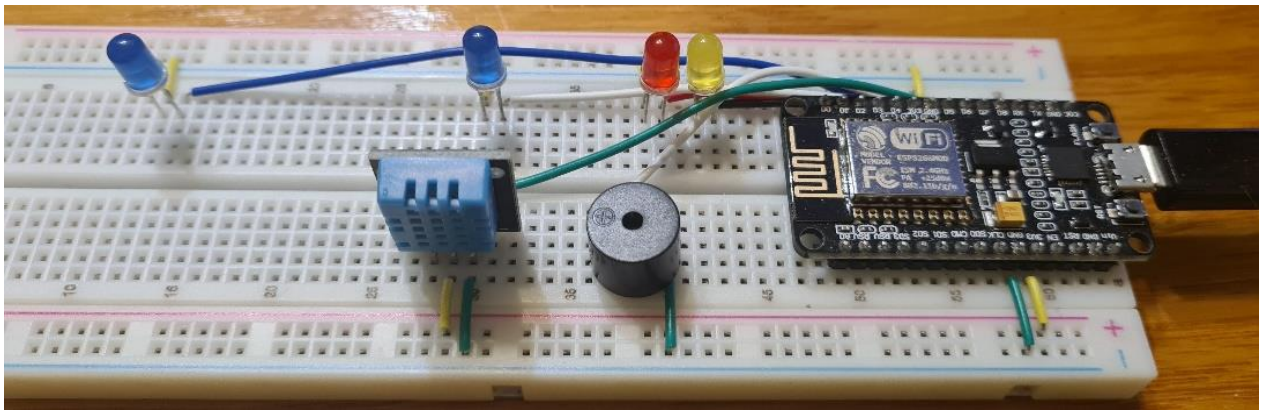
Pagal pateiktus pavyzdžius matoma, jog bus imituojami 3 valdikliai namų aplinkoje. Žemiau pateikiami galimi veiksmai su šiais valdikliais:

- **Valdiklis ID 100001 (garažo imitacija). Jautrumas – vidutinis, kadangi garaže gali būti brangių įrenginių.**
  - *Pakelti vartus*. Šį veiksmą imituoja garsą skleidžiantis švilpukas. Poveikis – didelis, kadangi suteikia patekimą į garažą.
  - *Nuleisti vartus*. Šį veiksmą imituoja garsą skleidžiantis švilpukas. Poveikis – vidutinis, kadangi gali sužaloti žmogų.
  - *Gauti statusą*. Šis veiksmas leidžia sužinoti garažo vartų statusą. Poveikis – mažas, kadangi pateikia neesminę informaciją.
- **Valdiklis ID 100002 (namų imitacija). Jautrumas – didelis, kadangi name saugomi visi dokumentai, brangūs daiktai, gyvena žmonės.**

- *Įjungti židinį*. Šį veiksmą imituoja įsijungusi raudona LED šviesel. Poveikis – vidutinis, kadangi neatsargiai elgiantis gali sukelti gaisrą.
- *Išjungti židinį*. Šį veiksmą imituoja išsijungusi raudona LED šviesel. Poveikis – mažas, kadangi užgesinant židinį nekyla kritinių pasekmių.
- *Įjungti šviesą*. Šį veiksmą imituoja įsijungusi geltona LED šviesel. Poveikis – mažas, kadangi šviesos įjungimas nesukelia esminių pasekmių.
- *Išjungti šviesą*. Šį veiksmą imituoja išsijungusi geltona LED šviesel. Poveikis – mažas, kadangi šviesos išjungimas nesukelia esminių pasekmių.
- *Įjungti signalizaciją*. Šį veiksmą imituoja įsijungusi mėlyna LED šviesel. Poveikis – mažas, kadangi namų signalizacijos įjungimas nesukelia esminių pasekmių.
- *Išjungti signalizaciją*. Šį veiksmą imituoja išsijungusi mėlyna LED šviesel. Poveikis – didelis, kadangi signalizacijos išjungimas gali sukelti esminių pasekmių leidžiant įsibrauti į namus.

– **Valdiklis ID 100003 (sodo imitacija). Jautrumas – mažas, kadangi sode nėra brangių įrenginių.**
  - *Paleisti/stabdyti laistytuvą*. Šį veiksmą imituoja be perstojo mirksinti mėlyna LED šviesel. Poveikis – vidutinis, kadangi esminių pasekmių šis veiksmas nesukelia.
  - *Gauti statusą*. Šis veiksmas leidžia sužinoti, ar laistytuvas veikia. Poveikis – mažas, kadangi gaunama nekritinė informacija apie sodo laistytuvo statusą.
  - *Gauti tinklo nustatymus*. Šis veiksmas leidžia gauti valdiklio tinklo nustatymus. Poveikis – didelis, kadangi gaunami esminiai valdiklio ir viso sodo įrenginių nustatymai ir konfigūracijos.

20 pav. matomas ESP8266 valdiklis, imituojantis visą išmaniųjų namų įrenginių aplinką, kuri bus naudojama platformos bei prieigos metodo tyrimui:



20 pav. ESP8266 su imitaciniais įrenginiais

Atliekama namų aplinkos kontekstinių sąlygų konfigūracija nustatant jų šaltinius, didelio rizikingumo reikšmes bei kritines reikšmes. Pavyzdys pateikiamas 21 pav.

21 pav. Konteksto konfigūracijos pavyzdys

Pagal pateiktą konteksto konfigūracijų sąrašą galima matyti, jog yra pridėtos skirtingos aplinkos sąlygos, būdingos valdiklių aplinkai. Visas šias sąlygas galima priskirti prie kritinių arba įprastų aplinkos sąlygų. Pateiktame pavyzdyje pastebima, jog situacija bus vertinama kaip kritinė, jeigu namuose bus aptikti dūmai, t. y. bus keliama potenciali grėsmė žmogaus gyvybei bei jo turtui. Užfiksavus dūmus, kitos aplinkos sąlygos nebus vertinamos, kadangi prieigos metodas šiuo atveju iškart suteiks prieigą subjektui.

Taip pat matoma, kokios įprastos situacijos aplinkos sąlygos yra sukonfigūruotos. Jos turi nustatytas didelio rizikingumo reikšmes, kurioms esant rizikingumo balas padidėja. Pagal pateiktą pavyzdį pastebima, jog nakties laikotarpis vertinamas kaip didelio rizikingumo aplinką apibūdinanti sąlyga, taip pat aukšta temperatūra, gaunama iš jutiklio DHT11 temperatūros bei drėgmės jutiklio. Subjektų užklausos iš išorinio tinklo arba lokacijos, kuri nėra Lietuvoje, laikomos turinčios didelį rizikingumą. Rizikingumas yra didelis ir tuo atveju, jeigu namų signalizacija suveikia arba tuo atveju, jeigu namo šeimininkų nėra namuose.

Sukurtos maksimalų rizikingumo balą nustatančios prieigos politikos taisyklės kiekvienai aplinkos rolei. Pavyzdys pateikiamas 22 pav.

22 pav. Taisyklių konfigūracijos pavyzdys

Žemiau išdėstyti maksimalūs leistinų rizikingumo reikšmių lygiai kiekvienai rolei:

- Svečias – NEREIKŠMINGAS;
- Vaikas – ŽEMAS;
- Auklė – ŽEMAS;
- Gyventojas – VIDUTINIS;
- Nuomininkas – AUKŠTAS;
- Savininkas – KRITINIS.

4.3. Tyrimo scenarijai

Tyrimo scenarijai atliekami atskirai, kiekvienam tyrimo parametrui įvertinti.

