Kauno technologijos universitetas

Informatikos fakultetas

Eligijus Kiudys

Rolėmis grįstas žiniatinklio programų prieigos valdymo metodas

Analizė

Vertino:

prof. Algimantas Venčkauskas

Kaunas 2022

Turinys

Įžanga 4

Tikslas ir uždaviniai 4

Dokumento struktūra 4

Žiniatinklio programų prieigos valdymo problemos 5

Konfidencialumas 5

Vientisumas 6

Prieinamumas 6

Prieigos valdymas. 6

Dinamiškumas 6

Globalus susitarimas. (role explosion problemos)(From ABAC to ZBAC: The Evolution of Access Control Models) 6

Modelio valdymas, ... 7

Saugumas 7

Sudėtingumas 7

Žiniatinklio programų prieigos valdymo metodai 8

Rolėmis grįstas prieigos valdymo metodas 9

RBAC Administration 11

Rolėmis grįsto prieigos valdymo metodo pritaikymas žiniatinklyje 12

Cloud services ..... **Error! Bookmark not defined.**

IOT 14

Išvados 15

Literatūros sąrašas 16

Paveikslėlių sąrašas

pav. 1 prieigos valdymo modelio problemos 5
pav. 2 Rolių valdymo modelis 9
pav. 4 Socialinės medijos 12
pav. 3 Wordpress pavyzdys 13
pav. 5 Debesų paslaugos 14

Įžanga

Rolėmis grįstas prieigos valdymo metodas yra vienas iš populiariausių metodų esančių, kurie skirti valdyti internetinių puslapių arba internetinių aplikacijų prieigą. Žinoma yra ne vienas metodas skirtas prieigos valdymui, bet dažniausiai tai būna minėto metodo variacijos arba visiškai skirtingas metodas. Prieigos valdymo problema išlieka ir šiai dienai. Vis daugiau ir daugiau plečiantis internetui ir didėjant tiekiamų paslaugų kiekiui internete reikia ir saugesnių ir patogesnių metodų valdyti internetines aplikacijas, puslapius. Rolėmis grįstas žiniatinklio programų prieigos valdymo metodas leis panaudoti rolėmis grįstą prieigos metodą žiniatinklio programoms. Kadangi žiniatinklio programų kiekis didėja reikia ir metodo, kuris leistų daugiai valdyti naudotojo prieigą prie programos funkcijų ir domenų. Metodas kurį analizuoju yra pagrindinis metodas prieigos valdymui. Analizės metu yra analizuojamos prieigos valdymo problemos, išsiaiškinama apie kitus esamus metodus.

Tikslas ir uždaviniai

Darbo tikslas – sukurti rolėmis grįstą prieigos valdymo metodą skirtą internetinėms programoms. Metodas turėtu leisti pritaikyti rolėmis grįstą prieigos valdymo metodą internetinių aplikacijų prieigos valdymui. Panaudojus naują metodą administratorius galės keisti roles, licencijas, naujoms internetinėms aplikacijoms.

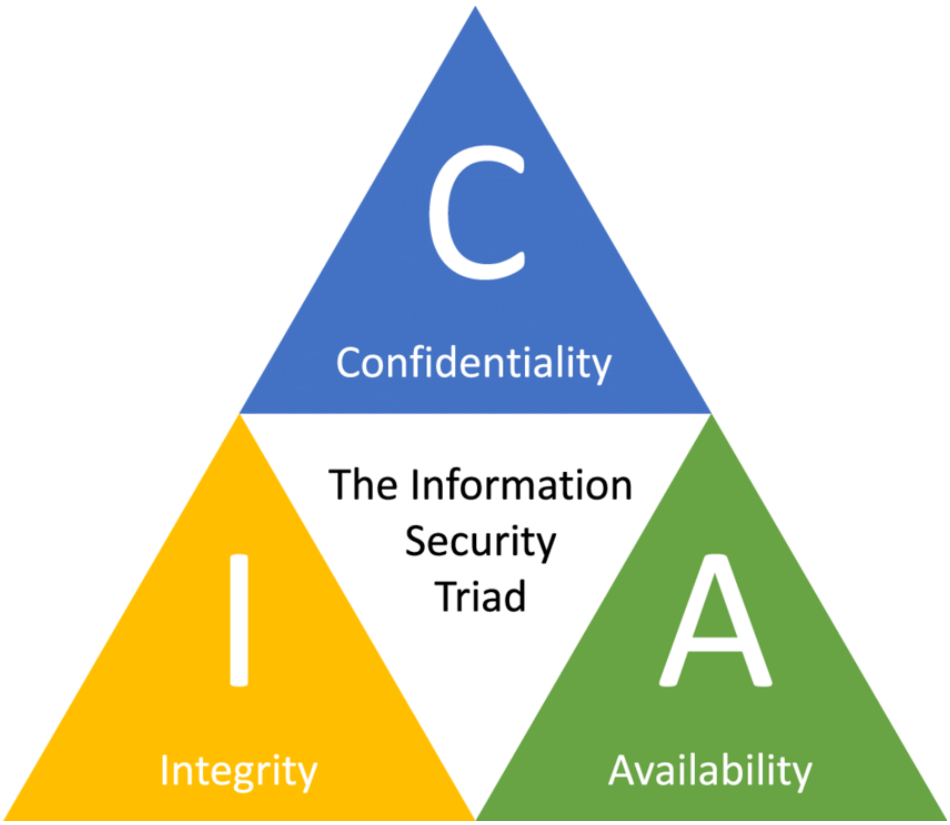
- Išanalizuoti žiniatinklio programų prieigos valdymo problemos.
- Išanalizuoti žiniatinklio programų prieigos valdymo metodus.
- Išanalizuoti rolėmis grįstas prieigos valdymo metodas.
- Išanalizuoti rolėmis grįsto prieigos valdymo metodo pritaikymą žiniatinklyje.
- Pasiūlyti rolėmis grįstą prieigos valdymo metodą skirtą žiniatinklių programoms.
- Įgyvendinti sukurtą metodą ir palyginti rezultatus.

