VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Duomenų bazių veiklos stebėsenos įrankio (IPS/IDS) funkcionalumo projektavimas pasirinktai DBVS

Design of Database Activity Monitoring Tool (IPS/IDS) Functionality for a Chosen DBMS

Bakalauro darbas

Atliko:	Tomas Orvidas	(parašas)
Darbo vadovas:	asist. Karolis Uosis	(parašas)
Darbo recenzentas:	lekt. Viktoras Golubevas	(parašas)

Vilnius – 2016

SANTRAUKA

Šiame darbe yra nagrinėjama problema, jog dėl duomenų bazių valdymo sistemų specifiškumo, tarpusavio skirtumų techninėje pusėje, jų veikime, realizacijoje ir esamų pažeidžiamumų universalios įsilaužimo aptikimo/prevencijos sistemos negali pilnai apsaugoti visų duomenų bazių valdymo sistemų. Todėl reikia kurti specializuotas įsilaužimo aptikimo/prevencijos sistemas, pritaikytas specifinėms duomenų bazių valdymo sistemoms. Sprendžiant šią problemą buvo išanalizuotas įsilaužimo aptikimo/prevencijos sistemų veikimas, techniniai aspektai, naudojamos technikos, išnagrinėti sunkumai apsaugant duomenų bazes nuo įsilaužimų. Buvo suprojektuotos įsilaužimo aptikimo sistema ir įsilaužimo prevencijos sistema, atitinkamai skirtos pasyviai ir aktyviai pasirinktos duomenų bazių valdymo sistemos apsaugai nuo įsilaužimų. Taip pat buvo gautos išvados, jog žinomiems įsilaužimams geriausiai tinka parašu paremtas aptikimas, naujiems/nežinomiems įsilaužimams geriausiai tinka anomalijomis paremtas aptikimas ir tai, kad duomenų bazių apsaugą apsunkina SQL kalbos ekspresyvumas.

Raktiniai žodžiai: įsilaužimo aptikimo sistema, įsilaužimo prevencijos sistema, duomenų bazių valdymo sistema.

SUMMARY

In this work the problem is examined that due to database management system specificity, differences in technical side, their operation, implementation and current vulnerabilities the universal intrusion detection/prevention systems cannot fully protect all database management systems. Therefore, it is necessary to develop specialized intrusion detection/prevention systems adapted to specific database management systems. While solving this problem the operation, technical aspects, used techniques of intrusion detection/prevention systems were analysed, difficulties of database protection from intrusions were investigated. An intrusion detection system and an intrusion protection system were designed respectively for passive and active protection of a chosen database management system from intrusions. Moreover, conclusions were found that for known intrusions it is best to use signature-based detection, for new/unknown intrusions it is best to use anomaly-based detection and that SQL expressiveness makes database protection more difficult.

Keywords: intrusion detection system, intrusion prevention system, database management system.

TURINYS

ĮVADAS

Per paskutinius dešimtmečius išpopuliarėjus kompiuteriniams tinklams ir bendram interneto naudojimui, vis daugiau organizacijų kuria vidinius kompiuterinius tinklus, kurių pagalba organizuojamas darbas, dalinamasi informacija organizacijos viduje, kuriama vis daugiau sistemų, leidžiančių vis daugiau paslaugų teikti internetu. Tačiau dėl šios plėtros išaugo ir įsilaužimo (angl. Intrusion) į elektroninius tinklus ir sistemas grėsmė. Įsilaužimas elektroninėse sistemose yra neautorizuoto vartotojo prisijungimas prie sistemos arba autorizuoto vartotojo neautorizuota veikla toje sistemoje. Apsaugoti kompiuteriams ir tinklams buvo kuriamos įvairios antivirusinės programos bei ugniasienės. Tačiau jos negali atpažinti bei užkirsti kelią daugumai įsilaužimo būdų, kadangi tobulėjant technologijoms atsiranda vis naujų būdų bei tobulėja esami. Norint kovoti su įsilaužimo grėsme atsirado ir išpopuliarėjo įsilaužimo aptikimo sistemos (angl. IDS – Intrusion Detection System). Šios sistemos stebi veiklą, vykstančią kompiuteriniuose tinkluose, pavieniuose kompiuteriuose ar programose/sistemose, siekiant juose aptikti pažeidžiamumus (angl. Vulnerabilities), galimą neleistiną veiklą ar neautorizuotą prieigą. Aptikus tokią veiklą, įsilaužimo aptikimo sistemos siunčia pranešimus administratoriams, kurie, remiantis pateikiama informacija, nusprendžia kaip toliau elgtis. Nors šios sistemos ir padeda aptikti grėsmes, tačiau jos yra tik pasyvūs stebėtojai, kadangi jos pačios neturi jokių prevencijos priemonių. Norint tai ištaisyti atsirado įsilaužimo prevencijos sistemos (angl. IPS – Intrusion Prevention System), kurios gali ne tik aptikti galimą įsilaužimą, bet ir turi prevencinių galimybių užkirsti kelią įsilaužimui ir apsaugoti tinklą, sistemą, kompiuterį nuo galimos žalos. Tarp sistemų, kurioms reikia tokios apsaugos yra ir duomenų bazių valdymo sistemos, kurios dažnai saugo asmenų ar organizacijų kritinę/konfidencialią informaciją, kurią siekiama apsaugoti nuo neteisėtos prieigos.

Duomenų bazių valdymo sistemos, kaip programų sistemos, turi savitą veiklą, apimančią funkcionalumą, naudojamus protokolus komunikacijai su kitomis sistemomis, savus apsaugos mechanizmus, todėl kiekvienai sistemai reikia atskirai kurti bei pritaikyti įsilaužimo aptikimo/prevencijos sistemas. Norint sukurti tinkamą įsilaužimo aptikimo/prevencijos sistemą kokiai nors programų sistemai reikia gerai išanalizuoti jų veikimą, galimus esamus defektus (angl. Defect, Bug) bei esamas saugumo spragas.

Tarp duomenų bazių valdymo sistemų yra labai pažeidžiamos atviro kodo sistemos. Kadangi jų išeities kodas yra visiems laisvai prieinamas, tie kas norėtų neleistinai įsilaužti į jas, nesvarbu ar jas naudoja pavieniai asmenys ar organizacijos, galėtų pasinaudoti esamu laiku jose esančiais defektais

bei saugumo spragomis arba jei pavyktų įterpti į kodą galines duris (angl. Backdoor), kurios leistų apeiti jų saugumo mechanizmus. Kadangi visos duomenų bazių valdymo sistemos yra skirtingos, tarp jų ir atviro kodo, nėra vieno įsilaužimo aptikimo/prevencijos sistemos sprendimo visoms tokioms sistemoms.

Norint išspręsti **problemą**, jog universalios įsilaužimo aptikimo/prevencijos sistemos negali pilnai apsaugoti visų duomenų bazių valdymo sistemų, reikia kurti individualizuotas sistemas, pritaikytas specifinėms duomenų bazių valdymo sistemoms.

Tad šio darbo **tikslas** – suprojektuoti individualizuotą įsilaužimo aptikimo/prevencijos sistemą duomenų bazių valdymo sistemai „Firebird“. Norint pasiekti šį tikslą reikės įvykdyti šiuos **uždavinius**:

1. Išanalizuoti įsilaužimo aptikimo/prevencijos sistemų bendras technologines savybes, tipus, panaudojimo atvejus.
2. Išanalizuoti įsilaužimo aptikimo/prevencijos sistemų pritaikymo duomenų bazių valdymo sistemoms galimybes.
3. Išanalizuoti „Firebird“ duomenų bazių valdymo sistemos techninius aspektus.
4. Suprojektuoti daugiau nei vieną architektūros variantą įsilaužimo aptikimo/prevencijos sistemos „Firebird“ duomenų bazių valdymo sistemai, juos palyginti.

Atliekant šiuos uždavinius bus išanalizuota įvairi literatūra, tokia kaip patentai ir moksliniai straipsniai, aprašantys įsilaužimo aptikimo/prevencijos sistemas, siekiant išsiaiškinti pagrindinius įsilaužimo aptikimo/prevencijos sistemų tipus, kokiais metodais aptinka įsilaužimus, kokius įsilaužimo būdus aptinka, kokias turi prevencijos galimybes bei kaip tokios sistemos yra integruojamos į esamą kompiuterinį tinklą, kompiuterį.

Toliau bus analizuojama kaip įsilaužimo aptikimo/prevencijos sistemos yra pritaikomos stebėti specifinėms programų sistemoms, sutelkiant dėmesį į duomenų bazių valdymo sistemas. Bus analizuojami kitų autorių darbai, kuriuose buvo projektuojamos įsilaužimo aptikimo/prevencijos sistemos tam tikroms duomenų bazių valdymo sistemoms ar jų rūšims, siekiant išsiaiškinti, kiek projektuojami sprendimai yra individualizuojami specifiniams produktams.

Po to bus išanalizuotos „Firebird“ duomenų bazių valdymo sistemos techninės savybės. Bus išsiaiškinta kokio tipo tai duomenų bazių valdymo sistema, kokiais protokolais ji komunikuoja su kitomis programų sistemomis, kokius saugumo mechanizmus turi bei kokios yra galimos saugumo spragos ar neištaisyti defektai.

Pabaigus analizuoti literatūrą bus projektuojama įsilaužimo aptikimo/prevencijos sistema „Firebird“ duomenų bazių valdymo sistemai. Projektavimo metu bus suprojektuotas daugiau nei vienas variantas būsimos įsilaužimo aptikimo/prevencijos sistemos, kiekvienam variantui nusprendžiant, kokius metodus naudos kuriama sistema aptikti galimiems įsilaužimams, ar turės prevencinių galimybių (t.y. ar bus pasyvi ar aktyvi apsaugos sistema), jei turės, kokios bus prevencinės galimybės. Taip pat bus numatyta, kokiu būdu kuriama sistema bus integruota arba susieta su „Firebird“ duomenų bazių valdymo sistema, kokią turės įtaką jos veikimui. Pabaigus projektavimą gauti variantai bus palyginti, tiriant kokiais aspektais tie variantai yra geresni vienas už kitą ir kokiais atvejais geriau naudoti vieną ar kitą variantą.

1. ĮSILAUŽIMO APTIKIMO/PREVENCIJOS SISTEMOS

Šiame skyriuje bus išanalizuotos įsilaužimo aptikimo sistemos (1.1), įsilaužimo prevencijos sistemos (1.2), pagrindiniai panaudojimo scenarijai (1.3), technologijos (1.4), tipai (1.5) ir tipiniai komponentai (1.6).

1.1. Įsilaužimo aptikimo sistema

Šiame poskyryje bus aptarti įsilaužimo aptikimo sistemos bendrosios savybės (1.1.1) bei pagrindinės naudojamos metodologijos aptikti įsilaužimams (1.1.2).

1.1.1. Bendras aprašas

Įsilaužimo aptikimas yra procesas, kurio metu yra stebimi įvykiai atsitinkantys kompiuterinėje sistemoje arba tinkle ir analizuojantis juos, norint aptikti požymius galimų incidentų, kurie yra saugumo politikos, priimtino naudojimo politikos arba standartinių saugumo praktikų pažeidimai arba neišvengiami pavojai galimo pažeidimo. Incidentai turi daug priežasčių, tokių kaip kenkėjiškos programos (angl. Malware: kirminai, šnipinėjimo programos), užpuolikai gaunantys neautorizuotą prieigą prie sistemos iš interneto bei autorizuoti sistemos naudotoji, kurie piktnaudžiauja savo privilegijomis arba bando gauti papildomų privilegijų, kurioms jie nėra autorizuoti. Nors dauguma incidentų yra kenksmingi savo prigimtimi, tačiau dalis jų nėra, pavyzdžiui, asmuo gali neteisingai įvesti kompiuterio adresą ir atsitiktinai pabandyti prisijungti prie kitokios sistemos be autorizacijos. Įsilaužimo aptikimo sistema yra programa, kuri automatizuoja įsilaužimų aptikimo procesą. [SM07, Row02]

1.1.2. Įsilaužimo aptikimo metodologijos

Įsilaužimo aptikimo sistemos naudoja įvairias metodologijas aptikti incidentams. Bus aptartos pagrindinės klasės aptikimo metodologijų: parašu paremtos (1.1.2.1), anomalijomis paremtos (1.1.2.2) bei būseną įsimenančių protokolų analizės (1.1.2.3). Dauguma įsilaužimo aptikimo sistemų naudoja kelis aptikimo technologijas, atskirai arba integruotai, norint pateikti platesnį ir tikslesnį aptikimą.

1.1.2.1. Parašu paremtas aptikimas

Parašas yra šablonas atitinkantis žinomą grėsmę. Parašu paremtas aptikimas yra procesas, kurio metu parašai lyginami su stebimais įvykiais norint identifikuoti galimus incidentus. Parašų pavyzdžiai:

- Bandymas naudoti telnet protokolą su vartotoju „root“, kas yra organizacijos saugumo politikos pažeidimas;
- Elektroninis laiškas su tema „Free pictures!” ir failo priedu su pavadinimu „freepicks.exe“, kas yra žinomos kenkėjiškų programų charakteristikos;
- Operacinės sistemos žurnalo įrašas su būsenos kodu 645, kas parodo, kad serverio auditavimas buvo išjungtas. [SM07, RS05, Mor06]

Parašu paremtas aptikimas yra labai efektyvus aptinkant žinomas grėsmes, tačiau labai neefektyvus aptinkant anksčiau nežinomas grėsmes, grėsmes užmaskuotas slėpimo technikomis bei daugeliu žinomų atakų variacijomis. Pavyzdžiui, jei užpuolikas pakeitė kenkėjiškos programos ankstesniame pavyzdyje pavadinimą į „freepicks2.exe“, parašas ieškantis pavadinimo „freepics.exe“ jo neatitiktų. [SM07, Mor06]

Parašu paremtas aptikimas yra paprasčiausias aptikimo metodas, nes jis tiesiog lygina dabartinį aktyvumo vienetą (paketą ar žurnalo įrašą) su sąrašu parašų, naudojant simbolių eilutės palyginimo operacijas. Parašu paremto aptikimo technologijos turi mažą supratimą apie daugelį tinklo ar taikomųjų protokolų ir negali sekti ar suprasti sudėtingų komunikacijų būsenų. Pavyzdžiui, jie negali suporuoti užklausos su ja atitinkančiu atsakymu, kaip kad žinoti, jog užklausa interneto serveriui dėl tam tikro puslapio sugeneravo atsakymą su būsenos kodu 403, reiškiančiu, kad serveris atsisakė išpildyti užklausą. Jie taip pat neturi gebėjimo prisiminti ankstesnių užklausų, kai apdorojama dabartinė. Tai neleidžia parašu paremto aptikimo metodams aptikti atakų kurios susideda iš daugiau nei vieno įvykio, jei nei vienas įvykis neturi aiškaus atakos požymio. [SM07, Mor06]