4.3.1. Administravimo kaštai

Visų pirma, atliekamas prieigos kontrolės metodo administravimo kaštų tyrimas. Tam skaičiuojamas prieigos politikų skaičius. Atliekami keli scenarijai:

- **Bandymas Nr. 1** – šiuo atveju vartotojų, objektų, veiksmų ir konteksto sąlygų yra tiek, kiek buvo apibrėžta aplinkos paruošimo metu. Sakykim, jog kiekvienas objektas turi vidutiniškai po 4 veiksmus.
- **Bandymas Nr. 2** – šiuo atveju vartotojų skaičius padaugėja 100%.
- **Bandymas Nr. 3** – šiuo atveju objektų padaugėja 100%.

Tyrimo rezultatams geriau suvokti, atliekamas kiekybinis sudaryto metodo palyginimas su kitais prieigos kontrolės metodais. Gauti administravimo kaštai, t. y. prieigos politikų skaičius, yra lyginami su kitais prieigos metodais: MAC, DAC bei RBAC tipo metodais.

Kuriamo metodo politikų skaičius priklauso nuo rolių, konteksto sąlygų, objektų bei veiksmų ir yra visų jų sandaugos atitikmuo. Jeigu metodas visiškai sukonfigūruotas, tokiu atveju gali daugėti tik

objektų, vartotojų skaičiaus didėjimas tam poveikio neturės dėl naudojamų rolių. Jeigu 100% padidinamas objektų skaičių, tiek pat kartų padaugėja ir politikų skaičius. 2 lent. pateikiami darbe sukurto metodo tyrimo rezultatai.

2 lentelė. Valdiklių valdymo platformos prieigos kontrolės metodo administravimo kaštų tyrimas

	Bandymas Nr. 1	**Bandymas Nr. 2**	**Bandymas Nr. 3**
Rolės	6	6	6
Konteksto sąlygos	7	7	7
Objektai	3	3	6
Veiksmai vienam objektui	4	4	4
Politikų skaičius:	504	504	1008

MAC metodo politikų skaičius priklauso nuo vartotojų, žymų bei objektų skaičiaus. Politikų skaičius atitinka vartotojų ir žymų bei žymų ir objektų sandaugų sumai. Sakykim, jog žymų yra tiek, kiek sudarytam metode yra sukurtų vartotojo rolių. Jeigu metodas visiškai sukonfigūruotas, tokiu atveju gali daugėti tik vartotojų ir objektų. Jeigu 100% padidinamas objektų skaičius, tai 70% padidėja politikų skaičius. Jeigu 100% padidinamas vartotojų skaičių, tai 30% padaugėja ir politikų skaičius. 3 lent. pateikiami MAC metodo tyrimo rezultatai.

3 lentelė. MAC metodo administravimo kaštų tyrimas

	Bandymas Nr. 1	**Bandymas Nr. 2**	**Bandymas Nr. 3**
Vartotojai	7	14	7
Žymos	6	6	6
Objektai	3	3	6
Politikų skaičius:	60	102	78

DAC metodo politikų skaičius priklauso nuo vartotojų, objektų bei veiksmų skaičiaus ir yra visų jų sandaugos atitikmuo. Jeigu metodas visiškai sukonfigūruotas, tokiu atveju gali daugėti tik vartotojų ir objektų. Jeigu 100% padidinamas objektų arba vartotojų skaičius, tiek pat kartų padaugėja ir politikų skaičius. 4 lent. pateikiami DAC metodo tyrimo rezultatai.

4 lentelė. DAC metodo administravimo kaštų tyrimas

	Bandymas Nr. 1	**Bandymas Nr. 2**	**Bandymas Nr. 3**
Vartotojai	7	14	7
Objektai	3	3	6
Veiksmai	4	4	4
Politikų skaičius:	84	168	168

RBAC metodo politikų skaičius priklauso nuo rolių, objektų bei veiksmų skaičiaus ir yra visų jų sandaugos atitikmuo. Jeigu metodas visiškai sukonfigūruotas, tokiu atveju gali daugėti tik objektų skaičius, vartotojų skaičiaus didėjimas tam poveikio neturės dėl naudojamų rolių. Jeigu 100% padidinamas objektų skaičius, tiek pat kartų padaugėja ir politikų skaičius. 5 lent. pateikiami RBAC metodo tyrimo rezultatai.

5 lentelė. RBAC metodo administravimo kaštų tyrimas

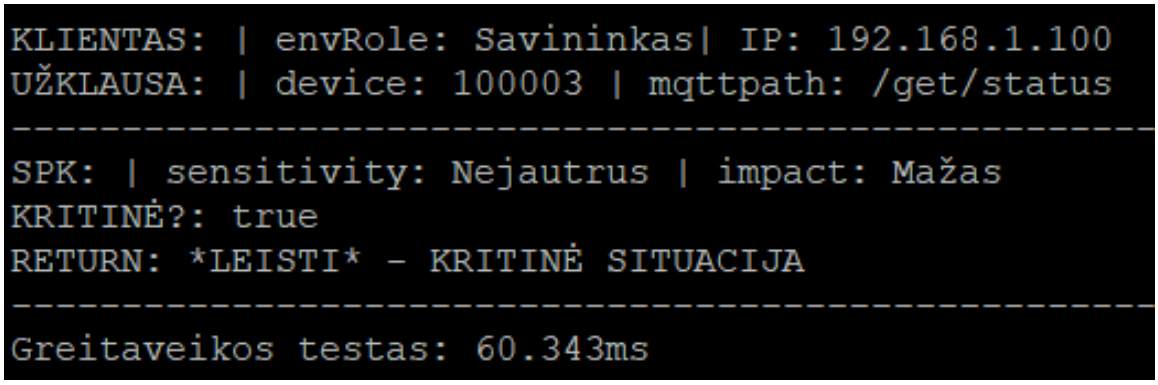
	Bandymas Nr. 1	**Bandymas Nr. 2**	**Bandymas Nr. 3**
Rolės	6	6	6
Objektai	3	3	6
Veiksmai	4	4	4
Politikų skaičius:	72	72	144

Apibendrinus, galima pastebėti, jog darbo metu sudarytą metodą sukonfigūruoti pradžioje yra sudėtingiau dėl naudojamų konteksto sąlygų, kas yra būdinga ir kitiems ABAC tipo logiką naudojantiems metodams. Šiuo bruožu metodas nusileidžia RBAC mechanizmui. Vis dėlto, darbe analizuojamas metodas yra atsparus vartotojų skaičiaus augimui. Šiuo atžvilgiu jis yra pranašesnis už DAC bei MAC metodus. Taigi, sudarytas metodas yra sąlyginai sudėtingas konfigūruojant jį pradžioje, tačiau patogus plečiant sistemą tolesniuose etapuose, kai daugėja vartotojų skaičius, kadangi administravimo kaštai nuo to nedidėja.

4.3.2. Greitaveika

Greitaveika yra kitas parametras, kuris tiriamas. Tyrimas atliekamas pagal pateiktus scenarijus:
1. sprendimų priėmimo procese nustatoma kritinė situacija;
2. sprendimų priėmimo procese nustatoma įprastinė situacija be aplinkos konteksto sąlygų;
3. sprendimų priėmimo procese nustatoma įprastinė situacija su aplinkos konteksto sąlygomis;
4. aplinkos konteksto sąlygų padidėja 100%.

Pirmiausia atliekamas bandymas, kurio metu nustatoma kritinė situacija. 23 pav. matomas pirmojo bandymo pavyzdinis rezultatas.



23 pav. Greitaveikos testas esant kritinei situacijai

Tokie bandymai pakartojami po 10 kartų, kad būtų gaunami statistiškai reikšmingi duomenys. Toks rezultatų rinkinys matomas 6 lent.