Dokumento struktūra

Darbą sudaro keturi pagrindiniai skyriai - žiniatinklio programų prieigos valdymo problemos, žiniatinklio programų prieigos valdymo metodus, rolėmis grįstas prieigos valdymo metodas, rolėmis grįsto prieigos valdymo metodo žiniatinklyje modelį analizės skyriai. Pirmame skyriuje yra analizuojamos žiniatinklio programų prieigos valdymo esamos problemos. Kitame skyriuje yra analizuojama skirtingi metodai skirti valdyti žiniatinklio programų prieigą. Trečiame skyriuje yra analizuojamas rolėmis grįsto prieigos valdymo metodą, kaip veikia minimas metodas ir šiek tiek istorijos apie metodą. Ketvirtame skyriuje yra analizuojami rolėmis grįsto prieigos valdymo metodo žiniatinklyje.

Žiniatinklio programų prieigos valdymo problemos

Prieigos valdymo problema egzistuoja ne vienerius metus. Ar tai būtu internete ar bibliotekoje, ar universitete. Visą laiką reikia verifikuoti, kad esi tikrai tu, vienokiu ar kitokių būdu. Realybėje yra naudojamos ID kortelės arba pasas. Realybėje galima pamatyti paso nuotrauką ir žmogaus veidą, taip verifikuojant, kad jis tikrai tas žmogus. Internetinėse aplikacijose būtu lengva verifikuoti tapatybę su ID kortele arba pasu, bet juos perkelti į elektroninę erdvę nėra labai lengva. Internetinėse aplikacijos neužtenka patvirtini naudotojo tapatybę, bet reikia valdyti naudotojo prieigą vienu ar kitu būdu. Problemai spręsti visos internetinės aplikacijos ir net tik naudoja pasirinktą modelį naudotojų prieigos valdymui. Kiekvienas modelis turi savus pliusus ir minusus. Sukurti ar panaudoti esamą prieigos valdymo modelį yra sunki užduotis, kuri kankina interneto aplikacijų kūrėjus ne vienerius metus. Naudojamo prieigos modelio užduotis yra paprasta, prileisti naudotojus prie jiems galimos informacijos, nei daugiau, nei mažiau.



pav. 1 prieigos valdymo modelio problemos

Konfidencialumas

Konfidencialumas reiškia kad reikia saugoti naudotojo duomenis ir išlaikyti jų privatumą. Į šitą kategoriją įeina svarbūs duomenys kaip memorandumo duomenys, finansinė informacija, valstybiniai duomenys, bei žinoma naudotojo informacija, kaip slaptažodžiai ir kita informacija svarbi naudotojui. [1]

Vientisumas

Vientisumas yra konceptas apie informacijos saugojimą nuo netinkamo informacijos pakeitimo arba saugojimas nuo naudotojų kuriems yra neleistina pakeisti informaciją, tokiems žmonėms kaip įsilaužėliai. Pavyzdžiui dauguma naudotojų nori užtikrinti, kad banko sąskaitos informacija nebūtu pakeista įsilaužėlių ar bet kokia finansinių programų. Naudotojo informaciją turi galėti pakeisti tik pats naudotojas, arba banko saugos darbuotojas. [1]