1.1.2.2. Anomalijomis paremtas aptikimas

Anomalijomis paremtas aptikimas yra procesas, kurio metu lyginami apibrėžimai normaliu laikomo aktyvumo su stebimu norint aptikti reikšmingus nukrypimus. Įsilaužimo aptikimo sistema naudojanti anomalijomis paremtą aptikimą turi profilius, kurie reprezentuoja tokių dalykų kaip vartotojai, serveriai, tinklo prisijungimai ar aplikacijos normalų elgesį. Profiliai yra sukuriami stebint tipinio aktyvumo charakteristikas tam tikrą laiko periodą. Pavyzdžiui, profilis tinklui gali parodyti,

kad interneto aktyvumas sudaro vidutiniškai 13% tinklo pralaidumo ties interneto riba tipinių darbo valandų metu. Įsilaužimo aptikimo sistema tuomet naudoja statistinius metodus, kad palyginti dabartinio aktyvumo charakteristikas su slenksčiais, susijusiais su profiliu, tokiu būdu pastebint, kada interneto aktyvumas apima žymiai daugiau pralaidumo nei tikėtasi ir perspėjant administratorių apie anomaliją. Profiliai gali būti kuriami daugeliui elgsenos atributų, tokių kaip vartotojo išsiųstų elektroninių laiškų kiekis, nepavykusių bandymų prisijungti prie serverio skaičius, serverio procesoriaus panaudojimo lygis duotuoju laiko periodu. [SM07, Row02]

Didžiausia nauda anomalijomis paremto aptikimo metodų yra tai, kad jie gali būti labai efektyvūs aptinkant anksčiau nežinomas grėsmes. Pavyzdžiui, tarkime, kad kompiuteris tampa užkrėstas naujo tipo kenkėjiška programa. Kenkėjiška programa gali sunaudoti kompiuterio apdorojimo išteklius, siųsti didelius kiekius elektroninių laiškų, inicijuoti didelį kiekį tinklo prisijungimų bei atlikti kitokį elgesį, kuris smarkiai skiriasi nuo nustatyto profilio kompiuteriui. [SM07, Row02]

Pradinis profilis yra sugeneruojamas per laiko periodą (paprastai dienas, kartais savaites), kartais vadinamą mokymo periodu. Profiliai anomalijomis paremtame aptikime gali būti statiniai arba dinaminiai. Sugeneravus, statinis profilis lieka nepakitęs iki įsilaužimo aptikimo sistemai yra konkrečiai nurodyta sugeneruoti naują profilį. Dinaminis profilis yra nuolat pakoreguojamas, kai yra stebimi vis nauji įvykiai. Kadangi sistemos ar tinklai keičiasi laikui bėganti, atitinkami matavimai normalaus elgesio taip pat keičiasi, tad laikui bėgant statiniai profiliai tampa netikslūs ir turi būti periodiškai iš naujo sugeneruoti. Dinaminiai profiliai neturi tokios problemos, bet jie yra jautrūs slėpimo bandymams iš užpuolikų. Pavyzdžiui, užpuolikas gali dažnai atlikti mažą kiekį žalingos veiklos, tada lėtai didinti veiklų dažnumą ir kiekį. Jei pokyčio dažnis pakankamai mažas, įsilaužimo aptikimo sistema gali pagalvoti, kad žalinga veikla yra normali veikla ir įtraukti ją į profilį. Žalinga veikla gali taip pat būti pastebėta įsilaužimo aptikimo sistemos, kai ji kuria pradinį profilį. [SM07, Row02]

Atsitiktinai įtraukus žalingą veiklą kaip profilio dalį yra dažna problema su anomalijomis paremtais įsilaužimo aptikimo sistemų produktais. Kai kuriais atvejais administratoriai gali pamodifikuoti profilį, kad pašalinti iš profilio veiklą, kuri yra žinoma kaip žalinga. Kita problema kuriant profilius yra tai, kad kartais gali būti labai sudėtinga padaryti juos tikslius, nes skaičiavimo veikla gali būti labai sudėtinga. Pavyzdžiui, jei tam tikra priežiūros veikla, kuri atlieka didelių failų persiuntimus, įvyksta kartą į mėnesį, ji gali būti nepastebėta mokymo periodo metu, todėl įvykus gali būti palaikyta žymiu nuokrypiu nuo profilio ir sukelti įspėjimą. Anomalijomis paremti įsilaužimo

aptikimo sistemų produktai dažnai sukelia daug antros rūšies klaidų (angl. False positives) dėl gerybinės veiklos, kuri smarkiai skiriasi nuo profilio, ypač dinaminėse ar su didesne įvairove aplinkose. Kita pažymėtina problema su anomalijomis paremto aptikimo technikų naudojimu yra tai, kad neretai sudėtinga analizei nustatyti, kodėl tam tikras įspėjimas buvo sugeneruotas ir patvirtinti, kad įspėjimas yra tikslus ir nėra antros rūšies klaida, dėl įvykiu sudėtingumo ir skaičiaus įvykių, kurie galėjo sukelti įspėjimo sugeneravimą. [SM07, Row02]

1.1.2.3. Būseną įsimenančių protokolų analizė

Būseną įsimenančių protokolų analizė yra procesas, kurio metu iš anksto nubrėžti paprastai priimtinų apibrėžimų gerybinės veiklos kiekvienai protokolo būsenai profiliai lyginami su stebimais įvykiais norint identifikuoti nukrypimus. Skirtingai nuo anomalijomis paremto aptikimo, kuris naudoja serveriui ar tinklui specifinį profilį, būseną įsimenančių protokolų analizė remiasi pardavėjų sukurtais universaliais profiliais, kurie nurodo, kaip tam tikras protokolas turėtų ir neturėtų būti naudojamas. „Būseną įsimenančių“ junginyje „būseną įsimenančių protokolų analizė“ reiškia, kad įsilaužimo aptikimo sistema sugeba suprasti bei sekti būsenas tinklo, transporto ir taikomųjų protokolų, kurie turi tokią sąvoką kaip „būsena“. Pavyzdžiui, kai naudotojas pradeda failų persiuntimo protokolo (angl. File Transfer Protocol – FTP) sesiją, iš pradžių sesija yra neautentifikuotoje būsenoje. Neautentifikuoti naudotojai turėtų atlikti tik kelias komandas šioje būsenoje, tokias kaip pamatyti pagalbinę informaciją ar pateikti prisijungimo vardą ir slaptažodį. Svarbi dalis suprantant būseną yra užklausų ir atsakų poravimas, tam, kad, kai įvyksta FTP autentifikacijos bandymas, įsilaužimo aptikimo sistema gali nustatyti ar jis buvo sėkmingas randant būsenos kodą atitinkamame atsakyme. Kai naudotojas sėkmingai autentifikuojasi, sesija yra autentifikuotoje būsenoje ir tikimasi, kad naudotojas atliks bet kurią iš daugelio galimų komandų. Atliekant daugelį šių komandų esant neautentifikuotoje būsenoje yra laikoma įtartina, tačiau autentifikuotoje būsenoje jų atlikimas yra laikomas gerybiniu. [SM07]

Būseną įsimenančių protokolų analizė gali identifikuoti netikėtas komandų sekas, tokias kaip tos pačios komandos pateikimas daug kartų arba pateikimas komandos prieš pateikiant komandą, nuo kurios pirmoji yra priklausoma. Kita būseną įsimenančių protokolų analizės būsenos sekimo ypatybė yra tai, kad protokolams, kurie atlieka autentifikaciją, įsilaužimo aptikimo sistema gali sekti autentifikatorių naudojamą kiekvienoje sesijoje ir išsaugoti autentifikatorių naudojamą įtartinai veiklai. Tai padeda tiriant incidentą. Kai kurios įsilaužimo aptikimo sistemos taip pat gali naudoti

autentifikatorių informaciją apibrėžiant priimtiną veiklą skirtingai kelioms naudotojų klasėms arba specifiniams naudotojams. [SM07]

„Protokolo analizė“ atliekama būseną įsimenančių protokolų analizės metodų dažniausiai įtraukia pagrįstumo patikrinimus individualioms komandoms, tokius kaip minimalus ir maksimalus argumentų ilgis. Jei komanda paprastai turi naudotojo vardo argumentą ir naudotojo vardo maksimalus ilgis yra 20 simbolių, tada argumentas, kurio ilgis yra 1000 simbolių yra įtartinas. Jei didelis argumentas turi dvejetainių duomenų, tada jis yra dar labiau įtartinas. [SM07]

Būseną įsimenančių protokolų analizės metodai naudoja protokolų modelius, kurie paprastai yra visų pirma paremti protokolų standartais iš programinės įrangos pardavėjų bei standartizacijos įstaigų (pvz., Internet Engineering Task Force [IETF], Request For Comments [RFC]). Protokolų modeliai paprastai atsižvelgia į skirtumus kiekvieno protokolo realizacijoje. Daugelis standartų nėra išsamiai pilni paaiškinant protokolo detales, kas sukelia variacijas tarp realizacijų. Taip pat daugelis pardavėjų arba pažeidžia standartus, arba prideda patentuotų ypatybių, kurių dalis gali pakeisti ypatybes iš standartų. Patentuotiems protokolams išsamios detalės apie protokolą yra dažnai neprieinamos, dėl to įsilaužimo aptikimo sistemų technologijoms sunkiau yra atlikti išsamią, tikslią analizę. Kai pardavėjai peržiūri arba pakeičia savo protokolų realizacijas, įsilaužimo aptikimo sistemos protokolo modeliai turi būti atnaujinti, kad atspindėtų tuos pasikeitimus. [SM07]

Pagrindinis būseną įsimenančių protokolų analizės metodų trūkumas yra tai, kad jie yra labai brangūs resursams dėl analizės sudėtingumo ir valdymo išlaidų (angl. Overhead), susijusių su būsenos sekimu daugeliui tuo pat metu vykstančių sesijų. Kita svarbi problema yra tai, kad būseną įsimenančių protokolų analizės metodai negali aptikti atakų, kurios nepažeidžia paprastai priimtino protokolo elgesio charakteristikų, tokių kaip atliekant daug gerybinių veiksmų per trumpą laiko periodą norint sukelti paslaugos neigimą (angl. Denial of Service). Dar kita problema yra tai, kad protokolų modeliai naudojami įsilaužimo aptikimo sistemoje gali konfliktuoti su tuo, kaip protokolas buvo realizuotas tam tikrose versijose specifinių aplikacijų ar operacinėse sistemose arba kaip skirtingos kliento ir serverio protokolo realizacijos bendrauja tarpusavyje. [SM07]

1.2. Įsilaužimo prevencijos sistema

Šiame poskyryje bus aptarti įsilaužimo aptikimo sistemos bendrosios savybės (1.2.1) bei pagrindiniai naudojami įsilaužimo prevencijos būdai (1.2.2).

1.2.1. Bendras aprašas

Įsilaužimo prevencijos sistema yra programa, kuri turi visas įsilaužimo aptikimo sistemos galimybes bei gali bandyti sustabdyti galimus incidentus. Šios sistemos iniciatyviai identifikuoja įsibrovėlius: identifikuoja unikaliais pirmo karto atakas, remiantis atakų modifikacijomis ir tradicinėmis atakomis, padeda identifikuoti įsilaužėlius pagal ketinimų šablonus, kai jie grįžta, prisijungiant iš kito adreso, bei reaguoja į visas atakas iškart, siekiant išvengti tolimesnio įsilaužimo ir žalos. Norint išvengti žalos, įsilaužimo sistemos turi numatyti grėsmę keliančią veiklą pakankamai greitai, norint laiku imtis prevencinių priemonių. [SM07, Jac01]

1.2.2. Įsilaužimo prevencijos būdai

Įsilaužimo prevencijos sistemos siekdamos užkirsti kelią galimam pavojui naudoja keletą atsako technikų:

- Įsilaužimo prevencijos sistema sustabdo pačia ataką. To pavyzdžiai būtų tokia:
    - Nutraukia tinklo prisijungimą arba naudotojo sesiją, kuri naudojama atakai.
    - Blokuoja prieigą prie taikinio (arba galimai kitų galimų taikinių) iš nusižengiančio vartotojo paskyros, IP adreso ar kitokio užpuoliko atributo.
    - Blokuoja visas prieigas prie nusitaikyto serverio, serviso, programos ar kito resurso.
- Įsilaužimo prevencijos sistema pakeičia saugumo aplinką. Įsilaužimo prevencijos sistema gali pakeisti kitų saugumo kontrolių konfigūraciją siekiant sutrikdyti ataką. Dažnas pavyzdys yra perkonfigūravimas tinklo prietaiso (pvz., ugniasienės, maršrutizatoriaus, jungiklio) norint užblokuoti prieigą iš užpuoliko arba į taikinį bei pakeičiant taikinyje esančia serverinę ugniasienę, kad blokuotų ateinančias atakas. Kai kurios įsilaužimo prevencijos sistemos gali sąlygoti pataisymų įrašymą į serverį, jei įsilaužimo prevencijos sistema aptinka, kad serveris turi pažeidžiamumų.
- Įsilaužimo prevencijos sistema pakeičia atakos turinį. Kai kurios įsilaužimo prevencijos sistemų technologijos gali pašalinti arba pakeisti kenksmingą atakos dalį padarant ją gerybine. Paprastas pavyzdys yra įsilaužimo prevencijos sistema pašalinanti užkrėstą failą iš elektroninio laiško priedo ir tada leidžiant išvalytam elektroniniam laiškui pasiekti jo gavėją. Sudėtingesnis pavyzdys yra, kai įsilaužimo prevencijos sistema veikia kaip įgaliotinis (angl. Proxy) ir normalizuoja ateinančius prašymus, kas reiškia, kad įgaliotinis perpakuoja prašymų

naudingą apkrovą, pašalinant antraštinę informaciją. Tai gali sukelti kai kurių atakų atmetimą kaip normalizacijos proceso dalį. [SM07, Jac01]

1.3. Pagrindiniai panaudojimo scenarijai

Įsilaužimo aptikimo/prevencijos sistemos yra visų pirma sutelktos į galimų incidentų identifikavimą. Pavyzdžiui, įsilaužimo aptikimo/prevencijos sistema galėtų pastebėti, kada užpuolikas sėkmingai kompromitavo sistemą išnaudodamas joje esantį pažeidžiamumą. Įsilaužimo aptikimo/prevencijos sistema galėtų pranešti apie incidentą saugumo administratoriui, kuris galėtų greitai inicijuoti incidento atsako veiksmus, kad minimizuoti žalą, sukeltą incidento. Įsilaužimo aptikimo/prevencijos sistema taip pat galėtų įrašyti informaciją, kurią galėtų naudoti incidentų prižiūrėtojai. Daugelis įsilaužimo aptikimo/prevencijos sistemų taip pat gali būti sukonfigūruotos atpažinti saugumo politikos pažeidimus. Pavyzdžiui, kai kurios įsilaužimo aptikimo/prevencijos sistemos gali būti sukonfigūruotos su ugniasienės taisyklių rinkinio tipo nustatymais, kurie leistų joms identifikuoti tinklo srautą, kuris pažeidžia organizacijos apsaugos arba priimtino naudojimo politikas. Taip pat, kai kurios įsilaužimo aptikimo/prevencijos sistemos gali stebėti failų persiuntimus ir identifikuoti tuos, kurie gali būti įtartini, tokius kaip didelės duomenų bazės kopijavimas į naudotojo nešiojamą kompiuterį. [SM07, Roz01]

Daugelis įsilaužimo aptikimo/prevencijos sistemų taip pat gali identifikuoti žvalgybinę veiklą, kuri gali signalizuoti apie gresiančią ataką. Pavyzdžiui, kai kurie atakos įrankiai ar kenkėjiškų programų formos, ypač kirminai, atlieka žvalgybinę veiklą, tokią kaip serverio ir prievado (angl. Port) skenavimas, norint identifikuoti taikinius vėlesnėms atakoms. Įsilaužimo aptikimo/prevencijos sistema gali blokuoti žvalgybą ir perspėti saugumo administratorių, kuris galėtų imtis veiksmų, jei reikia pakeisti saugumo kontroles, kad išvengti susijusio incidento. Kadangi žvalgybos veikla yra labai dažna internete, žvalgybos aptikimas yra visų pirma dažniausiai atliekamas apsaugotiems vidiniams tinklams. [SM07]

Be incidentų identifikavimo ir pagalbinių atsako incidentams pastangų, organizacijos rado kitų įsilaužimo aptikimo/prevencijos sistemų panaudojimo būdų:

- Saugumo politikos problemų identifikavimas. Įsilaužimo aptikimo/prevencijos sistema gali pateikti tam tikrą laipsnį kokybės kontrolės saugumo politikos realizacijai, tokios kaip ugniasienės taisyklių rinkinio dublikavimas bei įspėjimų kėlimas, kai ji pastebi

tinklo srautą, kuris turėjo būti užblokuotas ugniasienės, bet nebuvo dėl ugniasienės konfigūracijos klaidos.