6 lentelė. Statistiniai greitaveikos duomenys esant kritinei situacijai

Bandymo eil. nr.	**Nr. 1**	**Nr. 2**	**Nr. 3**	**Nr. 4**	**Nr. 5**	**Nr. 6**	**Nr. 7**	**Nr. 8**	**Nr. 9**	**Nr. 10**
Rezultatas (ms)	60,34	72,19	59,75	77,52	85,58	66,91	64,37	60,30	70,91	68,78

Apskaičiavus gautų duomenų statistinį vidurkį, gaunama, jog esant kritinei situacijai sprendimų priėmimo procesas vidutiniškai užtrunka 68,67 ms.

Toliau atliekamas analogiškas bandymas esant įprastinei situacijai, tačiau neturint su sistema susietų aplinkos konteksto sąlygų. Gauti rezultatai pateikiami 7 lent.

7 lentelė. Statistiniai greitaveikos duomenys esant įprastai situacijai be aplinkos konteksto sąlygų

Bandymo eil. nr.	**Nr. 1**	**Nr. 2**	**Nr. 3**	**Nr. 4**	**Nr. 5**	**Nr. 6**	**Nr. 7**	**Nr. 8**	**Nr. 9**	**Nr. 10**
Rezultatas (ms)	126,27	150,57	173,48	152,86	138,59	143,12	134,22	128,38	154,93	121,46

Apskaičiavus gautų duomenų statistinį vidurkį, gaunama, jog esant įprastinei situacijai, tačiau neturint aplinkos konteksto sąlygų, sprendimų priėmimo procesas vidutiniškai užtrunka 142,39 ms.

Kiti bandymai atliekami esant aplinkos konteksto sąlygoms. Sistemoje pridėtos 6 aplinkos konteksto sąlygos, kurios sprendimo priėmimo metu yra apdorojamos. Surinkti rezultatai pateikiami 8 lent.

8 lentelė. Statistiniai greitaveikos duomenys esant įprastai situacijai su 6 aplinkos konteksto sąlygomis

Bandymo eil. nr.	**Nr. 1**	**Nr. 2**	**Nr. 3**	**Nr. 4**	**Nr. 5**	**Nr. 6**	**Nr. 7**	**Nr. 8**	**Nr. 9**	**Nr. 10**
Rezultatas (ms)	221,52	210,93	235,10	212,10	231,09	204,46	214,02	205,16	185,62	201,94

Apskaičiavus gautų duomenų statistinį vidurkį, gaunama, jog esant įprastinei situacijai su 6 aplinkos konteksto sąlygomis, sprendimų priėmimo procesas vidutiniškai užtrunka 212,19 ms.

Sistemoje pridedamos papildomai 6 aplinkos konteksto sąlygos, kurios sprendimo priėmimo metu bus apdorojamos. Atlikus sprendimų priėmimo proceso greitaveikos tyrimus, šių rezultatai pateikiami 9 lent.

9 lentelė. Statistiniai greitaveikos duomenys esant įprastai situacijai su 12 aplinkos konteksto sąlygų

Bandymo eil. nr.	**Nr. 1**	**Nr. 2**	**Nr. 3**	**Nr. 4**	**Nr. 5**	**Nr. 6**	**Nr. 7**	**Nr. 8**	**Nr. 9**	**Nr. 10**
Rezultatas (ms)	239,50	223,98	233,37	262,72	224,8	203,52	242,78	207,15	204,15	226,08

Apskaičiavus gautų duomenų statistinį vidurkį, gaunama, jog esant įprastinei situacijai su 12 aplinkos konteksto sąlygomis, sprendimų priėmimo procesas vidutiniškai užtrunka 226,81 ms.

Apibendrinus, pastebima, jog esant kritinei situacijai sprendimų priėmimo komponentui užtrunka apie 68,67 ms priimti prieigos sprendimą. Esant įprastinei situacijai, tačiau be aplinkos konteksto sąlygų, sprendimų priėmimo procesas užtrunka dvigubai ilgiau, t. y. apie 142,39 ms. Toliau atliekant bandymus su 6 aplinkos konteksto sąlygomis, greitaveika sulėtėja maždaug 33% iki 212,19 ms. Padidinus aplinkos konteksto sąlygas 100%, greitaveika sulėtėja maždaug 6% iki 226,81 ms. Taigi, kritinės situacijos žymiai sumažina sprendidmų priėmimo proceso laiką.

4.3.3. Saugumas

Šiame skyrelyje atliekamas sudaryto metodo saugumo tyrimas. Jam atlikti pirmiausia sudaroma 10 lent., kurioje nurodyti kiekvieno valdiklių valdymo platformos veiksmo apskaičiuoti baziniai ir bendrieji rizikingumai pagal objektą ir atliekamą veiksmą. Bendrojo rizikingumo balas apima galimą reikšmių intervalą.

10 lentelė. Platformos veiksmų apskaičiuoti baziniai ir bendrieji rizikingumai

Objektas	**Veiksmas**	**Bazinis rizikingumas**	**Bendrojo rizikingumo intervalas**	**Rolės, kurioms leidžiama atlikti veiksmą (pagal bazinį rizikingumą)**
Namai	Išjungti signalizaciją	9	9-18	Savininkas, nuomininkas
	Įjungti židinį	6	6-12	Visos – (vaikas + svečias)
	Įjungti signalizacija	3	3-6	Visos
	Įjungti židinį	3	3-6	Visos
	Įjungti šviesą	3	3-6	Visos
	Išjungti šviesą	3	3-6	Visos
Garažas	Pakelti vartus	6	6-12	Visos – (vaikas + svečias)
	Nuleisti vartus	4	4-8	Visos – svečias
	Gauti statusą	2	2-4	Visos
Sodas	Gauti tinklo nustatymus	3	3-6	Visos
	Paleisti/stabdyti laistytuvą	2	2-4	Visos
	Gauti statusą	1	1-2	Visos