Prieinamumas

Prieinamumas yra skirtas apibrėžti, kad informaciją, kurią naudotojas turi pasiekti, gali pasiekti, kai reikia. Sistema turi būti apsaugota nuo įvairių atakų, kaip sistemos perkrova, duomenų perėmimas naudojant sistemą ar teisių gavimas kurių naudotojas negalėjo gauti. Įvykus atakai ir praradus duomenims įmonė turėtu paskelbti apie tokią informaciją naudotojams kurie nukentėjo ir patikrinti ar didelę žalą patyrė naudotojai. [1]

Prieigos valdymas

Pagrindinė problema kurią sprendžia bet kuris pasirinktas prieigos valdymo metodas, tai ir yra prieigos valdymas. Dažniausiai internetinėse aplikacijose yra naudotojai, kurie turi turėti prieigą tik prie tam tikrų duomenų kurie yra skirti tik jiems arba žmonių grupei. Kitaip sakant naudotojai turi turėti galimybę prieiti tik prie jam skirtų duomenų.

Dinamiškumas

Pradėkime nuo pirmos problemos: modelio dinamiškumo. Žinoma pasirinkus prieigos valdymo modelį pradžioje, atrodo, kad gali tikti, bet kuris pasirinktas modelis. Padarius projektą ir administratoriui sukonfigūravus teises naudotojams atrodo viskas veikia kaip tūrėtu veikti. Pradedant plėsti esamą sistemą kūrėjai gali susidurti prieigos valdymo problemomis, kadangi ne visi modeliai leidžia dinamiškai plėsti projektą. Plečiant dinamišką sistemą su blogu prieigos valdymo metodu, dažniausiai yra pradedami naudoti įvairūs apėjimai, kad sistema veiktų. Tokios problemos sistemos administratoriams apsunkina darbą, kadangi realizacija yra neintuityvi, administratoriai valdant tokį projektą gali pridaryti klaidų, kurios veda prie kitų prieigos valdymo problemų.

Globalus susitarimas

Prisijungimas prie sistemos per kitus puslapius. Naudojantis internetinėmis aplikacijomis galima prisijungti prie sistemos naudojant kitą sistemą, dažnai iškyla įvairios problemos prisijungiant su kita sistema, priklausant nuo pasirinkto prieigos valdymo modelio. Viena iš pagrindinių problemų yra naudotojo teisės. Tarp naudojamų metodų turi būti sudarytas susitarimas, kaip koks prieigos valdymo metodas su kitu metodu, jei norima prisijungti prie skirtingu internetiniu sistemų su viena paskyra.

Modelio valdymas

Modelio valdymas yra dar viena problema su kuria susiduriame pritaikant pasirinktą prieigos valdymo metodą. Pritaikius pasirinktą prieigos valdymo metodą, administratorius dažniausiai turi valdyti naudotojų dalinę arba pilną prieigą. Dažnai neintuityvus modelio valdymas gali privesti prie administratoriaus klaidų, kurios suteikia naudotojams prie mažai teisių arba per daug.

Saugumas

Kiekviena sistema turi saugumo spragų, ar tai būtu pritaikymo problemos, ar tai būtu administravimo spragos, ar kažkokios kitos klaidos kurios priveda prie nesaugios sistemos. Vienas iš pavyzdžių būtu rolėmis paremto prieigos metodo rolių sprogimo problema. Šitame pavyzdyje yra aprašoma kaip rolių kiekis didėja iki tokių skaičių kai jų nebegalima suvaldyti. Pavyzdžiui yra vienas naudotojas, dešimt sistemų ir dvi rolės per sistemą. Naudotojas reikalauja dviejų rolių per aplikaciją gaunasi taip, kad reikalauja dvidešimt rolių. Turint daugiau negu vieną naudotoją ir jiems visiems prašant dviejų rolių per aplikaciją, tikėtina kad tvarkant roles administratorius gali įvelti ne vieną klaidą.