- Egzistuojančios grėsmės organizacijai dokumentavimas. Įsilaužimo aptikimo/prevencijos sistemos įrašo informaciją apie grėsmes, kurias aptinka. Supratimas apie atakų prieš organizacijos skaičiavimo resursus dažnumą ir charakteristikas padeda identifikuojant atitinkamas apsaugos priemones norint apsaugoti šiuos resursus. Ši informacija galėtų būti panaudota apmokant vadovybę apie organizacijai gresiančius pavojus.
- Individų atgrasymas nuo apsaugos politikos pažeidimų. Jei individai žino, kad jų veiksmai yra stebimi įsilaužimo aptikimo/prevencijos sistemos technologijų dėl saugumo politikos pažeidimų, jie yra mažiau tikėtina, kad vykdys tokius pažeidimus dėl pavojaus būti aptiktiems. [SM07, Roz01]

Dėl didėjančios priklausomybės nuo informacinių sistemų bei įsilaužimų išplitimo ir galimo poveikio tokioms sistemoms, įsilaužimo aptikimo/prevencijos sistemos tapo reikalingu priedu saugumo infrastruktūrai beveik kiekvienoje organizacijoje. [SM07]

1.4. Pagrindinės technologijos

Yra daug įsilaužimo aptikimo/prevencijos sistemos technologijų tipų, kurie daugiausiai skiriasi įvykiais, kuriuos gali atpažinti, bei metodologijomis, kurias naudoja identifikuoti incidentams. Be įvykių stebėjimo ir analizės norint identifikuoti nepageidaujamą veiklą, visų tipų įsilaužimo aptikimo/prevencijos sistemos technologijos paprastai atlieka šiuos veiksmus:

- Informacijos apie stebimus įvykius rinkimas. Informacija yra įrašoma lokaliai ir gali būti siunčiama į atskirą sistemą, tokią kaip centralizuotus įrašų serverius, apsaugos informacijos ir įvykių valdymo (angl. Security Information and Event Management – SIEM) sprendimus ir verslo (angl. Enterprise) valdymo sistemas.
- Apsaugos administratorių įspėjimas apie svarbius stebimus įvykius. Šis pranešimas, žinomas kaip perspėjimas, įvyksta per bet kurį iš metodų, įskaitant: elektroninius laiškus, gaviklio žinutes, žinutes įsilaužimo aptikimo/prevencijos sistemos vartotojo sąsajoje, paprasto tinklo valdymo protokolo (angl. Simple Network Management Protocol – SNMP) spąstus, syslog pranešimus bei naudotojo apibrėžtas programas ir skriptus. Perspėjamasis pranešimas paprastai turi tik bazinę informaciją, susijusią su įvykiu –

administratorius turi patekti į įsilaužimo aptikimo/prevencijos sistemą dėl papildomos informacijos.

- Ataskaitų parengimas. Ataskaitos apibendrina stebimus įvykius arba pateikia tam tikrų dominančių įvykių detales. [SM07, Roz01]

Kai kurios įsilaužimo aptikimo/prevencijos sistemos taip pat gali pakeisti apsaugos profilį, kai aptinkama nauja grėsmė. Pavyzdžiui, įsilaužimo aptikimo/prevencijos sistema gali surinkti detalesnę informaciją apie tam tikrą sesiją po to, kai pastebima žalinga veiklą toje sesijoje. Įsilaužimo aptikimo/prevencijos sistema taip pat gali pakeisti nustatymus, kada tam tikri perspėjimai yra sukeliami ar koks prioritetas turi būti priskiriamas tolimesniems perspėjimams po tam tikros grėsmės aptikimo. [SM07]

Kita įsilaužimo aptikimo/prevencijos sistemos technologijų dažna savybe yra tai, kad jos negali pateikti pilnai tikslaus aptikimo. Kai įsilaužimo aptikimo/prevencijos sistema neteisingai identifikuoja gerybinę veiklą kaip žalingą, įvyksta antros rūšies klaida. Kai įsilaužimo aptikimo/prevencijos sistemai nepavyksta identifikuoti žalingos veiklos, įvyksta pirmos rūšies klaida. Neįmanoma pašalinti visų pirmos ir antros rūšies klaidų, daugeliu atveju sumažinimas vienos rūšies klaidų padidina atsiradimą kitos rūšies. Daugelis organizacijų pasirenka sumažinti pirmos rūšies klaidų kiekį antros rūšies klaidų atsiradimo kaina, kas reiškia, kad daugiau žalingos veiklos yra aptinkama, bet reikia daugiau analizės resursų atskirti antros rūšies klaidoms nuo žalingos veiklos. Įsilaužimo aptikimo/prevencijos sistemos konfigūracijos pakeitimas norint pagerinti aptikimo tikslumą, vadinamas suderinimu (angl. Tuning). [SM07]

Daugelis įsilaužimo aptikimo/prevencijos sistemos technologijų taip pat siūlo ypatybes, kurios kompensuoja įprastų slėpimo technikų naudojimą. Slėpimas yra modifikavimas žalingos veiklos formato ar laiko parinkimo taip, kad jos išvaizda pasikeičia, bet efektas išlieka toks pat. Pavyzdžiui, užpuolikas galėtų koduoti teksto simbolius tokiu būdu, kokiu žino, jog taikinys supras kodavimą, ir tikintis, kad bet kokia stebėjimo įsilaužimo aptikimo/prevencijos sistema nesupras. Daugelis įsilaužimo aptikimo/prevencijos sistemos technologijų gali įveikti įprastas slėpimo technikas dublikuojant specialų apdorojimą atliekamą taikinio. Jei įsilaužimo aptikimo/prevencijos sistema „pamato“ veiklą taip pat, kaip ją matytų taikinys, tada slėpimo technikos gali paprastai būti nesėkmingos slepiant atakas. [SM07]

1.5. Pagrindiniai tipai

Yra daug įsilaužimo aptikimo/prevencijos sistemų tipų, kuriuos galima suskirstyti į šias 4 pagrindines grupes:

- Tinklo pagrindo, kurios stebi tinklo srautą dėl tam tikrų tinklo segmentų arba prietaisus ir analizuoja tinklo bei taikomųjų protokolų aktyvumą norint identifikuoti įtartiną veiklą. Ji gali identifikuoti daug skirtingų tipų domimų įvykių. Ji yra dažniausiai įdiegiama riboje tarp tinklų, tokių kaip šalia ribos ugniasienių ar maršrutizatorių, virtualių privačių tinklų (angl. Virtual Private Network – VPN) serverių, nuotolinės prieigos serverių bei bevielių tinklų.
- Bevielės, kurios stebi bevielių tinklų srautą bei analizuoją jo bevielio tinklo protokolus norint identifikuoti įtartiną veiklą, įtraukiančią pačius protokolus. Ji negali identifikuoti įtartinos veiklos taikomojo ar aukštesnio lygmens tinklo protokolų (pvz., TCP, UDP), kuriuos bevielis tinklas persiunčia. Ji dažniausiai įdiegiama organizacijos bevielio tinklo zonoje, kad stebėti jį, bet gali būti diegiama vietose, kur neautorizuoti bevieliai tinklai gali atsirasti.
- Tinklo elgsenos analizės, kurios tiria tinklo srautą, kad identifikuoti pavojus, kurie generuoja neįprastą srautą, tokį kaip paskirstyto paslaugų paneigimo (angl. Distributed Denial of Service – DDoS) atakos, tam tikrų formų kenkėjiškos programos (pvz., kirminai, galinės durys) bei politikos pažeidimai (pvz., kliento sistema teikia tinko paslaugas kitoms sistemoms). Tinklo elgsenos analizės sistemos yra dažniausiai įdiegiamos stebėti srautų organizacijos vidiniuose tinkluose ir kartais įdiegiamos ten, kur jos gali stebėti srautus tarp organizacijos tinklų ir išorinių tinklų (pvz., interneto, verslo partnerio tinklų).
- Serverio paremtos, kurios stebi vieno serverio charakteristikas bei įvykius vykstančius tame serveryje dėl įtartinos veiklos. Pavyzdžiai charakteristikų, kurias serverio paremta įsilaužimo aptikimo/prevencijos sistema gali stebėti, yra tinklo srautas (tik to serverio), sistemos žurnalas, veikiantys procesai, programų veikla, failų prieiga ir modifikavimas bei sistemos ir programų konfigūracijos pasikeitimai. Serverio paremtos įsilaužimo aptikimo/prevencijos sistemos yra dažniausiai įdiegiamos kritiniuose serveriuose, tokiuose kaip viešai prieinami serveriai ir serveriai, saugantys jautrią informaciją. [SM07, Mor06, Roz01, Sul05]

Kai kurios įsilaužimo aptikimo/prevencijos sistemos formos yra brandesnės nei kitos, nes jos egzistuoja ilgesnį laiką. Tinklo pagrindo įsilaužimo aptikimo/prevencijos sistema bei kai kurios serverio paremtos įsilaužimo aptikimo/prevencijos sistemos formos yra komerciškai prieinamos daugiau nei 10 metų. Tinklo elgesio analizės programinė įranga yra naujesnės formos įsilaužimo aptikimo/prevencijos sistema, kuri evoliucionavo iš produktų, sukurtų pagrinde aptikti paskirstytas paslaugų paneigimo atakas ir dalinai iš produktų, sukurtų stebėti srautą vidiniuose tinkluose. Bevielės technologijos yra reliatyviai naujas įsilaužimo aptikimo/prevencijos sistemos tipas, sukurtas atsakant į bevielių vietinių tinklų (angl. Wireless Local Area Network – WLAN) populiarumą bei augančias grėsmes bevieliams vietiniams tinklams ir jų klientams. [SM07]

1.6. Tipiniai komponentai

Tipiniai įsilaužimo aptikimo/prevencijos sistemos sprendinio komponentai yra tokie:

- Sensorius arba agentas. Sensoriai ir agentai stebi ir analizuoja veiklą. Terminas „sensorius“ paprastai naudojamas įsilaužimo aptikimo/prevencijos sistemoms, kurios stebi tinklus, įskaitant tinklo pagrindo, bevieles bei tinklo elgsenos analizės technologijas. Terminas „agentas“ paprastai naudojamas serverio pagrindo įsilaužimo aptikimo/prevencijos sistemos technologijoms.
- Valdymo serveris. Valdymo serveris yra centralizuotas prietaisas, kuris gauna informaciją iš sensorių ir agentų ir juos valdo. Kai kurie valdymo serveriai atlieka įvykio informacijos, kurią pateikia sensoriai ar agentai, analizę ir gali identifikuoti įvykius, kurių individualūs sensoriai ar agentai negali. Įvykių informacijos iš daugelio sensorių ar agentų susiejimas, toks kaip aptikimas įvykių, sukeltų to paties IP adreso, vadinamas koreliacija. Valdymo serveriai yra prieinami tiek kaip prietaisai, tiek kaip tik programinės įrangos produktai. Kai kurios mažos įsilaužimo aptikimo/prevencijos sistemos įdiegimas nenaudoja valdymo serverių, bet daugelis įsilaužimo aptikimo/prevencijos sistemos diegimų naudoja. Didesnėse įsilaužimo aptikimo/prevencijos sistemos diegimuose, yra dažnai keli valdymo serveriai, ir kai kuriais atvejais yra dvi pakopos valdymo serverių.
- Duomenų bazės serveris. Duomenų bazės serveris yra saugykla įvykių informacijai, užfiksuotai sensorių, agentų ir/ar valdymo serverių. Dauguma įsilaužimo aptikimo/prevencijos sistemų teikia palaikymą duomenų bazės serveriams.

- Konsolė. Konsolė yra programa, kuri pateikia įsilaužimo aptikimo/prevencijos sistemos naudotojams ir administratoriams sąsają. Konsolės programinė įranga tipiškai įrašomą į standartinį stacionarų arba nešiojamą kompiuterį. Kai kurios konsolės yra naudojamos tik įsilaužimo aptikimo/prevencijos sistemos administravimui, tokiam kaip sensorių ar agentų konfigūravimas bei programinės įrangos atnaujinimų diegimas, tuo tarpu kitos konsolės naudojamos išskirtinai stebėjimui ir analizei. Kai kurios įsilaužimo aptikimo/prevencijos sistemos konsolės pateikia tiek administravimo tiek stebėjimo galimybių. [SM07, Mor06]

Įsilaužimo aptikimo/prevencijos sistemos komponentai gali būti sujungti vienas su kitu per organizacijos standartinius tinklus arba per atskirą tinklą išskirtinai suprojektuotą saugumo programinės įrangos valdymui, žinomą kaip valdymo tinklas. Jei naudojamas valdymo tinklas, kiekvienas sensoriaus ar agento serveris turi papildomą tinklo sąsają, vadinamą valdymo sąsaja, kuri prisijungia prie valdymo tinklo. Taip pat, kiekvienas sensoriaus ar agento serveris negali parsiųsti jokio srauto tarp valdymo sąsajos ir jo bet kurio kito tinklo sąsajos. Valdymo serveris, duomenų bazės serveris, ir konsolės yra prijungtos tik prie valdymo tinklo. Ši architektūra efektyviai izoliuoja valdymo tinklą nuo produkcijos tinklo. To nauda yra įsilaužimo aptikimo/prevencijos sistemos egzistavimo ir tapatybės paslėpimas nuo užpuolikų, norint apsaugoti pačią įsilaužimo aptikimo/prevencijos sistemą nuo atakos ir užtikrinti, kad įsilaužimo aptikimo/prevencijos sistema turi pakankamą pralaidumą (angl. Bandwidth) funkcionavimui net ir nepageidaujamomis sąlygomis (pvz., kirmino ataka ar paskirstytas paslaugų paneigimas stebimuose tinkluose). Valdymo tinklo naudojimo trūkumai apima papildomas išlaidas tinklo įrangai bei kitai techninei įrangai (pvz., kompiuteriams konsolėms) ir nepatogumus įsilaužimo aptikimo/prevencijos sistemos naudotojams ir administratoriams dėl skirtingų kompiuterių naudojimo įsilaužimo aptikimo/prevencijos sistemos valdymui ir stebėjimui. [SM07]

Jei įsilaužimo aptikimo/prevencijos sistema yra įdiegiama be atskiro valdymo tinklo, kitas būdas pagerinti įsilaužimo aptikimo/prevencijos sistemos apsaugą yra sukurti virtualų valdymo tinklą naudojant virtualų lokalų tinklą (VLAN) standartiniame tinkle. Virtualaus lokalaus tinklo naudojimas suteikia apsaugą įsilaužimo aptikimo/prevencijos sistemos komunikacijoms, tačiau ne tiek pat kiek atskiras valdymo tinklas. Pavyzdžiui, netinkamas virtualaus lokalaus tinklo sukonfigūravimas gali vesti link įsilaužimo aptikimo/prevencijos sistemos duomenų demaskavimo. Kitas rūpestis yra tai, kad nepageidaujamomis sąlygomis, tokiomis kaip paskirstyta paslaugų paneigimo ataka ar didelio kenkėjiškos programos incidento, tinklo prietaisai, kuriais dalinasi organizacijos pagrindiniai tinklai,

ir virtualus lokalus tinklas gali tapti visiškai prisotintu, neigiamai paveikiant įsilaužimo aptikimo/prevencijos sistemos prieinamumą ir darbą. [SM07]

2. ĮSILAUŽIMO APTIKIMO/PREVENCIJOS SISTEMŲ PRITAIKYMAS DUOMENŲ BAZĖMS

Šiame skyriuje bus nagrinėjama, kokios technikos naudojamos aptikti įsilaužimams duomenų bazių valdymo sistemose (2.1), kokios yra atakų prieš duomenų bazes rūšys (2.2) bei su kokiais sunkumais susiduria įsilaužimo aptikimas duomenų bazėse (2.3).