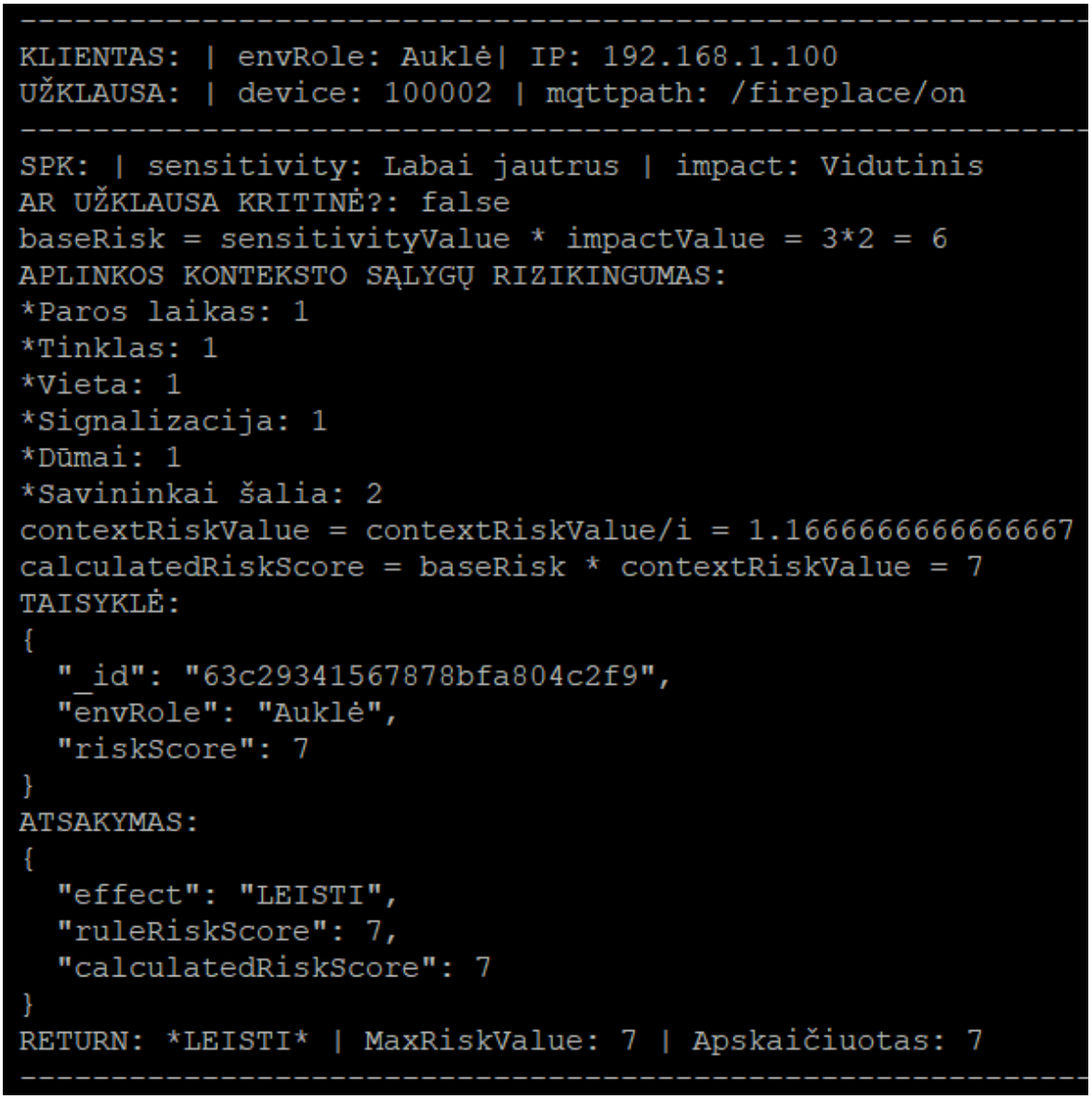
Taigi, pastebima, jog rizikingesni veiksmai (pvz.: išjungti signalizaciją, įjungti židinį, pakelti vartus, nuleisti vartus) nėra leidžiami žemesnio maksimalaus rizikingumo rolėms kaip *svečias* ar *vaikas*. Sprendimas gali pakisti, priklausomai nuo aplinkos situacijos kritiškumo bei esamų aplinkos konteksto sąlygų, kurioms esant apskaičiuotas rizikingumas gali išlikti toks pats arba padidėti 100%.

Toliau vertinama, kaip tie patys prieigos sprendimai gali pakisti, priklausomai nuo aplinkos konteksto sąlygų būsenos. Pagal anksčiau atliktas konfigūracijas, platformoje yra 5 sąlygos (paros laikas, užklausos šaltinis, subjekto šalis, savininko vieta, signalizacijos veikimas), kurios apibūdina aplinką ir gali daryti poveikį bendram rizikingumo balui.

24 pav. pateikiamas pavyzdys, kai auklės rolę turintis subjektas pateikia užklausą įjungti židinį namuose. Komandinės eilutės išvestyje pateikiama, kokią užklausą subjektas atsiuntė, taip pat kokie yra subjekto duomenys, t. y. rolė bei IP adresas. Toliau matoma, kokius tarpinius skaičiavimus atlieka sprendimų priėmimo komponentas.

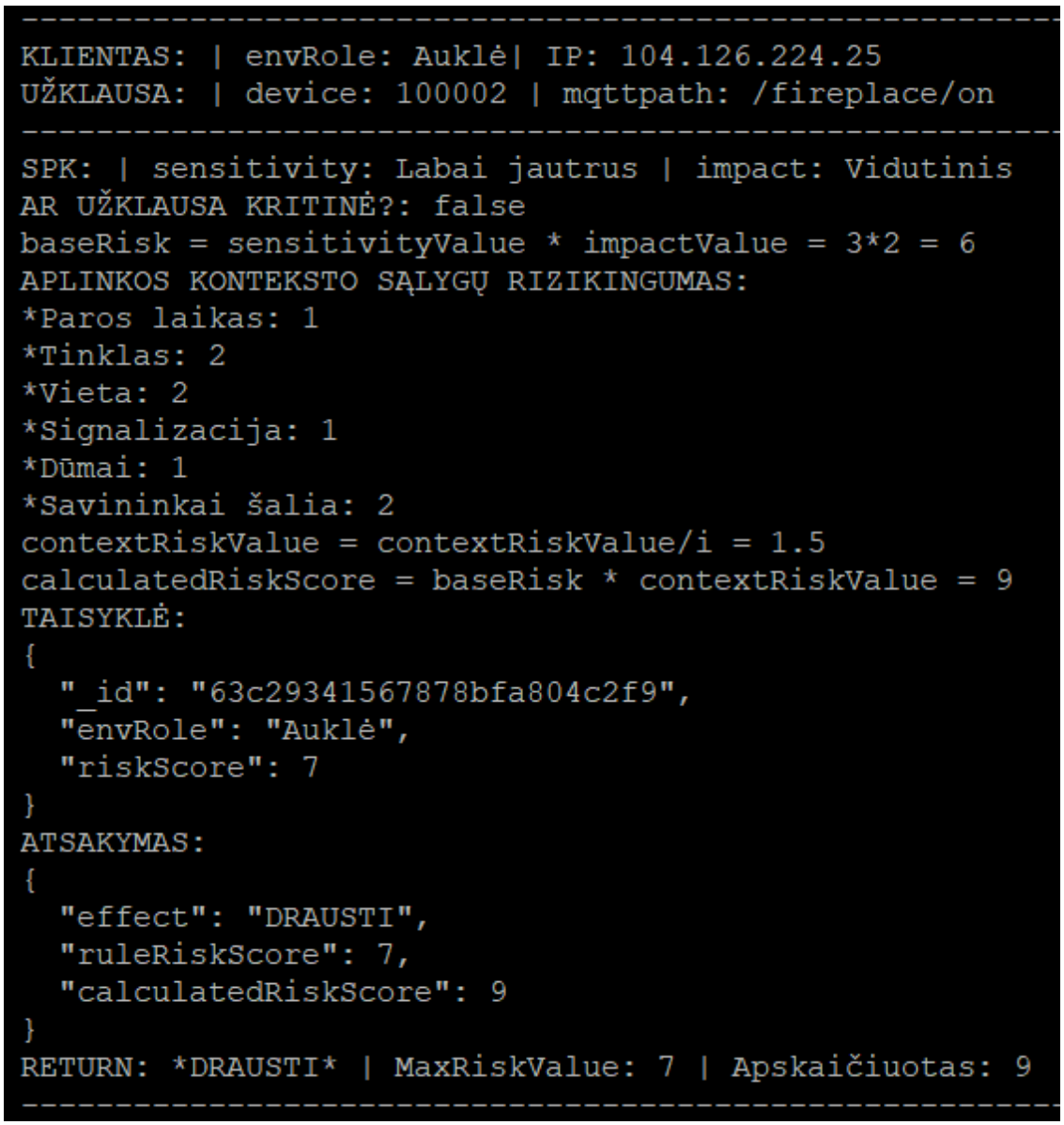
Pirmiausia išvedamos jautrumo ir veiksmo poveikio reikšmės bei patikrinama, ar situacija nėra kritinė. Jeigu situacija įprasta, toliau skaičiuojamas bazinis rizikingumas, t. y. jautrumo ir poveikio sandauga. Tą atlikus, pereinama prie konteksto sąlygų rizikingumo skaičiavimo. Matoma, koks rizikingumo balas skirtas kiekvienai iš aplinkos konteksto sąlygų. Pagal šias reikšmes išvedama galutinė konteksto sąlygų rizikingumo reikšmė *contextRiskValue*. Bendras rizikingumo balas, t. y. bazinio rizikingumo ir aplinkos konteksto rizikingumo sandauga, pateikiamas kaip kintamasis