Sudėtingumas

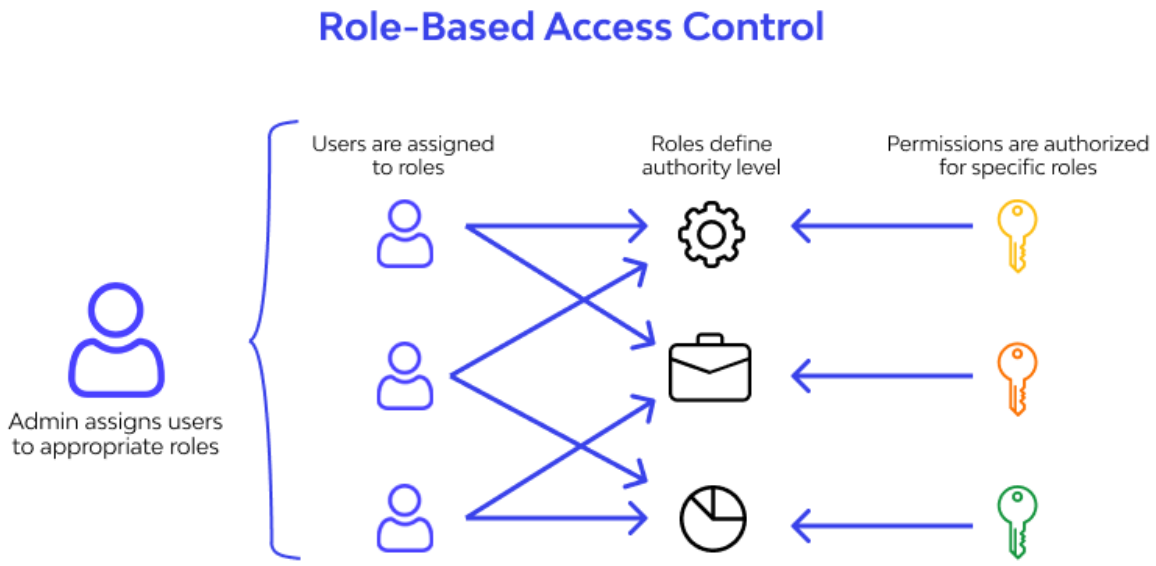
Sistemos sudėtingumas irgi yra problema. Norint integruoti pasirinktą prieigos valdymo metodą, sistemos sudėtingumas gali kišti koją. Programuotojai gali nesuprasti sudėtingos sistemos, ją pritaikant gali padaryti saugos klaidų. Žinoma administratoriui bus lengviau dirbti, bet kartai geriau yra pasirinkti sunkiau valdomą sistemą, bet lengviau suprantamą ir įdiegiamą.

Žiniatinklio programų prieigos valdymo metodai

- Object-Specific Role-Based Access Control (ORAC)[2]
  Konkrečių objektų rolėmis pagrystąs prieigos valdymo metodas. Šitas metodas valdo naudotojų prieigą rolėmis ir specifiniais objektais. Administratorius gali nustatyti naudotojui rolę, arba specifinį objektą, prie kurio naudotojas gali prieiti.
- Attribute-based access control[3]
  Naudojant atributais pagrįsto prieigos valdymo metodo prieiga prie išteklių gali būti nustatyta pagal skirtingus požymius pvz. vardas, IP adresas, laikas ir tt. Pagrindinė idėja ABAC yra, kad nėra tiesiogiai priskirti leidimų vartotojui, dėl to objektų prieiga leidžia visų objektų prieigą remiantis atributais. Atributas vaidina svarbų vaidmenį sistemoje ABAC leidimų suteikimas įgaliotiems vartotojams, pvz., vardas, vieta, IP adresas, vieta ir tt. Atributo reikšmė nusprendžia, ar vartotojas yra įgaliotas naudoti tam tikrą ištekli, ar ne. Vartotojas taip pat gali būti nurodyta kaip subjektas.
- Role-based Access Control[4]
  Pagrindinė RBAC koncepcija yra rolės, kurios gali parodyti prieigos kontrolės politiką tam tikrai organizacijai, institucijai ar įmonei. Leidimai sukuriami, kai objektai atliekami veiksmai, o po to šioms rolės priskiriami leidimai. Vartotojai nėra tiesiogiai priskirtas leidimams. Rolė yra tiltas tarp leidimų ir vartotojai. Vartotojams priskiriami nurodytos rolės, kad jie galėtų naudotis skirtingais leidimais.
- Attributed Role Based Access Control Model[5]
  Modelis naudoja roles kaip tiltą tarp objekto leidimų ir naudotojo. Administratorius naudotojui nustato pasiriktą rolę. Šitas modelis skiriasi nuo paprasto rolėmis pagrįsto prieigos metodo tuo, kad objektų leidimai yra priskiriami automatiškai prie rolių, naudojant atributais paremto prieigos valdymo modelio. Rolės yra išskirstytos lygiais, pvz. naudotojas kuri priklauso pirmo lygio rolei gali atlikti skaitymo, rašymo, tvarkymo ir trynimo veiksmus.
- RBAC-SC: Role-based Access Control using Smart Contract[6]
  Rolėmis grįstas prieigos valdymo metodas naudojant išmanų kontraktą, veikia laba panašiai kaip Rolėmis grįstas prieigos valdymo metodas, tik yra vienas skirtumas, minėtas metodas naudoja blockchain technologiją saugiai gauti rolės prieigą ir atlikti tik naudotojui skirtas funkcijas.
- Intent-Based Access Control (IBAC)[7]
  Šiek tiek istorijos apie prieigos valdymo metodus. Pradedant naudotis kompiuteriais, žmonės suprato, kad reikėjo neleisti naudotojams neleisti vienas kitam kištis į darbus, kai naudotojai naudojasi vienu kompiuteriu. Problemai spręsti buvo sukurtas IBAC modelis, kuris priklauso nuo naudotojo tapatybės. Leidimas naudoti sistemos resursus pzv. Failus, buvo indeksuojami pagal vartotojo tapatybę, tai reiškia, kad failą gali redaguoti vienas arba keli žmonės priklausant nuo leidimo.