2.1. Įsilaužimo aptikimo duomenų bazėse technikos

Yra siūloma keletas duomenų bazių įsilaužimo aptikimo technikų tipų. Šiame poskyryje bus aptarti: laiko analizė (2.1.1), priklausomybių ir sąryšių analizė (2.1.2), sekos lyginimo analizė (2.1.3), priklausomybių integracija į sekų lyginimo analizę (2.1.4), statistinė analizė (2.1.5), informacijos teorinė analizė (2.1.6), komandų šablono analizė (2.1.7).

2.1.1. Laiko analizė

Šios technikos susitelkia į laiko ypatybes, tokias kaip laiko tarpas tarp naudotojo veiksmų ar šių veiksmų trukmė. Tai naudoja vidurkio ir standartinio nuokrypio modelį, pagamintą iš laiko parašų norint patikrinti išskirtis iš anksto apibrėžtame diapazone realaus laiko duomenų bazių sistemose. Šis sprendimas transakciją laiko skaitymo ir/ar rašymo veiksmų kiekvienam duomenų objektui aibe, kuri yra vykdoma iš anksto nustatytais atnaujinimo periodais. [SBV14]

Pavyzdžiui, laiko duomenų objekto (įvykio) atnaujinimas gali iššaukti taisyklę, dėl kurios atnaujinimo laikas palyginamas su tikėtinu atnaujinimo laiku (sąlyga) ir atnaujinimas yra atmetamas (veiksmas), jei predikatas grąžina klaidą, laikant tai įsilaužimu. Mokymo periodas vyksta tol, kol reikšmingas vidurkis su 99 % pasikliovimo lygio normalia distribucija yra pasiekiami kiekvienai objekto/atnaujinimo porai. Duomenų bazės elgsena yra stebima sensorių transakciniame lygmenyje, jie laikomi mažo dydžio ir turi nustatytą semantiką, tokią kaip tik rašymo operacijos ir tiksliai apibrėžtus duomenų prieigos šablonus. Jei transakcija bando atnaujinti laiko objektą, kuris jau buvo atnaujintas tame periode, yra iškeliamas perspėjimas. [SBV14]

2.1.2. Priklausomybių ir sąryšių analizė

Įsilaužimo aptikimo technikos, besiremiančios priklausomybių ir sąryšių analize, skaičiuoja priklausomybes ir/ar ryšius tarp skirtingų naudotojo veiksmų aibių ir/ar prieitų duomenų, norint išsiaiškinti, kurie stulpeliai, eilutės, lentelės ir t.t. ir/ar komandos yra dažniausiai vykdomos ar apdorojamos kartu. [SBV14]

DEMIDS (angl. Detection of Misuse in Database Systems - piktnaudžiavimo duomenų bazių sistemose aptikimas) sistema kuria naudotojų profilius remiantis jų veikla, nustatant dažnus objektų rinkinius iš ypatybės/reikšmės porų ir apskaičiuoja atstumo matmenis tarp naudotojo veiklos ir išmokto dažnų objektų rinkinių, kad aptikti įsilaužimus, turint slenkstį. Ypatybės paprastai remiasi sintaksine analize naudotojo komandų, kur objektų rinkinių domenas yra aibė atributų, išduotų kartu. [SBV14, CGL00]

Kitas požiūris naudojantis dažną objektų rinkinių gavybą (angl. Mining) apibendrina kiekvieną naudotojo komandą kaip kortežą <OP, F, T, C>, kur OP yra SQL komandos tipas (insert, select ir t.t.), F yra atributų aibė, T yra lentelių aibė ir C yra suvaržytų sąlygų aibė. Algoritmas išgauna naudotojo užklausų profilius naudojant šiuos kortežus, remiantis pateiktų užklausų šablonais transakciniame lygyje. Algoritmas adaptuoja palaikymą ir pasitikėjimo asociacijos taisyklės gavybą pridedant užklausų struktūrą ir atributų ryšius į skaičiavimus. [SBV14]

Dar kitas požiūris siūlo rolėmis paremtą priėjimo kontrolės duomenų bazių įsilaužimo aptikimo/prevencijos sistemą naudojančią rinkinius apibendrinti kiekvieno naudotojo komandai. Nagrinėjant paprastą komandą SELECT {Tikslų/Taikinių sąrašas} FROM {Santykių sąrašas} WHERE {Sąlyga}, rinkinys yra apibrėžiamas kaip (C, PER, PA, SR, SA), kur C yra SQL komanda (insert, select ir t.t.), PER yra projekcijos/santykio informacija, PA yra projekcijos/atributo informacija, R yra pasirinkimo/santykio informacija ir SA yra pasirinkimo/atributo informacija. [SBV14, BKT+05, KTB08]

Autorių apibrėžiami trys rinkinių tipai su skirtingu detalumu: turint santykį (kitaip lentelę) R1 su atributais A1, B1, C1, D1 ir santykį R2 su atributais A2, B2, C2, D2 ir naudotojo komandą SELECT R1.A1, R1.C1, R2.B2, R2.D2 FROM R1, R2 WHERE R1.B1=R2.B2, jie sugeneruos šiurkštų c-rinkinį (SELECT,<2>,<4>,<2>,<2>), vidutinį m-rinkinį (select,<1,1>,<2,2>,<1,1>,<1,1>) bei gerą f-rinkinį (select,<1,1>,<[1,0,1,0],[0,1,0,1]>,<1,1>,<[0,1,0,0],[0,1,0,0]>). [SBV14, BKT+05, KTB08]

Anomalijų aptikimui, kai duomenų bazė turi rolėmis paremtus naudotojus (t.y. galima susieti kiekvieną naudotojo veiksmą su duota role), NBC (angl. Naïve Bayes Classifier) klasifikatorius naudojamas taip: visoms užklausoms audito žurnaluose ir kiekvienai rolei klasifikatorius kiekvieno

tipo rinkiniui yra sukuriamas klasifikatorius (mokymo fazė). Kiekvienai pateiktai užklausai, jei bet kuris jos klasifikatorius yra kitoks nei tie, kurie apibrėžti rolei, veiksmas yra laikomas įsilaužimu ir sugeneruojamas perspėjimas (testavimo fazė). [SBV14]

Jei rolėmis paremta prieigos politika nėra realizuota duomenų bazėje, jos siūlo neprižiūrimą anomalijų aptikimą. Šiuo atveju pozicijos ir atstumo funkcijos yra apibrėžiamos kiekvienam rinkiniui ir klasterizavimo technikos (k-centrinė ir k-vidurkio) atvaizduoja kiekvieną naudotoją į jį reprezentuojantį klasterį, kuris yra klasteris su didžiausiu skaičiumi mokymo įrašų tam naudotojui po klasterizavimo fazės (testavimo fazės). Norint ištestuoti kiekvieną naują užklausą, galima naudoti du būdus: 1) turint nustatymą jį reprezentuojančio klasterio, naudojamas NBC kaip ir rolėmis paremto anomalijų aptikimo atlikti panašiems testams, arba 2) patvirtinti, ar nauja užklausa yra statistinis nukrypimas naudojat MAD (angl. Median of Absolute – absoliuti mediana) testą, kuris, jei tiesa, veiksmą laiko įsilaužimu ir sugeneruoja įspėjimą. [SBV14]

2.1.3. Sekos lyginimo analizė

Sekos lyginimas daugiausia susideda iš bendrų įvykių sekų nustatymo (tokių kaip komandos, duomenų atributai, prieitos reikšmės ir t.t.). Duomenų bazių įsilaužimo aptikimo sistemos naudojančios šio tipo technikas paprastai išmoksta ir identifikuoja reikšmingo ilgio pasikartojančias įvykių sekas bei galiausiai suskaido jas į mažesnio dydžio poaibius, kad paženklinti ar klasifikuoti tas sekas ir jų poaibius kaip normalų naudotojo elgesį. Aptikimo fazėje kiekviena naujų įvykių seka yra lyginama su išmoktomis naudotojų sekomis ir jų poaibiais, kad pamatuoti, kiek jie skiriasi norint įvertinti jos tikimybę būti įsilaužimu. [SBV14]

Vienas iš sprendimų yra identifikuoti prieitų atributų, komandų ar lentelių sekas statant naudotojų profilius. Siūlomos savybės yra komandų tipai (insert, select, update ir t.t.), jautriais nurodyti atributai, visi atributai, operacijos atributai bei visų ypatybių mišiniai. Šis sprendimas taip pat apibrėžia kriterijus renkantis tarp naudotojo paremto, role paremto ar organizacija paremto profilio, turint duomenų bazės darbinį kontekstą. Mokymo fazėje, jis stato sekų modelius turint slenkstį, apibrėžianti maksimalų skirtumų skaičių. Aptikimo fazėje jis naudoja slenkstį apskaičiuojant didžiausią leidžiamą skaičių skirtumų tarp testavimo sekos ir tų, kurios buvo išsaugotos mokymo fazėje, kad įvertinti sekas kaip normalias arba nenormalias. [SBV14]

2.1.4. Priklausomybių integracija į sekų lyginimo analizę

Vienas iš požiūrių yra apie priklausomybių ryšių tarp transakcijos lygio atributų su aukštu palaikymu ir pasikliovimo taisyklėmis radimas. Jis laiko, kad visada, kai atributas yra atnaujinamas, šis veiksmas yra susiejamas su seka kitų įvykių, užfiksuotų duomenų bazėje (pavyzdžiui, dėl duoto atributo atnaujinimo kiti atributai taip pat yra perskaitomi ar įrašomi). Todėl, kiekvienas atnaujinimas yra apibrėžiamas kaip 3 aibės: skaitymo aibė (aibė atributų, kurie buvo perskaityti dėl atnaujinimo), iki rašymo aibė (aibė atributų, kurie buvo įrašyti prieš atnaujinimą ir dėl jo) bei po rašymo aibė (aibė atributų, kurie buvo įrašyti po atnaujinimo ir kaip jo pasekmė). Transakcijos, kurios nesilaiko jokių gavybos duomenų priklausomybių taisyklių yra laikomos žalingomis. [SBV14]

Kitas požiūris atsižvelgia į atributų jautrumą, t.y. kiekvienam atributui suteikia svarbumo matą. Jis siūlo tris atributų jautrumo lygius, remiantis jų palaikymu analizuojamose transakcijose: aukštas, vidutinis ir žemas. Svoriais paremtas duomenų gavybos algoritmas naudojamas išgauti priklausomybėms tarp duomenų bazės atributų ir generuoti taisykles, kurios atspindi šias priklausomybes, turint pamatuotas operacijų (skaitymo, rašymo) sekas ir kiekvieno atributo jautrumą. Bet kuri transakcija, kuri nesilaiko šių taisyklių, identifikuojama kaip žalinga. Šis požiūris taip pat išplečia Esybių-Ryšių (E-R) modelį, norint sintaksiškai pažymėti atributų jautrumą. [SBV14, SSM06a, SSM06b]

Dar vienas požiūris siūlo mokymo algoritmą, kuris vaizduoja transakcijas kryptiniais grafais, apibūdinančias vykdymo kelius. Naujų transakcijų aibės, kurios skiriasi nuo išmoktų kelių yra laikomos neautorizuotomis SQL komandų sekomis. Ypatybės naudojamos statyti vykdymo keliams yra komandos tipas (select, insert, delete ir t.t.), tiksliniai objektai (lentelės), pasirinkti stulpeliai ir apribojimų atributai, kurių visi yra gaunami iš tipinių duomenų bazės valdymo sistemos audito įrašų, saugančių informaciją apie vartotoją (UserID), sesiją (SessionID), komandą (CommandID), transakciją (TransactionID), naudotojo komandą, objekto savininką bei vykdymo laiko žymę. [SBV14]

2.1.5. Statistinė analizė

Statistinė analizė yra naudojama keliose duomenų bazių įsilaužimo aptikimo sistemose skaičiuojant naudotojo veiklos statistiką. Vienas požiūris naudoja statistines funkcijas nuorodinių reikšmių, gautų iš santykių duomenų (kitaip lentelių) delta ryšių (pasikeitimai reikšmių stebimuose objektuose/atributuose visoms nuorodinėms reikšmėms per atributą tarp dviejų duomenų bazių

įsilaužimo aptikimo sistemos vykdymų), anomalijų aptikimui. Išplėtimas apibrėžiamas kaip aibė visų duomenų įterpimo/modifikacijos eilučių bei santykis reiškia lentelę ar vaizdą (angl. View). Nuorodinės reikšmės apima kiekį, minimumą, maksimumą, vidurkį, standartinį nuokrypį, diapazoną, apskaičiuotus santykius, nulinio ilgio tikrinimą bei bitų skaičiavimą. Taip pat įeina piktnaudžiavimo aptikimo metodas, kuris tikrina duomenų bazės objektus (duomenų bazė, funkcija, indeksas, privilegija, procedūra, taisyklė, schema, statistikos, lentelė, trigeris ir vaizdas) bei visas jų operacijas. Tai padaroma iš anksto apibrėžiant, jei pora <Duomenų bazės objektas, operacija> yra pavojinga ar ne. [SBV14, SL05]

Kitas požiūris remiasi skaičiavimu apibendrintų statistikų, tokių kaip kiekio, maksimumo, minimumo, vidurkio, medianos, standartinio nuokrypio bei kardinalumo reikšmės kiekvienam duomenų rinkinio atributui, kuris yra rezultatas arba paveiktas kiekvieno naudotojo veiksmų. Šios statistikos yra saugomos nustatytos dimensijos vektoriuje, vadinamame S-vektoriumi, nesvarbu kokia didelė galėtų būti komandos rezultato duomenų aibė. Kai duomenų aibė, iš kurios gaunamas S-vektorius, yra didelė, yra siūloma atrinkti mėginius iš duomenų aibės paimant pirmus pradinius k kortežų norint išlaikyti efektyvumą bei plečiamumą. Aibė kiekvieno naudotojo S-vektorių yra naudojama atliekant technikas, tokias kaip klasterizavimas, naivus Bayes (angl. Naïve Bayes), pagalbinės vektorių mašinos ar sprendimų medžiai norint gauti modelį, kuris atvaizduoja naudotojo normalų elgesį duotam S-vektoriui. Įsilaužimo aptikimo fazėje statistinis nukrypimas bei išskirčių patvirtinimas yra atliekamas, norint patikrinti kiekvieną naudotojo komandą ir klasifikuoti ją kaip normalią ar nenormalią. [SBV14, MPN+10]

2.1.6. Informacijos teorinė analizė

Požiūriai naudojantys informacijos teorinę analizę skaičiuoja matavimus, tokius kaip entropija ir informacijos nauda charakterizuojant naudotojų profilius ir palyginant juos su vėlesniųjų komandų, norint pamatyti, kiek šie skiriasi nuo originaliųjų. [SBV14]

Vienas iš variantų yra toks, kad ypatybės susideda iš audito duomenų kortežo su n kintamųjų kiekvienam duomenų objektui (pvz., įvykių tipai, tokie kaip komandų sąrašas), kur kiekvienas įrašas vaizduoja klasę. Kuo mažesnė entropija, tuo mažesnis skaičius skirtingų įrašų (t.y. didesnis skaičius dublikatų), kas nurodo reguliaresnius audito duomenų rinkinius. Faktas, jog daug įvykių yra pasikartojantys (arba identiški) duomenų rinkinyje, nurodo, kad šie įvykiai gali pasikartoti ateityje.