calculatedRiskScore. Vėliau atliekamas prieigos taisyklės nuskaitymas ir išvestyje pateikiama subjekto rolę atitinkanti prieigos kontrolės taisyklė su nurodytu maksimaliu tai rolei leistinu rizikingumo balu *riskScore*. Pagal šią taisyklę atliekamas palyginimas tarp taisyklėje nurodyto maksimalaus rizikingumo balo bei dinamiškai apskaičiuoto. Pabaigoje pateikiamas atsakymas, ar prieiga yra leidžiama. Taigi, šiuo atveju rizikingumas neviršija auklės rolei leistino maksimalaus rizikingumo, todėl prieiga yra leidžiama.



24 pav. Prieigos užklausos apdorojimas esant normalioms aplinkos sąlygoms

Toliau tas pats bandymas atliekamas atsiradus papildomoms aplinkos sąlygoms. Šiam bandymui atlikti pasitelkiamas *X-Forwarded-For Header* 0.6.2 versijos Google Chrome naršyklės įskiepis, leidžiantis pakeisti HTTP užklausos antraštes. Atliekant bandymą, įskiepis įjungiamas bei pasirenkamas IP adresas, esantis už Lietuvos ribų. 25 pav. matoma, jog nauja subjekto užklausa pateikiama iš išorinio tinklo, o subjekto IP adresas nėra iš Lietuvos.



25 pav. Prieigos užklausos apdorojimas esant rizikingoms aplinkos sąlygoms

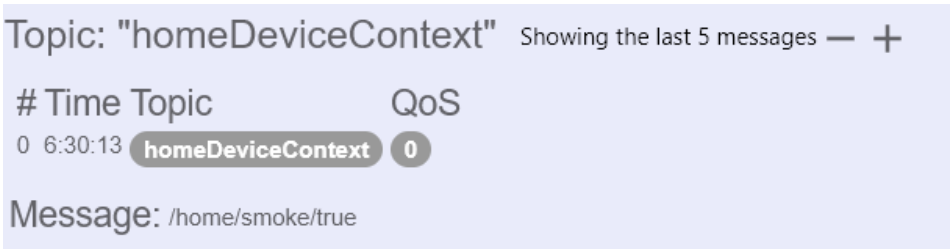
Visa tai vertinama kaip papildomas rizikas keliančios aplinkybės, todėl galutinis rizikingumo balas yra didesnis nei tas, kuris buvo apskaičiuotas 24 pav. pateiktame pavyzdyje. Subjekto sukurta užklausa sprendimų priėmimo komponente šį kartą yra draudžiama, kadangi viršija maksimalų subjekto rolei nustatytą rizikingumą. Taigi, matoma, jog prieigos kontrolės metodas lanksčiai vertina skirtingas situacijas ir atitinkamai priima sprendimus.

Atliekamas kitas scenarijus, kur tikrinamas atvejis, kai vaiko rolę turintis subjektas bando pakelti garažo vartus. Toks veiksmas pagal rizikos nustatymus šiai rolei yra draudžiamas, kas matoma ir 26 pav.

```
------------------------------------------------------------
KLIENTAS: | envRole: Vaikas| IP: 192.168.1.100
UŽKLAUSA: | device: 100001 | mqttpath: /garage/lift
------------------------------------------------------------
SPK: | sensitivity: Jautrus | impact: Didelis
AR UŽKLAUSA KRITINĖ?: false
baseRisk = sensitivityValue * impactValue = 2*3 = 6
APLINKOS KONTEKSTO SĄLYGŲ RIZIKINGUMAS:
*Paros laikas: 1
*Tinklas: 1
*Vieta: 1
*Signalizacija: 1
*Dūmai: 1
*Savininkai šalia: 2
contextRiskValue = contextRiskValue/i = 1.1666666666666667
calculatedRiskScore = baseRisk * contextRiskValue = 7
TAISYKLĖ:
{
  "_id": "63ac929fe3cc8070f484932b",
  "envRole": "Vaikas",
  "riskScore": 4
}
ATSAKYMAS:
{
  "effect": "DRAUSTI",
  "ruleRiskScore": 4,
  "calculatedRiskScore": 7
}
RETURN: *DRAUSTI* | MaxRiskValue: 4 | Apskaičiuotas: 7
------------------------------------------------------------
```

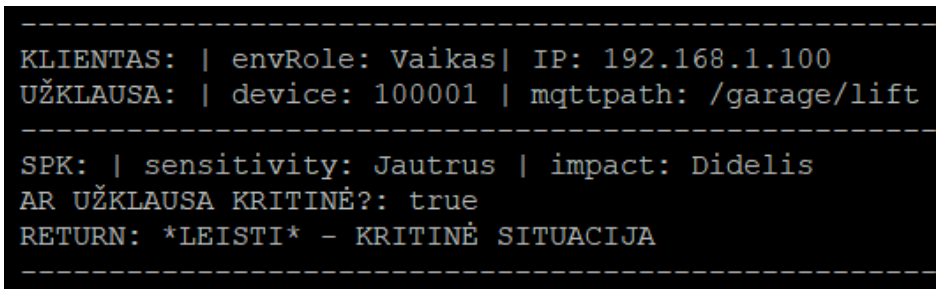
26 pav. Prieigos užklausos apdorojimas esant įprastai situacijai

Toliau inicijuojama aplinkos sąlygos situacija, kuri sukelia kritinę situaciją. Tai inicijuojama sistemoje gavus 27 pav. matomą MQTT žinutę, jog buvo aptikti dūmai:



27 pav. MQTT žinutė apie namuose aptiktus dūmus

Atsižvelgiant į aplinkos konteksto nustatymus tokia žinutė yra apdorojama valdiklių valdymo platformoje ir pakeičiama aplinkos situacija iš įprastinės į kritinę. Tokiu atveju, dar kartą pateikus analogišką prieigos užklausą, 28 pav. matoma, jog vaiko rolę turinčiam subjektui prieiga yra suteikiama.



28 pav. Prieigos užklausos apdorojimas esant kritinei situacijai

Galima pastebėti, jog pastaruoju atveju nėra atliekamas rizikingumo balo skaičiavimas, tačiau iškart aptikus kritinę situacija, suteikiama prieiga. Esant kritinei situacijai, tokia logika galimai leidžia išsaugoti žmogaus sveikatą ir gyvybę.

Apibendrinus, metodo detalumas leidžia įvertinti skirtingas aplinkos sąlygas ir priimti saugiausią mažiausių teisių principą išlaikantį prieigos sprendimą. Vis dėlto, dinamiškas sprendimų priėmimas sunkina auditavimą.