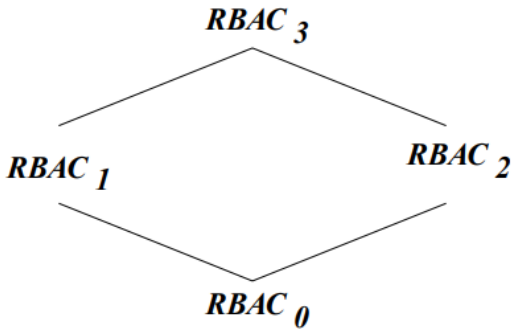
Rolėmis grįstas prieigos valdymo metodas

Taigi kas yra ta rolėmis grįstas prieigos valdymo metodas. Pagrindinė RBAC koncepcija yra rolės, kurios valdo naudotojo prieigą prie sistemos. Rolės yra tiltas tarp naudotojo ir sistemos licencijų. Naudotojas gavęs rolę gali pasiekti administratoriaus skirtą funkcionalumą. Administratorius valdo naudotojus, roles, licencijas. Dažniausiai administratoriui yra skirta valdymo rolė, su kuria gali valdyti naudotojus, roles ryšiu ir t.t. Niekas kitas be administratoriaus negali prieiti prie rolių valdymo.

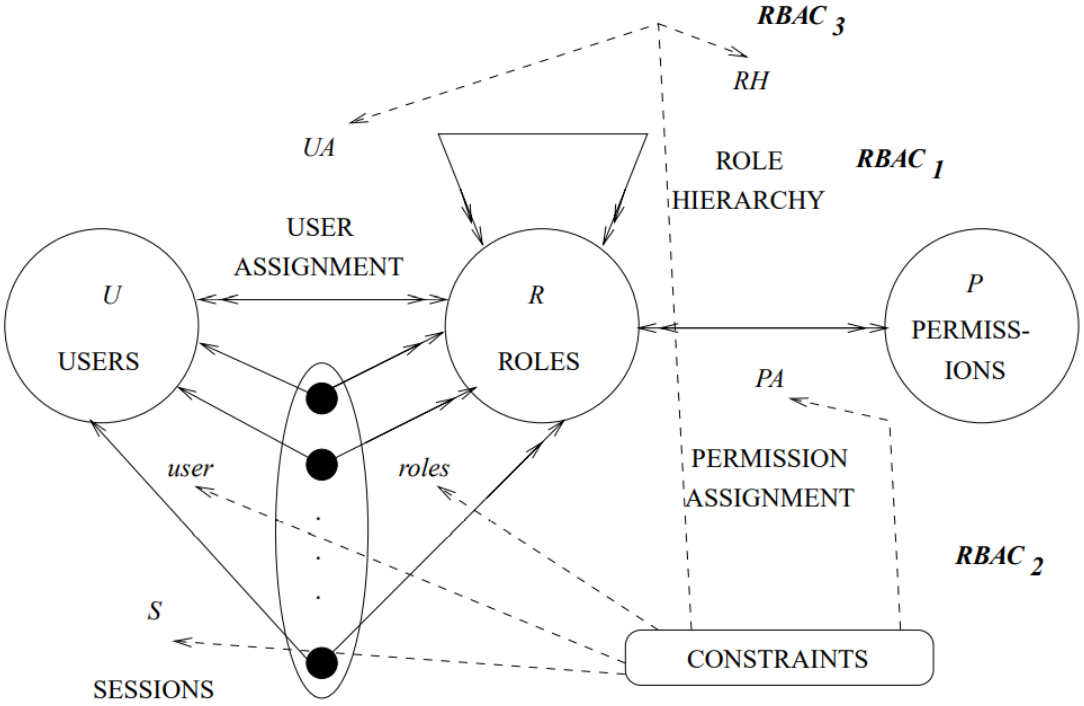


pav. 2 Rolių valdymo modelis

Rolėmis grįstas prieigos metodas atsirado 1990-ais, kaip patikima technologija valdant dideles sistemas. Rolėmis grįsto prieigos metodo paprastas paaiškinimas yra naudotojams yra priskirtos rolės kurios yra susietos su leidimais. Toks metodas palengvina sistemos valdymą. Galime pagalvoti, kas būtu jei nebūtu rolėmis pagrįsto prieigos valdymo metodo. Greičiausia arba naudotume senesnį modelį arba dar nematytą modelį.[7]



(a) Relationship among RBAC models



(b) RBAC models

Fig. 1. A family of RBAC models.