Anomalijų aptikimo modelis gaminamas iš duomenų rinkinio su maža entropija bus paprastesnis ir turės geresnį aptikimo efektyvumą. [SBV14]

Sąlyginė entropija yra naudojama apibrėžiant audito duomenų laikų sekas. H(X|Y) parodo kiek neapibrėžtumo lieka likusiems audito įvykiams sekoje X po to kai peržiūrima Y. Anomalijų aptikimui tai naudojama kaip nuoseklių priklausomybių reguliarumo matas. Jei audito seka yra seka tokio paties tipo įvykių, tada sąlyginė entropija yra lygi 0 ir įvykių sekos yra deterministinės. Atvirkščiai, didelė sąlyginė entropija parodo, kad sekos yra ne tokios deterministinės ir dėl to sudėtingesnės modeliavimui. [SBV14]

Santykinė sąlyginė entropija tarp distribucijų yra naudojama apskaičiuoti reguliarumams (atstumams) tarp dviejų audito duomenų sekų, kur mokymo duomenų rinkinys yra patvirtintas audito duomenų rinkinys ir testuojamas duomenų rinkinys yra tas, kuris bus tikrinamas. Dar kartą, geriausias sprendimas yra tas, kuris turi mažesnę santykinę sąlyginę entropiją. Informacijos nauda yra supažindinama norint palengvinti ypatybių pasirinkimą bei statymo procesą, kad pagerinti aptikimo efektyvumą dėl jos tiesioginio ryšio su sąlygine entropija. Kuo aukštesnė informacijos nauda, priklausanti ypatybei, tuo mažesnė sąlyginė entropija ir, taip pat, geresnis aptikimo efektyvumas. [SBV14]

2.1.7. Komandų šablono analizė

Komandų modeliavimo duomenų bazių įsilaužimo aptikimo sistema naudoja komandų žurnalą analizuojant visas reguliarias naudotojo komandas ir statant apibendrintą šabloną, kuris bendrai reprezentuoja tipinę naudotojo darbo apkrovą. [SBV14]

Vienas iš siūlomų algoritmų apibendrina aibę tariamų „teisėtų“ užklausų į SQL šablonus, kurie reprezentuoja visų užklausų modelius. Kiekvienas sąlyginio filtravimo kintamasis WHERE išlygoje panašiose komandose yra laikomas parametru. Norint pamatyti, ar nepriklausomas kintamasis arba baigtinis sąrašas reikšmių turėtų būti naudojamas kiekvienam parametrui, atliekamas Kolmogorov-Smirnov testas su 90 % pasikliovimo lygiu. Algoritmas taip pat tabuliuoja kiekvieno išmokto bruožo dažnumą, t.y. kaip dažnai jis pasirodo aibėje SQL sakinių. [SBV14]

Imant naują bruožą F ir anksčiau apibrėžtą bruožą F‘, F yra laikomas teisėtu jei F skiriasi nuo F‘ jei: 1) papildoma sąlyga WHERE išlygoje esanti F, bet kurios nėra F‘, ir prijungta su AND operatoriumi; 2) F pasirenka lygų ar mažesnį skaičių stulpelių nei F‘. Šis algoritmas taip pat siūlo metodą išvesti trūkstamiems bruožams (t.y. diapazonai užklausų, panašių į duomenų bazės žurnalo

užklausas, naudotas mokymo fazėje), remiantis galimų WHERE išlygos sąlygų kombinacijos iš anksčiau įgytų bruožų maišymo. Testavimo fazėje, kiekviena komanda, reikšmingai besiskirianti nuo apskaičiuotų bruožų, yra laikoma nenormalia. [SBV14]

Kitame požiūryje autoriai siūlo taikyti gramatika paremtą analizę, naudojančią medžio branduoliu paremtą mašinų mokymo techniką vietoje paprastai naudojamų vektoriumi paremtų duomenų. Šis požiūris naudoja SQL nagrinėjimo medžio struktūrą koreliuojant komandas su programomis ir diferencijuojant tarp gerybinių ir žalingų, tikrinant pasikeitimus komandų sintaksės medžiuose. Jie išveda atstumo matą sukeltą medžio branduolio funkcijos išmatuoti panašumui tarp SQL komandų naudojant jų nagrinėjimo medžius. Palaikymo vektoriaus mašinos yra naudojamos mokymo fazėje ir klasterizavimas yra naudojamas atskyrimui gerybinių nuo žalingų veiklų su išskirčių aptikimu. Šis metodas skatina kontekstui jautrų panašumą, kuris leidžia rasti artimiausią pageidaujamą komandą žalingam sakiniui, kas padeda priežasties analizėje. [SBV14, BAM09]

2.2. Atakų prieš duomenų bazes rūšys

Atsižvelgiant į įsilaužėlių ketinimus, yra trys pagrindinės atakų prieš duomenų bazes rūšys:

- Atakos, skirtos sugadinti duomenis (vientisumo atakos). Šio tipo atakose įsilaužėlis siekia prieigos prie duomenų bazės, norint atlikti veiksmus, kurie sukompromituotų jos vientisumą, tokios kaip duomenų sugadinimas ar ištrynimas duotame duomenų bazės objekte (pavyzdžiui, pakeitimas lentelės turinio);
- Atakos, skirtos pavogti informaciją (konfidencialumo ataka). Šiose atakose įsilaužėlis susitelkia į konfidencialumo pažeidimą, t.y. pavogti verslo informaciją vietoje to, kad ją sugadinti;
- Atakos, skirtos padaryti duomenų bazę nepasiekiamą (prieinamumo atakos). Šios atakos siekia padaryti duomenų bazės paslaugas neprieinamas naudotojams, t.y. jo yra pagrinde paslaugos neigimo (angl. Denial of Service) atakos (pavyzdžiui, užtvindinimas duomenų bazės paslaugų bei srauto dideliu kiekiu užklausų, užlaužiant duomenų bazės serverio instancijas, ištrinant duomenų bazės objektus it t.t.). [SBV14]

2.3. Įsilaužimo aptikimo duomenų bazėse sunkumai

Norint apsaugoti duomenų bazes, kurios saugo įmonių kritinę informaciją ir paslaptis, nuo įsilaužimų yra plačiai naudojamos duomenų bazių įsilaužimo aptikimo sistemos. Dauguma duomenų bazių įsilaužimo aptikimo sistemų negali užkirsti kelio naudotojo veiksmams prieš baigiantis jų vykdymui. [SBV14]

Dauguma duomenų bazių įsilaužimo aptikimo sistemų susitelkia į naudotojo komandų sintaksės analizę (t.y. išnagrinėja užklausų SQL išraiškų sintaksę statant naudotojų profilius). Dažniausios su tuo susijusios problemos:

- Įprastos naudotojo užklausos gali plačiai skirtis sintaksiškai, nors ir pateikia „normalų“ (t.y. gerą pageidaujamą (angl. Non-intrusive)) rezultatą, kuris generuoja antros rūšies klaidas (t.y. netikrus pavojus);
- Užpuoliko užklausos gali būti kuriamos taip, kad menkai sintaksiškai skirtųsi nuo „normalaus“ naudotojo elgesio profilio, bet pateiktų „nenormalius“ (t.y. žalingus ir nepageidaujamus) rezultatus, kurie generuoja pirmos rūšies klaidas (t.y. atakos, kurios praeina nepastebėtos). [SBV14]

Dėl SQL kalbos ekspresyvumo ir poreikio nustatyti užklausos ekvivalentiškumą ar panašumą, sintaksinė analizė yra sudėtinga ir sunkiai atliekama teisingai. Užklausų izoliavimas ir ekvivalentiškumas yra NP-pilni jungiamosioms užklausoms ir neapibrėžti užklausoms su neigimais. [SBV14]

Duomenų bazėse, kur tipinės naudotojų darbo apkrovos turi gerai apibrėžtą skaičių skirtingų komandų, kurios yra kviečiamos pakartotinai, remiantis komandų sintaksine analize gali būti pasiekiamas aukštas įsilaužimo aptikimo efektyvumas. Tai dažniausiai įvyksta transakcinėse sistemose. Tačiau, analitinėse sistemose, tokiose kaip duomenų sandėliai (angl. Data Warehouse), daugelis veiksmų yra ekspromtiniai (angl. Ad hoc) ir turi kintančius vykdymo laikus su kintančiais duomenų priėjimo šablonais ir dimensijų dydžio dažniais, dėl to yra dažniausiai nenuspėjami ir plataus spektro. Tai padaro atskyrimą normalios komandos nuo nenormalios duomenų sandėliuose ekstremaliai sudėtinga užduotimi. Tokiose analitinėse duomenų bazėse, apribojimas įsilaužimo aptikimo iki komandų sintaksės analizės paprasčiausiai modeliuojant SQL komandų šablonus ar statinius dažnus duomenų prieigos šablonus (pvz., kurios lentelės ar stulpeliai prieinami) yra nepatikima, arba, bent jau, minimalistiška. [SBV14]

Atsižvelgiant į duomenų sandėlio naudotojų darbo apkrovos charakteristikas, įsilaužimo aptikimo sprendimai, besiremiantys laiko analize yra netinkami ir dažniausiai pateikia labai prastus įsilaužimo aptikimo rezultatus dėl darbo apkrovų nenuspėjamo dažnio ir vykdymo laiko. Dėl daugumos darbo apkrovų ekspromtinės prigimties, įsilaužimo aptikimo sprendimai, kurie remiasi komandų šablono analize, neturi reikiamo dinamiško efektyviai atlikti įsilaužimo aptikimo procesų ir dėl to taip pat pateikia prastus įsilaužimo aptikimo rezultatus. Transakcinėse sistemose, laiko analizė yra labai efektyvi, kai naudotojų veiksmai įvyksta gerai apibrėžtuose laiko perioduose ir turi nuspėjamus vykdymo laikus. Kitu atveju, ji nukenčia su laiko analize dėl to pačių problemų kaip ir duomenų sandėliai. [SBV14]

Nors vienas iš siūlomų požiūrių prideda į duomenis centruota analizę kiekvienos naudotojo komandos vykdymo rezultato duomenų aibei, analizė yra besiremianti faktais vykdymo atžvilgiu. Turint laiko periodą tarp įsilaužimo pradžios ir jo aptikimo, kartu su resursų panaudojimu ir pasirinktų duomenų jautrumo, dauguma įmonių gali patirti didelių nuostolių, jei įsilaužimas pažeidžia sistemos prieinamumą arba paviešina verslo paslaptis, jei duomenų bazių įsilaužimo aptikimo sistema per ilgai užtrunka, kad perspėtų apie žalingą įsilaužimą arba negali išvengti arba nutraukti jo vykdymo. Šiuo atveju, besiremiantys faktais požiūriai neefektyvūs sprendimai įsilaužimo aptikimui tiek transakcinėse duomenų bazėse, tiek duomenų sandėliuose. [SBV14]

Galutinai, duomenų saugyklų naudotojų darbo apkrovų nenuspėjamas vykdymo dažnumas ir ekspromtiškumas padaro laiku paremtą ir SQL modeliavimo įsilaužimo aptikimo požiūrius netinkamus. Alternatyviai, duomenų bazių aptikimo sistemos, atliekančios įsilaužimo aptikimą grubiu pagrindu, tokiu kaip duomenų bazės sesijos ar transakcijos komandų aibė, vietoje tinkamo pagrindo, tokio kaip analizuoti kiekvieną SQL komandą, rizikuoja tuo, kad seka žalingų komandų gali būti įvykdytos prieš susidorojant su įsilaužimu. Todėl, duomenų priklausomybių ir sekų lyginimo požiūriai, kurie gali patikrinti kiekvieną naudotojo komandą prieš jos įvykdymą, bet tik kai reikšmingas kiekis komandų buvo įvykdytas, turėtų būti atsargiai naudojami, atsižvelgiant į kiekvienos duomenų bazė kontekstą. [SBV14]

Duomenų centruotos technikos gali pridėti vertės prieš faktinėms įsilaužimo aptikimo sistemoms atliekant po faktinę analizę duomenų, kuriuos paveikė naudotojo veiksmas. Kombinuojant šias technikas su duomenų prieigos šablonų analizės technikomis, kurios numano apdorotus duomenis, yra labiausiai tinkama ir efektyvi duomenų bazių įsilaužimo aptikimo sistema abiejų tipų duomenų bazėms. [SBV14]

3. „FIREBIRD“ DUOMENŲ BAZIŲ VALDYMO SISTEMA

Šiame skyriuje bus aprašyti duomenų bazių valdymo sistemos „Firebird“ bendras aprašymas ir techniniai aspektai (3.1) ir šiuo metu esantys pažeidžiamumai (3.2).