4.4. Prieigos metodo tyrimo išvados

Tyrimas atliktas siekiant įvertinti sudaryto prieigos kontrolės metodo veiksmingumą. Remiantis tyrimo eiga ir scenarijų metu gautais rezultatais galima padaryti šias išvadas:

1. Eksperimentams atlikti realizuota valdiklių valdymo platforma sukonfigūruota ir pritaikyta išmaniųjų namų automatizacijos sprendimams. Pridėti namų aplinką atitinkantys 7 vartotojai, 6 rolės, 6 aplinkos konteksto sąlygos, 3 valdikliai su veiksmų sąrašais. Panaudotas ESP8266 mikrovaldiklis su prijungtais jutikliais bei vykdikliais, imituojančiais išmaniąsias namų sistemas.
2. Atliekant kiekybinį prieigos metodo administracinių kaštų tyrimą buvo skaičiuojamas sudaryto metodo politikų skaičius esant pradinėms platformos konfigūracijoms, taip pat vertintas politikų skaičiaus kitimas didėjant objektų bei subjektų skaičiui. Nustatyta, jog sudarytas metodas reikalauja daugiau administracinių kaštų konfigūravimo pradžioje lyginant su DAC, MAC, RBAC tipo metodais, todėl yra sudėtingesnis. Taip yra dėl to, jog metodui reikalingi atlikti ne tik rolių, bet ir aplinkos konteksto sąlygų konfigūracijas. Sudarytas prieigos metodas, kaip ir RBAC tipo metodai, yra atsparus vartotojų skaičiaus augimui, kadangi naudojamos iš anksto apibrėžtos rolės, taigi, tai leidžia lengviau plėsti sistemą lyginant su DAC ar MAC metodais. Vis dėlto, platformoje augant objektų skaičiui, lygiagrečiai auga ir politikų skaičius, todėl plečiamumas šiuo atžvilgiu nukenčia. Tą galima paaiškinti tuo, jog sudarant metodą buvo siekiama išsaugoti pradinės konfigūracijos paprastumą, dėl ko nebuvo plačiai naudojami ABAC tipo metodams būdingi objektų ar subjektų atributai, kurie būtų išsprendę plečiamumo problemą, tačiau apsunkinę pradinį sistemos konfigūravimą.
3. Tiriant sudaryto prieigos kontrolės metodo greitaveiką, analizuotas laikas, reikalingas sprendimų priėmimo komponente priimti prieigos sprendimą esant skirtingoms situacijoms. Nustatyta, jog esant įprastai situacijai be aplinkos konteksto sąlygų, sprendimas vidutiniškai priimamas per 142,39 ms. Esant 6 aplinkos konteksto sąlygoms, sprendimo priėmimas užtrunka 33% ilgiau, t. y. apie 212,19 ms. Esant kritinėms situacijoms, sprendimas priimamas vidutiniškai per 68,67 ms t. y. 3 kartus greičiau negu esant įprastai situacijai su 6 aplinkos

konteksto sąlygomis. Tai rodo, jog esant rizikingai situacijai, kai galimai gresia pavojus turtui ar žmogaus sveikatai, sprendimų priėmimo komponentas geba prisitaikyti prie situacijos ir priimti sprendimą žymiai greičiau. Atliekant bandymus, kai aplinkos konteksto sąlygų padaugėja 2 kartus, pastebėta, jog greitaveika sumažėja tik 6%.

4. Sudaryto metodo veiksmingumui įvertinti, atliktas kokybinis saugumo tyrimas. Analizuotos skirtingos prieigos užklausos ir situacijos. Įsitikinta, jog metodas yra lankstus, kadangi priima sprendimus remdamasis situacijomis bei rizikingumo balais pagal platformos objektų, veiksmų ir aplinkos konteksto konfigūracijas. Taigi, nėra iš anksto apibrėžtų ir nurodytų prieigos teisių, būdingų statiniams metodams. Kritinės situacijos metu subjektas bet kuriuo atveju gauna prieigą prie objekto. Tokia logika sunkina auditavimą, tačiau suteikia lankstumo ir apsaugo fizinį subjektą, kadangi įvykus kritiniam atvejui, kai galima grėsmė žmogaus sveikatai ar jo turtui, pirmiausia kreipiamas dėmesys į turto ir sveikatos išsaugojimą, o ne informacijos saugumą. Sudarytam prieigos metodui būdingas ir detalumas, kadangi atsižvelgiama į aplinkos savybes, t. y. vertinamos nuolat kintančios aplinkos konteksto sąlygos. Vienu atveju užklausa gali būti įvertinta kaip nerizikinga, kitu atveju – kaip rizikinga, atsižvelgiant į aplinką. Apibendrinus, sudaryto metodo lankstumas ir detalumas leidžia efektyviau užtikrinti saugumą išlaikant mažiausių teisių principą.

Išvados

1. Analizės metu nustatyta, jog daiktų interneto taikomajame sluoksnyje, kuriame veikia valdiklių valdymo platformos, susiduriama su silpna prieigos kontrole. Tradiciniai metodai nėra tinkami daiktų interneto taikymuose, kadangi daiktų interneto infrastruktūroje vyrauja dinamika ir nevienalytė iš daug skirtingų dalių susidedanti aplinka. Statiniai metodai kaip DAC, MAC, RBAC iš anksto apibrėžia prieigos teises, neatsižvelgia į detales ir taip neužtikrina mažiausių teisių principo. Taigi, valdiklių valdymo platformoms reikalingas prieigos kontrolė metodas, atsižvelgiantis tiek į vartotojų roles, objektus, tiek į nuolat kintantį aplinkos kontekstą.
2. Nustatyta, jog siekiant prieigos metodui suteikti detalumą ir lankstumą, naudinga taikyti rizikingumo skaičiavimo algoritmą bei ABAC metodų architektūrą ir veikimo principus: panaudoti komponentų architektūrą bei aplinkos konteksto sąlygų vertinimą, adaptuoti prieigos sprendimus pagal aplinkos situacijas. Identifikuota, jog subjekto ir objekto požymių vertinimas didina metodo sudėtingumą. Pastebėta, jog siekiant dinamiškai įvertinti aplinką ir tuo pačiu išsaugoti metodo paprastumą, rizikos balo skaičiavimui naudinga pasitelkti standartinės bazinės rizikingumo formulės ir aplinkos konteksto sąlygų rizikingumo sandaugą. Paprastumui išlaikyti taip pat efektyvu naudoti RBAC metodų tipams būdingas roles. Metodo lankstumui užtikrinti, veiksminga naudoti situacijų kritiškumo vertinimą, t. y. esant kritinei situacijai subjektui bet kuriuo atveju suteikti prieigą prie objekto darant prielaidą, jog subjektui galimai iškilusi grėsmė sveikatai ar jo turtui.
3. Realizacijos dalyje sudarant valdiklių valdymo platformą įsitikinta, jog „Raspberry Pi 4 model B“ mikrokompiuteris bei „ESP8266 NodeMCU“ mikrovaldiklis yra pakankamai paprasti ir efektyvūs atlikti bandymus valdiklių infrastruktūroje. Nustatyta, jog MQTT protokolas yra lengvasvoris bei pakankamai saugus, kadangi palaiko autentifikaciją bei geba užtikrinti šifruotą duomenų apsikeitimą, todėl puikiai tinka komunikacijai užtikrinti tarp nedaug resursų turinčių daiktų interneto įrenginių. Atliekant vartotojo užklausos IP adreso išgavimą pastebėta, jog geriausia tai atlikti nuskaitant HTTP užklausos antraštę *x-forwarded-for*, kadangi tokiu atveju gaunamas ne platformos IP adresas, bet vartotojo.
4. Tyrimo dalyje atliekant administracinių kaštų eksperimentus nustatyta, jog sudarytą metodą yra sudėtingiau konfigūruoti ir audituoti lyginant su DAC, MAC, RBAC metodais. Papildomi administraciniai kaštai atsiranda dėl siekio užtikrinti detalumą, t. y. dėl aplinkos konteksto sąlygų konfigūravimo, taip pat rolių nustatymų. Metodas naudoja roles, todėl pastebėta, jog vartotojų skaičiaus augimas platformoje nesukelia prieigos politikos skaičiaus augimo. Visgi, augant objektų skaičiui, lygiagrečiai auga ir politikų skaičius, todėl šiuo atžvilgiu platformos plečiamumas nukenčia kaip ir kitų paminėtų metodų. Greitaveikos tyrimai parodė, jog aplinkos konteksto sąlygų vertinimas sulėtina sprendimų priėmimo komponento darbą 33%, o esant kritinėms situacijoms sprendimai priimami 3 kartus greičiau negu esant įprastai situacijai su 6 aplinkos konteksto sąlygomis. Tas rodo, jog esant pavojingoms situacijoms metodas lanksčiai ir greitai vertina susidariusią situaciją ir teikiant prioritetą subjekto saugumui priimamas sprendimas leisti prieigą. Vertinant metodo saugumą, įsitikinta metodo lankstumu esant daiktų interneto infrastruktūrai būdingai dinamikai. Taigi, dėl rizikos balo skaičiavimui naudojamų aplinkos sąlygų ir objekto požymių vertinimo, prieigos teisės nėra statinės ir nekintančios kaip tradiciniuose metoduose. Pastebėta, jog rizikos balo skaičiavimas supaprastina taisyklių sudarymą lyginant su kitais ABAC tipo metodais. Apibendrinus, pasiekti išsikelti reikalavimai metodui, tačiau norint pritaikyti jį platesniam naudojimui, reikėtų leisti konfigūruoti aplinkos