Be abejonių, ši technologija yra dabar viena iš populiariausių technologijų, valdant prieigą internete ir įvairiuose įrenginiuose.

Naudodami RBAC, sistemos administratoriai gali kurti roles, suteikti toms rolėms leidimus ir priskirti vartotojų roles pagal jų konkrečias darbo pareigas ir politiką. Visų pirma, vaidmenų ir teisių ryšiai gali būti nustatyti iš anksto, todėl vartotojus lengva priskirti iš anksto nustatytoms rolėms. Be RBAC būtu sunku nustatyti, kokie leidimai turi būti suteikti naudotojams.

RBAC Administration

Pradėkime nuo to kas valdo roles. Tai roles valdo administratorius. Administratorius yra asmuo kuris sutvarko sistemos saugos politiką. Administratorius atlieka sistemos auditą, tvarko sistemos roles, tvarko sistemos leidimus kurie yra susieti su rolėmis, kitaip sakant priskiria leidimus prie rolių. Administratorius yra pagrindinis asmuo kuris prižiūri sistemos saugumą. [8]

Į administravimo funkcijas įeina leidimų kūrimas ir palaikymas elementams. Su administravimo funkcijomis galima sukurti naudotojus, roles, operacijas ir objektus. Turint prieigą prie tokių funkcijų administratorius gali valdyti visos sistemos prieiga naudotojams i naudotojų kiekį. Naudojant administravimo funkcijas galime nustatyti ryšius tarp naudotojų, rolių ir objektų. Pasinaudojus funkcijomis taip pat galima ir panaikinti sukurtus ryšius. Tokios funkcijos suteikia administratoriui valdyti visą sistema efektyviai ir dinamiškai. Žinoma augant sistemai auga ir darbo kiekis, kadangi reikia valdyti didesnį kiekį rolių ir jų ryšių.

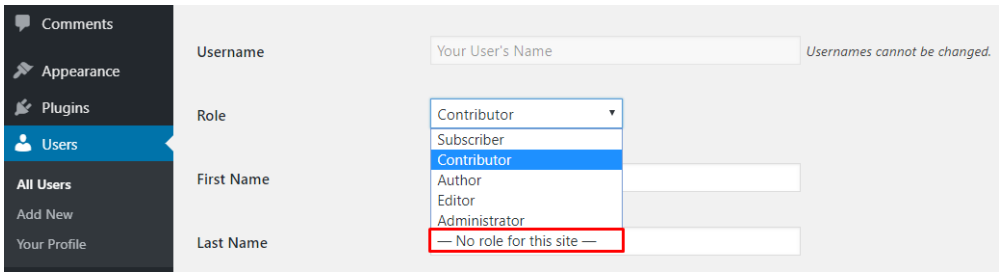
Rolėmis grįsto prieigos valdymo metodo pritaikymas žiniatinklyje

Aptarėme kas yra rolėmis grįstas valdymo metodas, dabar galima išsiaiškinti kur jį galima pritaikyti žiniatinklių, programų srityje. Pirmąjį pritaikymą radau socialiniuose tinkluose. Taigi trumpai apie socialinius tinklus. Socialiniais tinklų puslapiais naudojasi ne vienas milijonas žmonių. [9]

pav. 3 Socialinės medijos

Klausimas kyla kodėl butu galima panaudoti socialiniuose tinkluose. Užtenka truputį pagalvoti ir viskas tampa aišku. Socialinėse medijos paaiškintas prieigos valdymo metodas yra naudojamas naudotojų ir prieigos valdymui. Socialinės medijos turi didelį funkcionalumo kiekį, didelį vidinių grupių bei vidinių puslapių kiekį. Kiekvieną puslapį ar grupę kažkas valdo. Būtent tam valdymui yra naudojamas rolėmis pagrystą prieigos valdymo metodas, arba nors panašus metodas kuris remiasi šiuo metodu.

Susipažinus su pačiu metodu galiu paaiškinti, kur jis yra naudojamas. Nuo pat modelio sukūrimo laiko buvo aišku, kad šis modelis bus naudojamas įvairiuose sistemose. Viena iš tokių sistemų yra internetiniai puslapiai. Šie puslapiai rolėmis pagrįsto prieigos valdymo metodą ne vienerius metus. Galima paimti kaip pavyzdį puslapius sukurtus su Wordpress įrankiu. Šitas įrankis naudoja rolėmis pagrystą prieigos valdymą. Tinklalapiai kurie yra sukurti Wordpress pagrindu, prieigą valdo naudojant minėtą metodą.