3.1. Bendras aprašymas ir techniniai aspektai

„Firebird“ yra reliacinė duomenų bazių valdymo sistema, siūlanti daug ANSI SQL standarto savybių ir veikianti Linux, Windows ir įvairiose UNIX platformose. „Firebird“ siūlo gerą konkurencingumą, aukštą efektyvumą ir stiprų kalbos palaikymą saugomoms procedūroms ir trigeriams. Ji yra naudojama produkcinėse sistemose įvairiais vardais nuo 1981 metų. [FOW]

„The Firebird Project“ yra komerciškai nepriklausomas C ir C++ programuotojų, techninių patarėjų ir palaikytojų projektas, kuriantis ir tobulinantis daugelio platformų reliacinę duomenų bazių valdymo sistemą, remiantis išeities kodu, išleistu Inprise Corp (dabar žinoma kaip Borland Software Corp) 2000 metų liepos 25 dieną. [FOW]

„Firebird“ palaiko didelį skaičių techninės ir programinės įrangos platformų: Windows, Linux, MacOS, HP-UP, AIX, Solaris ir kitus. Ji veikia ant x386, x64, PowerPC, Sparc ir kitų techninės įrangos platformų bei palaiko lengvą mechanizmų migraciją tarp šių platformų. [FOW]

Viena iš pagrindinių „Firebird“ savybių yra jos kelių kartų architektūra, kuri įgalina hibridinių OLTP ir OLAP aplikacijų kūrimą. Tai padaro „Firebird“ duomenų bazę galinčią tuo pat metu tarnauti kaip analitinei ir operacinei duomenų saugyklai, nes skaitytojai neblokuoja rašytojų, kai prieinami tie patys duomenys daugeliu atvejų. [FOW]

„Firebird“ pateikia išsamų SQL92 palaikymą:

- Aukštas suderinamumas su ANSI SQL;
- Dažnos lentelių išraiškos (angl. Common Table Expressions – CTE);
- Lankstus transakcijų valdymas;
- Pilnai išvystytos saugomos procedūros (angl. Stored procedures);
- Kelių duomenų bazių užklausos;
- Aktyvių lentelių koncepcija ir įvykiai;
- Vartotojo apibrėžtos funkcijos (angl. User Defined Functions). [FOW]

„Firebird“ siūlo pėdsako (angl. Trace) API bei turtingą rinkinį stebėjimo lentelių (MON$):

- Realaus laiko stebėjimas;

- SQL derinimas;
- Auditas:
    - Įvykių;
    - Dalinis arba pilnas registravimas;
    - Per nuotolinį prisijungimą. [FOW]

„Firebird“ palaiko tokias apsaugos priemones:

- Naudotojai ir rolės;
- GRANT/REVOKE taikymas pagrindinėms operacijoms;
- Duomenų bazės savininko koncepcija;
- Vienas prisijungimas (angl. Single Sign-On – SSO) galutiniams naudotojams;
- Integracija su Windows domeno / aktyvios direktorijos apsauga;
- Vienas atviras prievadas (įprastai 3050, tačiau konfigūruojama);
- Pseudonimai (angl. Aliases) paslepia kelią iki duomenų bazės. [FOW]

Duomenų bazių limitai:

- Maksimalus duomenų bazės dydis: 32 TB;
- Maksimalus skaičius lentelių: 32768;
- Maksimalus vienos lentelės dydis: apie 18 TB;
- Maksimalus dydis išorinio lentelės failo: neribotas;
- Maksimalus skaičius eilučių lentelėje: $> 2^{40}$;
- Maksimalus eilutės dydis: 64 KB;
- Maksimalus duomenų bazės puslapio dydis: 16 KB;
- Maksimalus skaičius indeksų lentelėje: keli šimtai, priklausomai nuo duomenų bazės puslapio dydžio ir indeksų tipo (vieno segmento ar sudėtiniai);
- Maksimalus dydis (visas plotis) indekso rakto: 4 KB;
- Maksimalus skaičius indeksų duomenų bazėje: neribotas. [FOW]

3.2. Pažeidžiamumai

Su naujausiomis „Firebird“ versijomis susiję pažeidžiamumai:

- TraceManager „Firebird“ 2.5.0 ir 2.5.1 versijose, kai įjungiami pėdsakai, leidžia nuotoliniam autentifikuotam naudotojui sukelti paslaugos neigimą (NULL rodyklės

nurodiklinimas ir užlūžimas) paruošiant tuščią dinaminę SQL užklausą. Tai gali sukelti sumažėjusį efektyvumą arba resursų prieinamumo trukdžius.

- Steku paremto buferio perpildymas „Firebird“ versijose nuo 2.1.3 iki 2.1.5 prieš 18514 ir 2.5.1 iki 2.5.3 prieš 26623, Windows aplinkoje leidžia nuotoliniam užpuolikui įvykdyti arbitrišką kodą iš pagaminto paketo TCP 3050 prievadui, susijusį su trūkstamu dydžio patikrinimu išgaunant grupės numerį iš CNCT informacijos. Tai gali sukelti žymų informacijos atskleidimą, modifikavimą kai kurių sisteminių failų ar informacijos, sumažėjusį efektyvumą arba resursų prieinamumo trukdžius.
- „Firebird“ 2.5.5 leidžia nuotoliniam autentifikuotam naudotojui sukelti paslaugos neigimą (tarnybos užlaužimą) naudojant paslaugų valdytoją, iškviečiant gbak paslaugų programą su negaliojančiu parametru. Tai gali sukelti sumažėjusį efektyvumą arba resursų prieinamumo trukdžius. [FSW]

4. ĮSILAUŽIMO APTIKIMO/PREVENCIJOS SISTEMOS ARCHITEKTŪRA

Šiame skyriuje bus aprašytos dvi galimos įsilaužimo aptikimo/prevencijos sistemos realizacijos: pasyvi įsilaužimo aptikimo sistema (4.1), aktyvi įsilaužimo prevencijos sistema (4.2) bei atliktas jų palyginimas (4.3).

4.1. Įsilaužimo aptikimo sistemos architektūra

Šiame poskyryje bus aprašyta pasyvi įsilaužimo aptikimo sistema: jos veikimas (4.1.1), administratoriaus atliekamos užduotys (4.1.2), sistemos statinis modelis (4.1.3) ir dinaminis modelis (4.1.4).

4.1.1. Veikimo aprašymas

Įsilaužimo aptikimo sistema bus apsaugos sistema pasyviai stebinti ir tirianti ar nėra galimų įsilaužimo bandymų į „Firebird“ duomenų bazių valdymo sistemą. Šios sistemos pagrindiniai komponentai bus:

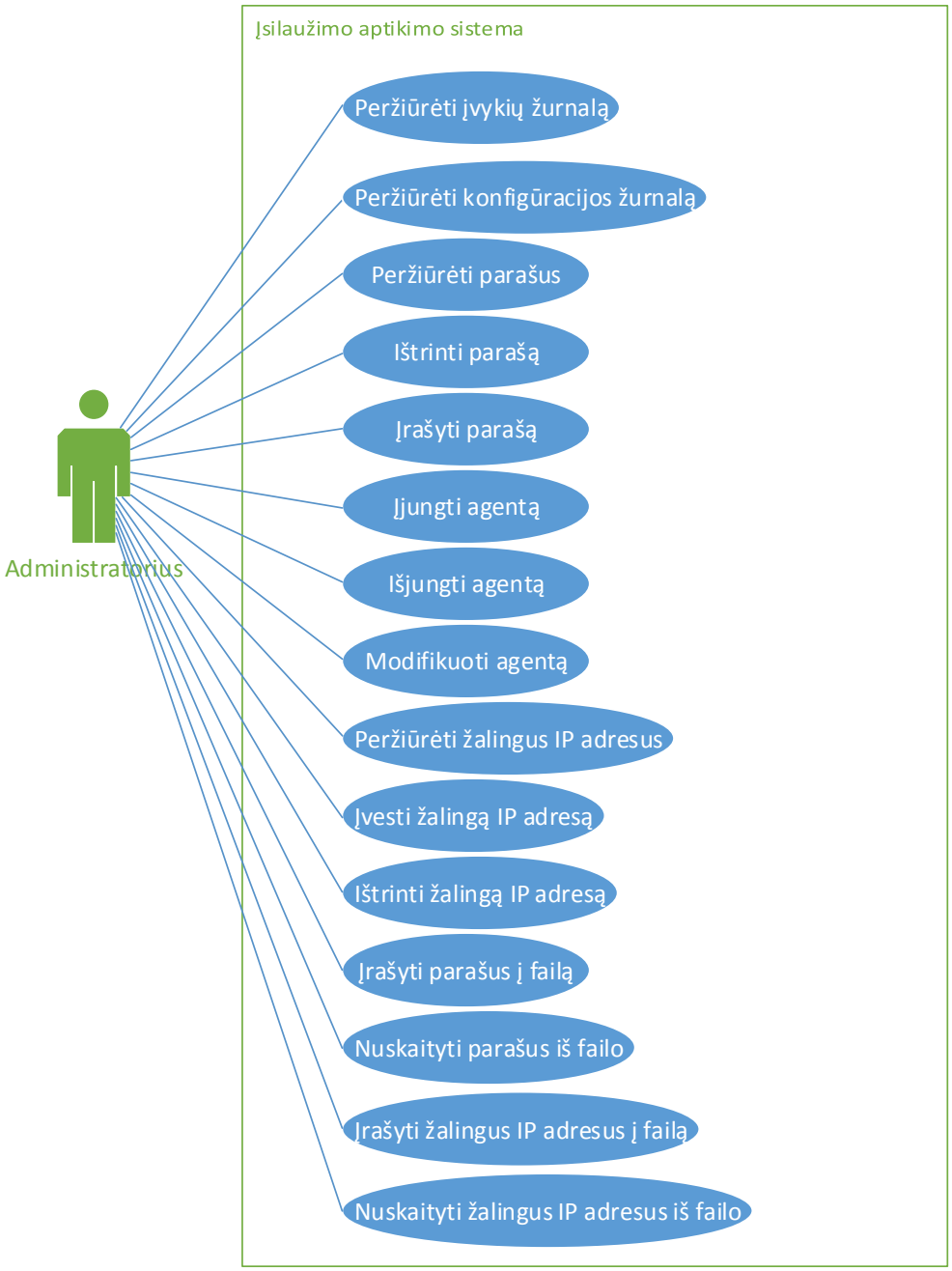
- Agentas – sistemos dalis, kuri pasyviai skaito tinklo srauto, tenkančio duomenų bazių valdymo sistemai, paketus ir tiria, ar nebuvo įsilaužimo bandymų;
- Valdymo serveris – pagrindinė sistemos dalis, kuri valdys agentą, kontroliuos visų kitų komponentų veikimą;
- Parašų saugykla – saugykla atliekanti parašų saugojimą, naujų pridėjimą ir esamų pašalinimą;
- Žurnalų valdytojas – sistemos dalis valdanti visus jos žurnalus: įvykių, konfigūracijos pakeitimų;
- Konsolė – tekstinė sąsaja administratoriams, skirta valdyti įsilaužimo aptikimo sistemai.

Agentas naudodamas paketų pagavimą skaitys ir analizuos kopijas paketų tinklo srauto, skirto duomenų bazių valdymo sistemai. Agentas bus pasyvus, kadangi daryti įtakos tinklo srautui negalės. Agentas analizuos gautus paketus naudodamas parašu paremtą aptikimą pagal parašus, saugomus parašų saugykloje. Aptikus galimą įsilaužimą – paketą atitinkantį nors vieną parašą ar gautą iš žinomo žalingo IP adreso, agentas išsaugos detalią informaciją apie jį įvykių žurnale. Konsolė leis administratoriui valdyti, kada agentas turi būti įjungtas, kokio prievado (angl. Port) klausyti, valdyti

naudojamus parašus, valdyti žalingų IP adresų sąrašą, peržiūrėti žurnalus. Žurnalų valdytojas registruos įvairius konfigūracijos pasikeitimus į konfigūracijos žurnalą.

4.1.2. Užduotys ir jų vykdymo scenarijai

Žemiau pateikta įsilaužimo aptikimo sistemos panaudojimo atvejų diagrama (1 pav. Įsilaužimo aptikimo sistemos panaudojimo atvejų diagrama).



1 pav. Įsilaužimo aptikimo sistemos panaudojimo atvejų diagrama

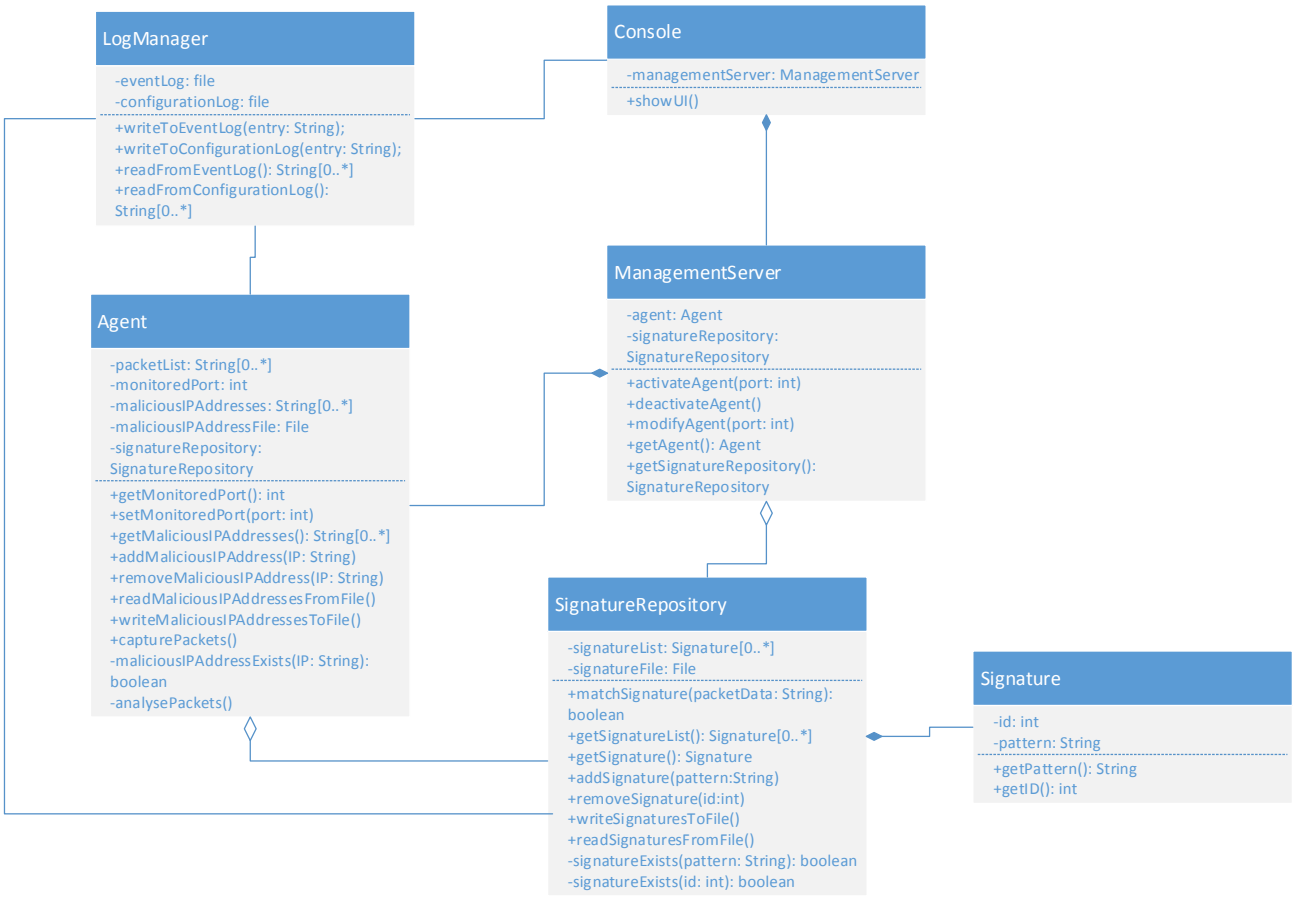
Pagrindinės administratoriaus atliekamos užduotys:

1. Peržiūrėti įvykių žurnalą. Administratorius naudodamasis konsole peržvelgia įvykių žurnalą, kuriame registruojami visi aptikti įsilaužimo bandymai su informacija apie juos (iš kokio adreso atėjo paketas, kokią komandą/duomenis laikė paketas ir t.t.).
2. Peržiūrėti konfigūracijos žurnalą. Administratorius naudodamasis konsole peržvelgia konfigūracijos žurnalą, kuriame saugomi agento nustatymų bei parašų saugyklos pasikeitimai.
3. Peržiūrėti parašus. Administratorius naudodamasis konsole peržvelgia parašų saugykloje esančius parašus.
4. Ištrinti parašą. Administratorius naudodamasis konsole ištriną parašą su nurodytu identifikacijos numeriu.
5. Įrašyti parašą. Administratorius naudodamasis konsole sukuria naują parašą, kuriam automatiškai sugeneruojamas identifikacijos numeris.
6. Įjungti agentą. Administratorius naudodamasis konsole įjungia agentą, kad šis skaitytų paketus, skirtus duomenų bazių valdymo sistemai.
7. Išjungti agentą. Administratorius naudodamasis konsole išjungia agentą, kad šis nebeskaitytų jokių paketų.
8. Modifikuoti agentą. Administratorius naudodamasis konsole pakeičia prievadą, kurio klausosi agentas.
9. Peržiūrėti žalingus IP adresus. Administratorius naudodamasis konsole peržiūri sąrašą užregistruotų žalingų IP adresų.
10. Įvesti žalingą IP adresą. Administratorius naudodamasis konsole įveda naują žalingą IP adresą.
11. Ištrinti žalingą IP adresą. Administratorius naudodamasis konsole ištrina žalingą IP adresą.
12. Įrašyti parašus į failą. Administratorius naudodamasis konsole nurodo parašų saugyklai visus turimus parašus įrašyti į failą.
13. Nuskaityti parašus iš failo. Administratorius naudodamasis konsole nurodo parašų saugyklai nuskaityti parašus iš failo.
14. Įrašyti žalingus IP adresus į failą. Administratorius naudodamasis konsole nurodo agentui visus turimus žalingus IP adresus įrašyti į failą.

15. Nuskaityti žalingus IP adresus iš failo. Administratorius naudodamasis konsole nurodo agentui nuskaityti žalingus IP adresus iš failo.

4.1.3. Statinis sistemos modelis

Žemiau pateikiama įsilaužimo aptikimo sistemos klasių diagrama (2 pav. Įsilaužimo aptikimo sistemos klasių diagrama).



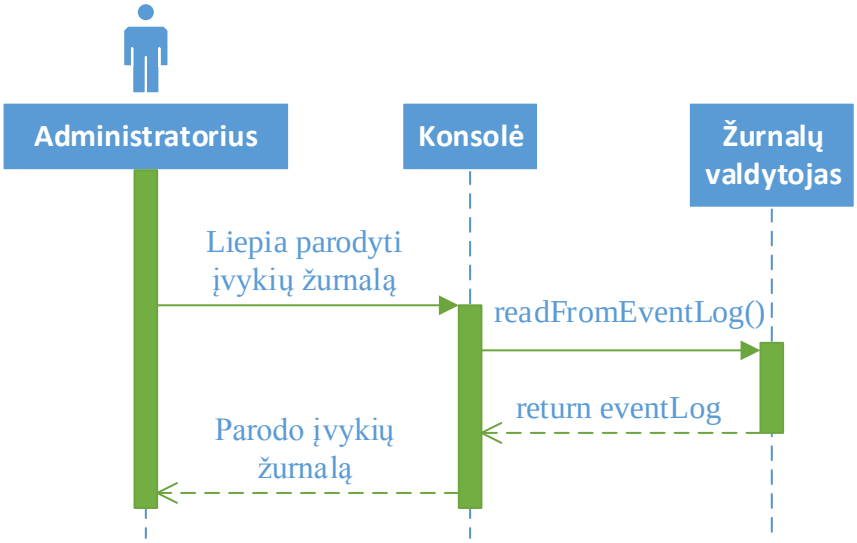
2 pav. Įsilaužimo aptikimo sistemos klasių diagrama

4.1.4. Dinaminis sistemos modelis

Žemiau pateikiamos sekų diagramos, atitinkančios kiekvieną panaudojimo scenarijų:

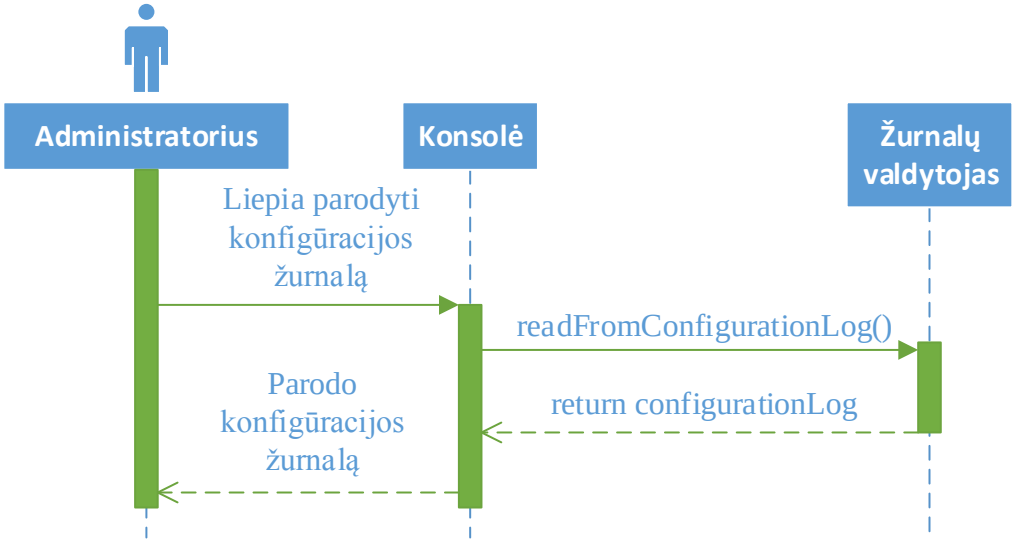
1. Peržiūrėti įvykių žurnalą (3 pav. Įvykių žurnalo peržiūrėjimas);
2. Peržiūrėti konfigūracijos žurnalą (4 pav. Konfigūracijos žurnalo peržiūrėjimas);
3. Peržiūrėti parašus (5 pav. Parašų peržiūrėjimas);
4. Įrašyti parašą (6 pav. Naujo parašo pridėjimas);

5. Ištrinti parašą (7 pav. Parašo ištrynimas);
6. Įjungti agentą (8 pav. Agento įjungimas);
7. Išjungti agentą (9 pav. Agento išjungimas);
8. Modifikuoti agentą (10 pav. Agento modifikavimas);
9. Peržiūrėti žalingus IP adresus (11 pav. Žalingų IP adresų peržiūrėjimas);
10. Įvesti žalingą IP adresą (12 pav. Naujo žalingo IP adreso įvedimas);
11. Ištrinti žalingą IP adresą (13 pav. Žalingo IP adreso ištrynimas);
12. Įrašyti parašus į failą (14 pav. Parašų įrašymas į failą);
13. Nuskaityti parašus iš failo (15 pav. Parašų nuskaitymas iš failo);
14. Įrašyti žalingus IP adresus į failą (16 pav. Žalingų IP adresų įrašymas į failą);
15. Nuskaityti žalingus IP adresus iš failo (17 pav. Žalingų IP adresų nuskaitymas iš failo).



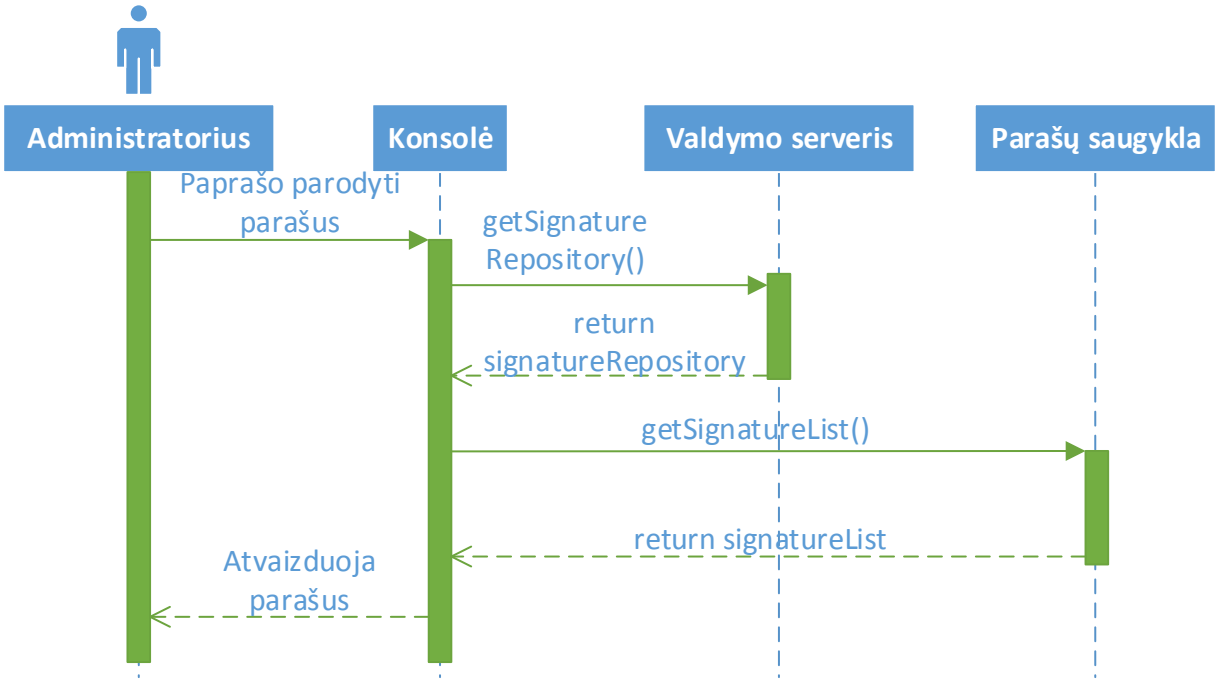
3 pav. Įvykių žurnalo peržiūrėjimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori peržiūrėti įvykių žurnalą. Konsolė kreipiasi į žurnalų valdytojo readFromEventLog() metodą, kuris konsolei grąžina iš failo nuskaitytą įvykių žurnalą kintamajame eventLog. Konsolė tada atvaizduoja administratoriui kintamojo eventLog turinį.



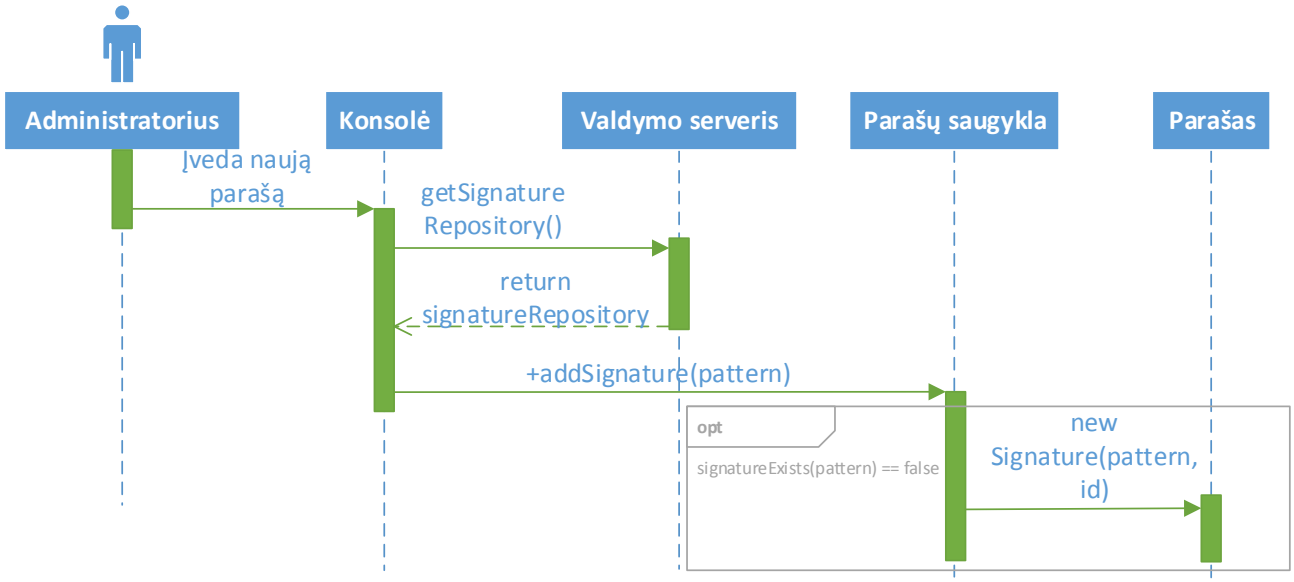
4 pav. Konfigūracijos žurnalo peržiūrėjimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori peržiūrėti konfigūracijos žurnalą. Konsolė kreipiasi į žurnalų valdytojo readFromConfigurationLog() metodą, kuris konsolei grąžina iš failo nuskaitytą konfigūracijos žurnalą kintamajame configurationLog. Konsolė tada atvaizduoja administratoriui kintamojo configurationLog turinį.