konteksto sąlygas atskiroms infrastruktūros grupėms, kadangi ne visos aplinkos sąlygos yra aktualios visai infrastruktūrai.

Literatūros sąrašas

1. TEWARI, Aakanksha a B.B. GUPTA. Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework. *Future Generation Computer Systems* [interaktyvus]. 2020, **108**, 909–920. ISSN 0167739X [žiūrėta 2021-12-16]. Prieiga per: doi:10.1016/j.future.2018.04.027

2. Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper. *Cisco* [interaktyvus]. [žiūrėta 2021-12-16]. Prieiga per: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

3. BANSAL, Sharu a Dilip KUMAR. IoT Ecosystem: A Survey on Devices, Gateways, Operating Systems, Middleware and Communication. *International Journal of Wireless Information Networks* [interaktyvus]. 2020, **27**(3), 340–364. ISSN 1068-9605, 1572-8129 [žiūrėta 2021-12-16]. Prieiga per: doi:10.1007/s10776-020-00483-7

4. GUPTA, B.b. a Megha QUAMARA. An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols. *Concurrency and Computation: Practice and Experience* [interaktyvus]. 2020, **32**(21), e4946. ISSN 1532-0634 [žiūrėta 2021-12-17]. Prieiga per: doi:10.1002/cpe.4946

5. ISMAIL, Leila a Huned MATERWALA. IoT-Edge-Cloud Computing Framework for QoS-Aware Computation Offloading in Autonomous Mobile Agents: Modeling and Simulation. In: [interaktyvus]. 2021, s. 161–176. ISBN 978-3-030-67549-3 [žiūrėta 2021-12-17]. Prieiga per: doi:10.1007/978-3-030-67550-9_11

6. AGARWAL, Preeti a Mansaf ALAM. Investigating IoT Middleware Platforms for Smart Application Development. In: Sirajuddin AHMED, S. M. ABBAS a Hina ZIA, ed. *Smart Cities—Opportunities and Challenges* [interaktyvus]. Singapore: Springer Singapore, 2020 [žiūrėta 2021-12-19], Lecture Notes in Civil Engineering, s. 231–244. ISBN 9789811525445 [žiūrėta 2021-12-18]. Prieiga per: doi:10.1007/978-981-15-2545-2_21

7. DIZDAREVIĆ, Jasenka, Francisco CARPIO, Admela JUKAN a Xavi MASIP-BRUIN. A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration. *ACM Computing Surveys* [interaktyvus]. 2019, **51**(6), 1–29. ISSN 0360-0300, 1557-7341 [žiūrėta 2021-12-20]. Prieiga per: doi:10.1145/3292674

8. MARTON, Anna. Latest IoT Threat Statistics. *IoTAC* [interaktyvus]. 21. leden 2021 [žiūrėta 2021-12-24]. Prieiga per: https://iotac.eu/latest-iot-threat-statistics/

9. *Evaluating Critical Security Issues of the IoT World: Present and Future Challenges* [interaktyvus]. [žiūrėta 2021-12-24]. Prieiga per: https://ieeexplore-ieee-org.ezproxy.ktu.edu/document/8086136/

10. *OWASP Internet of Things Project - OWASP* [interaktyvus]. [žiūrėta 2021-11-05]. Prieiga per: https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=Main

11. *OWASP Top 10:2021* [interaktyvus]. [žiūrėta 2021-12-25]. Prieiga per: https://owasp.org/Top10/

12. GUPTA, Maanak, Smriti BHATT, Asma Hassan ALSHEHRI a Ravi SANDHU. *Access Control Models and Architectures For IoT and Cyber Physical Systems* [interaktyvus]. Cham: Springer International Publishing, 2022 [žiūrėta 2023-04-24]. ISBN 978-3-030-81088-7. Prieiga per: doi:10.1007/978-3-030-81089-4

13. KRISHNA, Ritika Raj, Aanchal PRIYADARSHINI, Amitkumar V. JHA, Bhargav APPASANI, Avireni SRINIVASULU a Nicu BIZON. State-of-the-Art Review on IoT Threats and Attacks: Taxonomy, Challenges and Solutions. *Sustainability* [interaktyvus]. 2021, **13**(16), 9463. ISSN 2071-1050 [žiūrėta 2021-12-22]. Prieiga per: doi:10.3390/su13169463

14. BABUN, Leonardo, Kyle DENNEY, Z. Berkay CELIK, Patrick MCDANIEL a A. Selcuk ULUAGAC. A survey on IoT platforms: Communication, security, and privacy perspectives. *Computer Networks* [interaktyvus]. 2021, **192**, 108040. ISSN 1389-1286 [žiūrėta 2021-12-22]. Prieiga per: doi:10.1016/j.comnet.2021.108040

15. NEBBIONE, Giuseppe a Maria Carla CALZAROSSA. Security of IoT Application Layer Protocols: Challenges and Findings. *Future Internet* [interaktyvus]. 2020, **12**, 55 [žiūrėta 2021-12-17]. Prieiga per: doi:10.3390/fi12030055

16. ALJERAISY, Atheer, Masoud BARATI, Omer RANA a Charith PERERA. *Privacy Laws and Privacy by Design Schemes for the Internet of Things: A Developer's Perspective* [interaktyvus]. 2020 [žiūrėta 2021-12-26]. Prieiga per: https://hal.archives-ouvertes.fr/hal-02567959

17. AZAM, Fahad, Rashid MUNIR, Mehboob AHMED, M. AYUB, Ahthasham SAJID a Zaheer ABBASI. INTERNET OF THINGS (IOT), SECURITY ISSUES AND ITS SOLUTIONS. *Science Heritage Journal* [interaktyvus]. 2019, **3**(2), 18–21. ISSN 25210858, 25210866 [žiūrėta 2021-12-27]. Prieiga per: doi:10.26480/gws.02.2019.18.21

18. RAVIDAS, Sowmya, Alexios LEKIDIS, Federica PACI a Nicola ZANNONE. Access control in Internet-of-Things: A survey. *Journal of Network and Computer Applications* [interaktyvus]. 2019, **144**, 79–101. ISSN 1084-8045 [žiūrėta 2021-12-28]. Prieiga per: doi:10.1016/j.jnca.2019.06.017