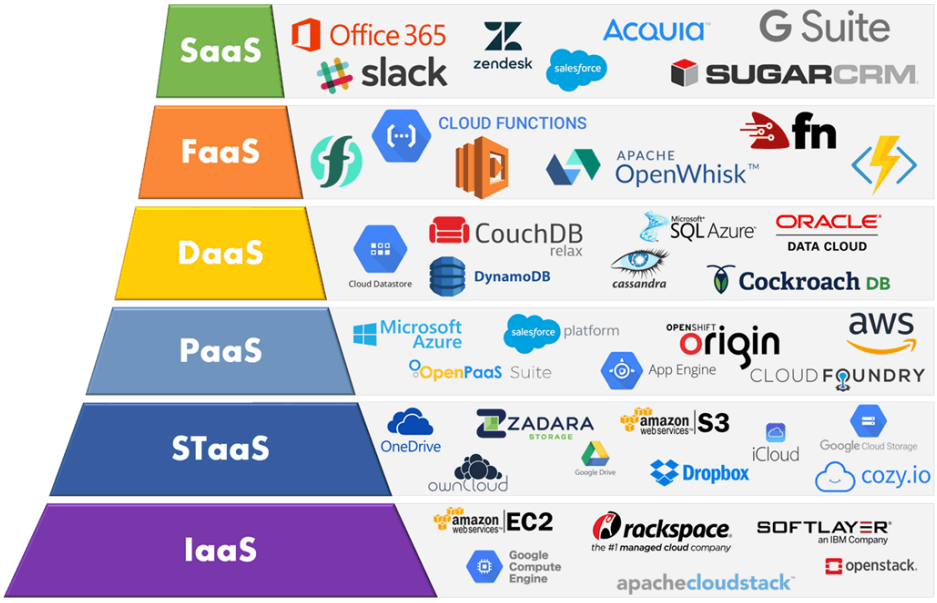


pav. 4 Wordpress pavyzdys

Kitas pritaikymas yra duombazėse. Taigi duombazės dažniausiai yra valdomos tiesiogiai valdant leidimus. Duombazėje yra naudotojai kurie gali valdyti skirtingas lenteles, kad naudotojas galėtų valdyti lenteles prie naudotojo yra priskirti leidimai skirti prieigai prie lentelės. Duombazėse būtu galima pritaikyti rolėmis grysta prieigos valdymo metodą, palengvinti naudotojų ir skirtingų duombazių valdymą.

Debesų paslaugos

Vis dažniau ir dažniau išgirstame terminą debesų paslaugos. Taigi pirma išsiaiškinkime kas yra debesų paslaugos ir tada galėsime pardėti kalbėti apie rylelėmis grįsto prieigos metodo taikymą šituose paslaugose. Debesų paslaugos yra teikiamos įvairios paslaugos per internetą, naudojant serverį, tai gali būti elektroninis paštas, Microsoft Office 365 ir kitokios aplikacijos kurias galima pasiekti per internetą. [10] Tokioms paslaugoms reikia valdyti naudotojų prieigą prie aplikacijos ar paslaugos. Žinoma galima įvairiai valdyti naudotojų prieigą, bet šiuo momentu kalbame apie rolėmis grįsto prieigos metodą. Taigi kaip galima pritaikyti rolėmis grįstą prieigos metodą debesio paslaugomas. Na pirmiausia kaip ir prie visų paslaugų naudotojas turi prisijungti prie sistemos. Paslaugų teikimas gali būti nevienodas visiems, tam galima pritaikyti rolėmis grįstą prieigos metodą. Kiekviena rolė turi prieiga prie tam tikrų paslaugų. Žinoma administratoriui atitenka daug darbo reguliuoti roles, todėl yra naudojamas kitas metodas arba automatinis rolių valdymo metodas.



pav. 5 Debesų paslaugos

IOT

Daiktų internetas yra įrenginys prijungtas prie interneto, kuris atlieka kažkokias funkcijas. Daiktų internetas nuolatos stipriai auga, Bilijonai įrenginių yra prijungti prie interneto ir internetinių aplikacijų, kurios suteikia įvairias galimybes tokiems įrenginiams.[3] Taigi šiomis dienomis visi įrenginiai yra susieti per internetą, kur atlieką įvairius skaičiavimus arba tiesiog teikia įvairias funkcijas naudotojams. Suvaldyti duomenis kurie yra apdorojami įrenginių reikia kažkokio modelio kuris sugebėtu tai įvykdyti. Kaip ir minėjau prie debesų paslaugų tiekėjų, reikia suvaldyti prieigą prie paslaugų. IoT įrenginiai naudoja debesų teikiamas paslaugas, todėl jų prieigą reikia valdyti. Kadangi prie IoT įrenginio gali prisijungti ne vienas naudotojas, reikia valdyti naudotojų prieiga, prie ko gali prieiti kiekvienas naudotojas

Išvados

- Išanalizavus žiniatinklio programų prieigos valdymo problemas pastebėta, kad kuriant prieigos valdymo modelį yra ne viena problema su kuria susiduriama. Pirmiausia tai yra žinomiausia problema duomenų apsaugos problema. Kita viena iš pagrindinių problemų yra naudotojo konfidencialumas, apie kurį reikia pagalvoti. Paskutinė didelė problema yra sistemos prieiga, reikia pagalvoti apie sistemos apkrovą naudojant prieigos valdymo metodą. Žinoma yra ir kitų problemų su kuriomis galima susidurti.
- Išanalizavus žiniatinklio programų prieigos valdymo metodus buvo pastebėta, kad jų yra daug ir skiriasi drastiškai nuo analizuojamo metodo, arba yra metodai yra paremti rolėmis grįsto valdymo metodu.
- Atlikus rolėmis grįsto prieigos valdymo metodo analizę buvo suprasta, kad šitas metodas nėra toks lengvas kaip iš pradžių buvo galvota. Buvo pamatytą, kad jį panaudoti reikia daug žingsnių kuriuos reikia atlikti. Pastebėta kaip veikia naudotojų valdymas, rolių valdymas ir licencijų tvarkymas. Išsiaiškinta, kaip turėtų veikti prieigos valdymas iš administratoriaus pusės.
- Buvo išanalizuota kur žiniatinklyje yra pritaikytas rolėmis grįstas metodas. Buvo pastebėta, kad yra metodų kurie remiasi rolių prieigos valdymo metodus svetainėse, kurios naudoja programinę įrangą Wordpress ar panašią įrangą. Taip pat buvo išsiaiškinta kad rolėmis grįstas prieigos valdymo metodas yra naudojamas IoT įrenginiams ir jų funkcionalumui valdyti.

Buvo išanalizuotas rolėmis grįstas prieigos valdymo metodas, bei jo pritaikymas internete. Taip pat buvo atkreiptas dėmėsis ir į kitus metodus, kurie yra taip pat skirti prieigos valdymui.

Literatūros sąrašas

[1] D. R. K. R. C. David F.Frraiolo, *Role-Based Access Controll*. 2003.

[2] N. Mundbrod and M. Reichert, "Object-Specific Role-Based Access Control," *Int. J. Coop. Inf. Syst.*, vol. 28, no. 1, pp. 1–30, 2019, doi: 10.1142/S0218843019500035.

[3] S. Bhatt, T. K. Pham, M. Gupta, J. Benson, J. Park, and R. Sandhu, "Attribute-Based Access Control for AWS Internet of Things and Secure Industries of the Future," *IEEE Access*, vol. 9, pp. 107200–107223, 2021, doi: 10.1109/ACCESS.2021.3101218.

[4] M. Belchior, D. Schwabe, and F. Silva Parreiras, "Role-based access control for model-driven web applications," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7387 LNCS, pp. 106–120, 2012, doi: 10.1007/978-3-642-31753-8_8.

[5] M. U. Aftab *et al.*, "Permission-based separation of duty in dynamic role-based access control model," *Symmetry (Basel).*, vol. 11, no. 5, 2019, doi: 10.3390/sym11050669.

[6] J. P. Cruz, Y. Kaji, and N. Yanai, "RBAC-SC: Role-based access control using smart contract," *IEEE Access*, vol. 6, no. c, pp. 12240–12251, 2018, doi: 10.1109/ACCESS.2018.2812844.

[7] A. Karp, H. Haury, and M. Davis, "From ABAC to ZBAC: The evolution of access control models," *5th Eur. Conf. Inf. Manag. Eval. ECIME 2011*, vol. 9, no. 2, pp. 202–211, 2011.

[8] L. Dongdong, X. Shiliang, Z. Yan, T. Fuxiao, N. Lei, and Z. Jia, "Role-based access control in educational administration system," *MATEC Web Conf.*, vol. 139, pp. 1–8, 2017, doi: 10.1051/matecconf/201713900120.

[9] J. Li, Y. Tang, C. Mao, H. Lai, and J. Zhu, "Role based access control for social network sites," *2009 Jt. Conf. Pervasive Comput. JCPC 2009*, pp. 389–393, 2009, doi: 10.1109/JCPC.2009.5420153.

[10] W. T. Tsai and Q. Shao, "Role-based access-control using reference ontology in clouds," *Proc. - 2011 10th Int. Symp. Auton. Decentralized Syst. ISADS 2011*, vol. 2, pp. 121–128, 2011, doi: 10.1109/ISADS.2011.21.