19. PAL, Shantanu. *Internet of Things and Access Control: Sensing, Monitoring and Controlling Access in IoT-Enabled Healthcare Systems* [interaktyvus]. Cham: Springer International Publishing, 2021 [žiūrėti 2021-12-28]. Smart Sensors, Measurement and Instrumentation. ISBN 978-3-030-64997-5. Prieiga per: doi:10.1007/978-3-030-64998-2

20. AL RESHAN, Mana Saleh. IoT-based Application of Information Security Triad. *International Journal of Interactive Mobile Technologies (iJIM)* [interaktyvus]. 2021, **15**(24), 61–76. ISSN 1865-7923 [žiūrėta 2021-12-28]. Prieiga per: doi:10.3991/ijim.v15i24.27333

21. RAGOTHAMAN, Kaushik, Yong WANG, Bhaskar RIMAL a Mark LAWRENCE. Access Control for IoT: A Survey of Existing Research, Dynamic Policies and Future Directions. *Sensors* [interaktyvus]. 2023, **23**(4), 1805. ISSN 1424-8220 [žiūrėta 2021-12-28]. Prieiga per: doi:10.3390/s23041805

22. QIU, Jing, Zhihong TIAN, Chunlai DU, Qi ZUO, Shen SU a Binxing FANG. A Survey on Access Control in the Age of Internet of Things. *IEEE Internet of Things Journal* [interaktyvus]. 2020, **7**(6), 4682–4696. ISSN 2327-4662 [žiūrėta 2021-12-28]. Prieiga per: doi:10.1109/JIOT.2020.2969326

23. AFTAB, Muhammad Umar, Ali HAMZA, Ariyo OLUWASANMI, Xuyun NIE, Muhammad Shahzad SARFRAZ, Danish SHEHZAD, Zhiguang QIN a Ammar RAFIQ. Traditional and Hybrid Access Control Models: A Detailed Survey. *Security and Communication Networks* [interaktyvus]. 2022, **2022**, e1560885. ISSN 1939-0114 [žiūrėta 2021-12-28]. Prieiga per: doi:10.1155/2022/1560885

24. HU, Vincent C., David FERRAIOLO, Rick KUHN, Adam SCHNITZER, Kenneth SANDLIN, Robert MILLER a Karen SCARFONE. *Guide to Attribute Based Access Control (ABAC) Definition and Considerations* [interaktyvus]. NIST SP 800-162. B.m.: National Institute of Standards and Technology. 2014 [žiūrėta 2023-04-24]. Prieiga per: doi:10.6028/NIST.SP.800-162

25. CREMONEZI, Bruno, Alex VIEIRA, José Augusto NACIF a Michele NOGUEIRA. *Survey on Identity and Access Management for Internet of Things* [interaktyvus]. preprint. B.m.: In Review. 2020 [žiūrėta 2021-12-28]. Dostupné z: doi:10.21203/rs.3.rs-66793/v1

26. OUADDAH, Aafaf, Hajar MOUSANNIF, Anas ELKALAM a Abdellah OUAHMAN. Access control in The Internet of Things: Big challenges and new opportunities. *Computer Networks* [interaktyvus]. 2016, **112** [žiūrėta 2021-12-28]. Prieiga per: doi:10.1016/j.comnet.2016.11.007

27. MALIK, Ahmad Kamran, Naina EMMANUEL, Sidra ZAFAR, Hasan Ali KHATTAK, Basit RAZA, Sarmadullah KHAN, Ali H. AL-BAYATTI, Madini O. ALASSAFI, Ahmed S. ALFAKEEH a Mohammad A. ALQARNI. From Conventional to State-of-the-Art IoT Access Control Models. *Electronics* [interaktyvus]. 2020, **9**(10), 1693. ISSN 2079-9292 [žiūrėta 2021-12-20]. Prieiga per: doi:10.3390/electronics9101693

28. KAYES, A. S. M., Rudri KALARIA, Iqbal H. SARKER, Md. Saiful ISLAM, Paul A. WATTERS, Alex NG, Mohammad HAMMOUDEH, Shahriar BADSHA a Indika KUMARA. A Survey of Context-Aware Access Control Mechanisms for Cloud and Fog Networks: Taxonomy and Open Research Issues. *Sensors (Basel, Switzerland)* [interaktyvus]. 2020, **20**(9), 2464. ISSN 1424-8220 [žiūrėta 2021-12-20]. Prieiga per: doi:10.3390/s20092464

29. DRAMÉ-MAIGNÉ, Sophie, Maryline LAURENT, Laurent CASTILLO a Hervé GANEM. Centralized, Distributed, and Everything in between: Reviewing Access Control Solutions for the IoT. *ACM Computing Surveys* [interaktyvus]. 2022, **54**(7), 1–34. ISSN 0360-0300, 1557-7341 [žiūrėta 2021-12-22]. Prieiga per: doi:10.1145/3465170

30. SCIANCALEPORE, Savio, Giuseppe PIRO, Daniele CALDAROLA, Gennaro BOGGIA a Giuseppe BIANCHI. OAuth-IoT: An access control framework for the Internet of Things based on open standards. In: *2017 IEEE Symposium on Computers and Communications (ISCC)*: *2017 IEEE Symposium on Computers and Communications (ISCC)* [interaktyvus]. 2017, s. 676–681 [žiūrėta 2021-12-28]. Prieiga per: doi:10.1109/ISCC.2017.8024606

31. BERTIN, Emmanuel, Dina HUSSEIN, Cigdem SENGUL a Vincent FREY. Access control in the Internet of Things: a survey of existing approaches and open research questions. *Annals of Telecommunications* [interaktyvus]. 2019, **74**(7), 375–388. ISSN 1958-9395 [žiūrėta 2021-12-25]. Prieiga per: doi:10.1007/s12243-019-00709-7

32. JONES, M., J. BRADLEY a N. SAKIMURA. *JSON Web Token (JWT)* [interaktyvus]. RFC7519. B.m.: RFC Editor. 2015 [žiūrėta 2022-01-14]. Prieiga per: doi:10.17487/RFC7519

33. *eXtensible Access Control Markup Language (XACML) Version 3.0* [interaktyvus]. [žiūrėta 2023-04-25]. Prieiga per: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html

34. WESMC7777. *Access control and security for IoT Hub* [interaktyvus]. [žiūrėta 2022-01-14]. Prieiga per: https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-security

35. *Access control with IAM | Cloud IoT Core Documentation | Google Cloud* [interaktyvus]. [žiūrėta 2022-01-14]. Prieiga per: https://cloud.google.com/iot/docs/how-tos/iam

36. *Identity and access management for AWS IoT - AWS IoT Core* [interaktyvus]. [žiūrėta 2022-01-14]. Prieiga per: https://docs.aws.amazon.com/iot/latest/developerguide/security-iam.html

37. *Securing Communication and Access* [interaktyvus]. [žiūrėta 2022-01-14]. Prieiga per: https://www.openhab.org/docs/installation/security.html

38. *Authentication | Home Assistant Developer Docs* [interaktyvus]. [žiūrėta. 2022-01-14]. Prieiga per: https://developers.home-assistant.io/docs/auth_index

39. ATLAM, Hany F. a Gary B. WILLS. An efficient security risk estimation technique for Risk-based access control model for IoT. *Internet of Things* [interaktyvus]. 2019, **6**, 100052. ISSN 2542-6605 [žiūrėta 2021-01-17]. Prieiga per: doi:10.1016/j.iot.2019.100052

40. LTD, Raspberry Pi. Raspberry Pi 4 Model B specifications. *Raspberry Pi* [interaktyvus]. [žiūrėta 2023-04-24]. Prieiga per: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/