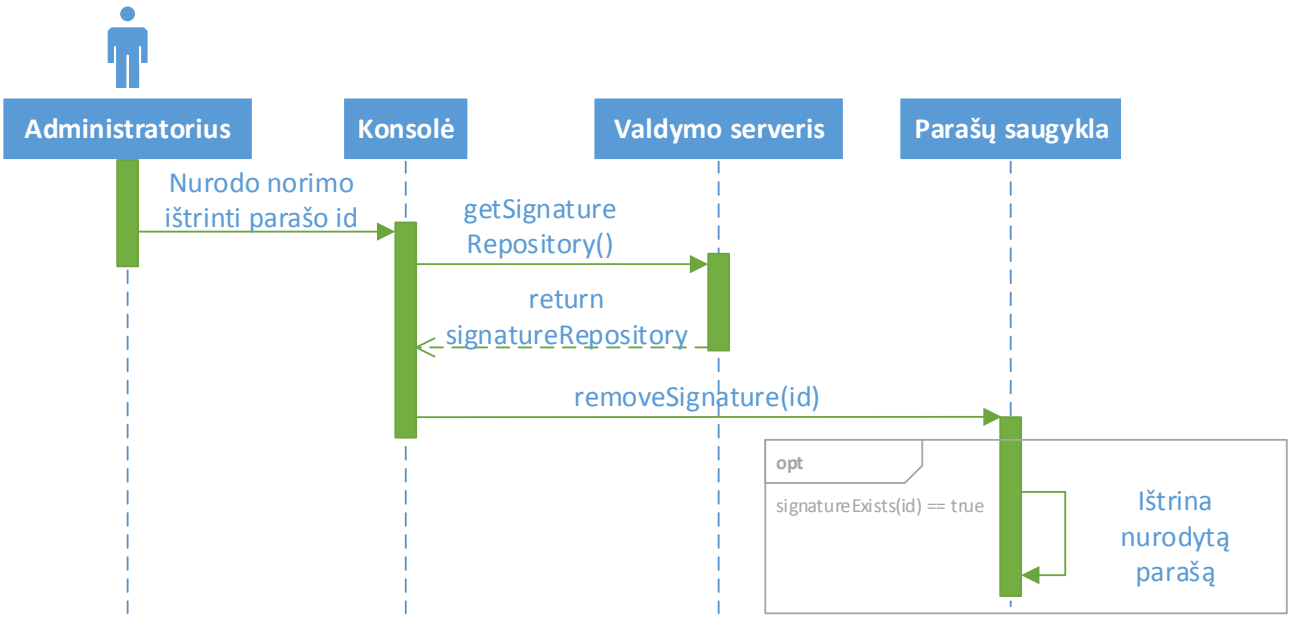
5 pav. Parašų peržiūrėjimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori peržiūrėti įvestus parašus. Konsolė kreipiasi į valdymo serverio getSignatureRepository() metodą, kuris konsolei grąžina parašų saugyklą kintamajame signatureRepository. Tuomet konsolė kreipiasi į parašų saugyklos getSignatureList() metodą, kuris konsolei grąžiną sąrašą visų įvestų parašų kintamajame signatureList. Galiausiai konsolė administratoriui atvaizduoja kintamojo signatureList turinį.



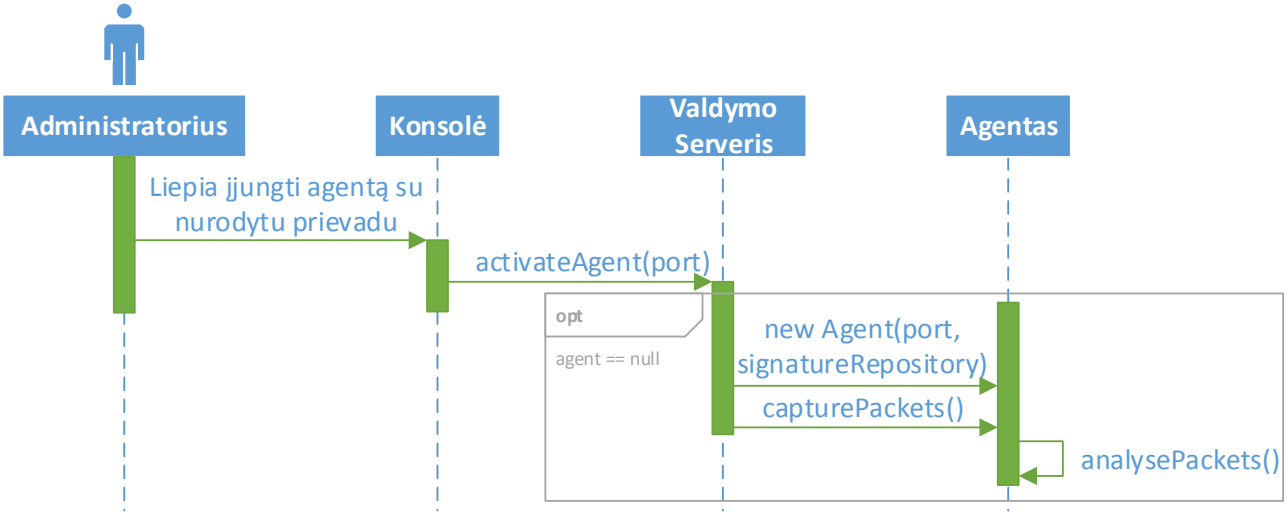
6 pav. Naujo parašo pridėjimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori įvesti naują parašą ir įveda jo šabloną. Konsolė kreipiasi į valdymo serverio getSignatureRepository() metodą, kuris konsolei grąžina parašų saugyklą kintamajame signatureRepository. Tuomet konsolė kreipiasi į parašų saugyklos addSignature(pattern) metodą, kur kaip argumentas pattern pateikiamas administratoriaus įvestas šablonas. Parašų saugykla patikrina, ar nėra parašo su pateiktu šablonu. Jei tokio parašo nėra, tuomet sukuriamas naujas parašas su pateiktu šablonu ir jam sugeneruojamas naujas identifikacijos numeris.



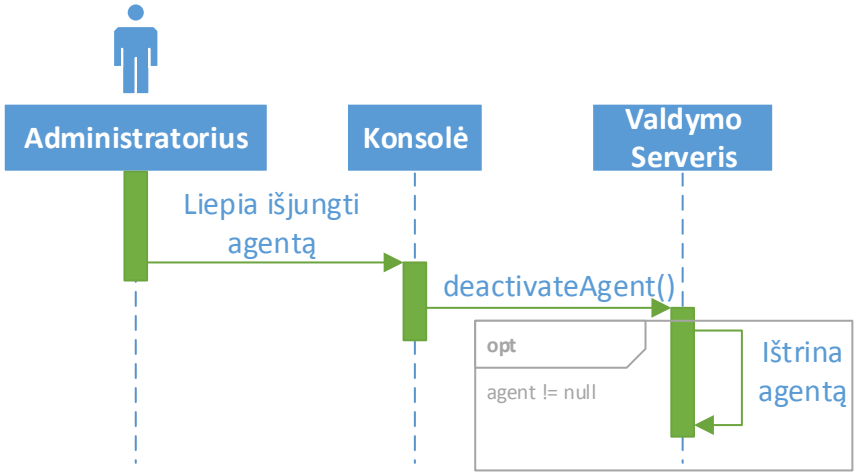
7 pav. Parašo ištrynimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori ištrinti esamą parašą ir įveda parašo identifikacijos numerį. Konsolė kreipiasi į valdymo serverio gerSignatureRepository() metodą, kuris konsolei grąžina parašų saugyklą kintamajame signatureRepository. Tuomet konsolė kreipiasi į parašų saugyklos removeSignature(id) metodą, kur kaip parametras id pateikiamas administratoriaus įvestas identifikacijos numeris. Parašų saugykla patikrina, ar egzistuoja parašas su nurodytu identifikacijos numeriu. Jei toks parašas egzistuoja, tuomet nurodytas parašas yra ištrinamas.



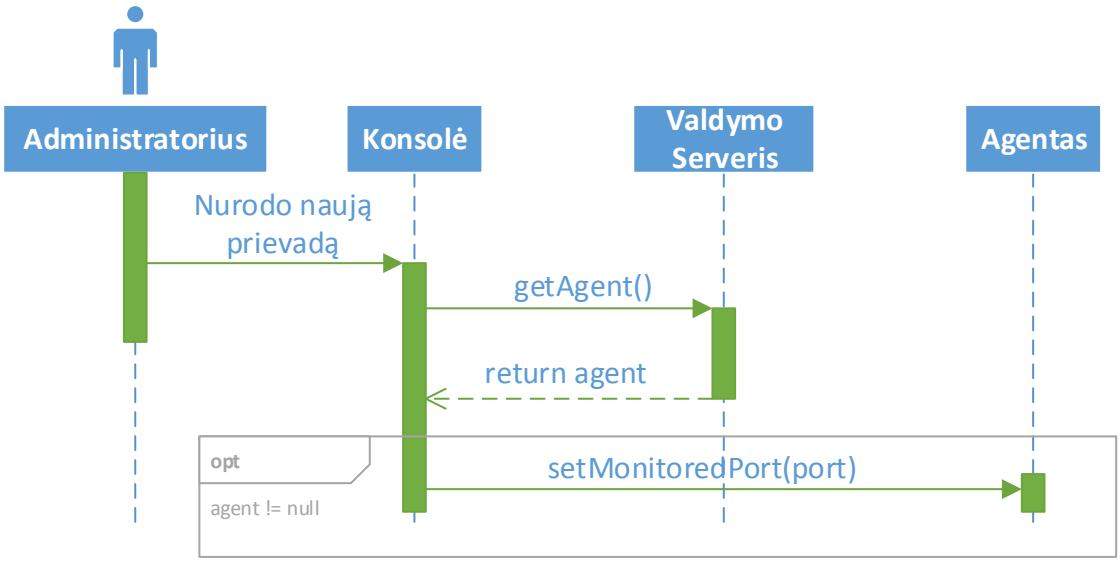
8 pav. Agento įjungimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori įjungti agentą ir įveda prievadą, kurio nori, kad agentas klausytųsi. Konsolė iškviečia valdymo serverio activateAgent(port) metodą, kur kaip argumentas port pateikiamas administratoriaus įvestas prievadas. Valdymo serveris patikrina ar agentas egzistuoja. Jei agentas neegzistuoja, tuomet valdymo serveris sukuria naują agentą su nurodytu prievadu ir aktyvuoja agento capturePackets() metodą, kuris aktyvuoja agento paketų klausymą ir iškviečia analysePackets() metodą, kuris vykdo pastebėtų paketų analizę.



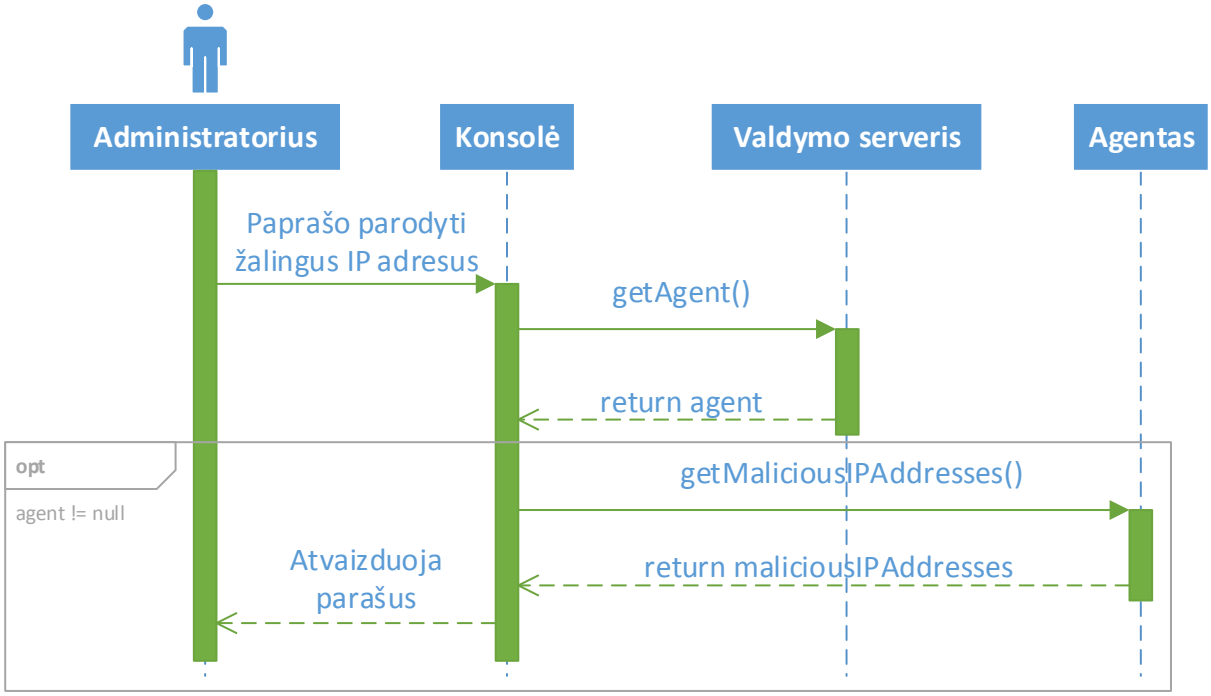
9 pav. Agento išjungimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori išjungti agentą. Konsolė iškviečia valdymo serverio deactivateAgent() metodą, kuris, jei agentas egzistuoja, ištrina agentą.



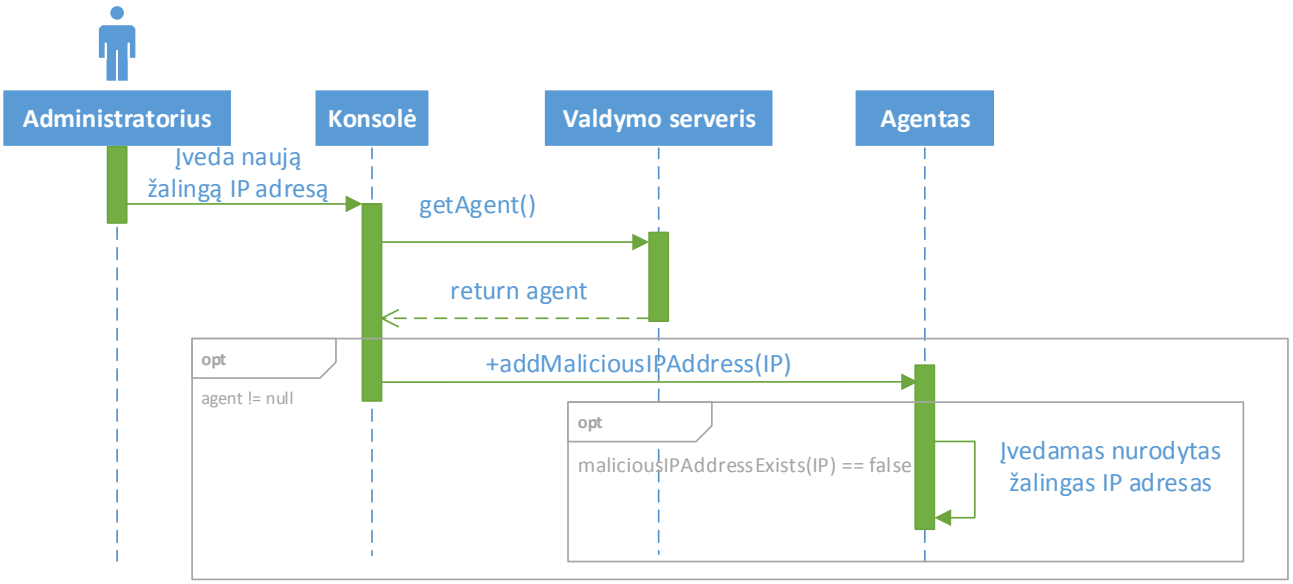
10 pav. Agento modifikavimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori modifikuoti agentą ir nurodo naują prievadą. Konsolė iškviečia valdymo serverio getAgent() metodą, kuris grąžina agentą kintamajame agent. Jei grąžintas kintamasis netuščias (jo reikšmė nėra lygi null), tuomet konsolė iškviečia agento setMonitoredPort(port) metodą, kur kaip argumentas port pateikiamas administratoriaus nurodytas naujas prievadas. Agentui tuomet nustatomas naujas prievadas.



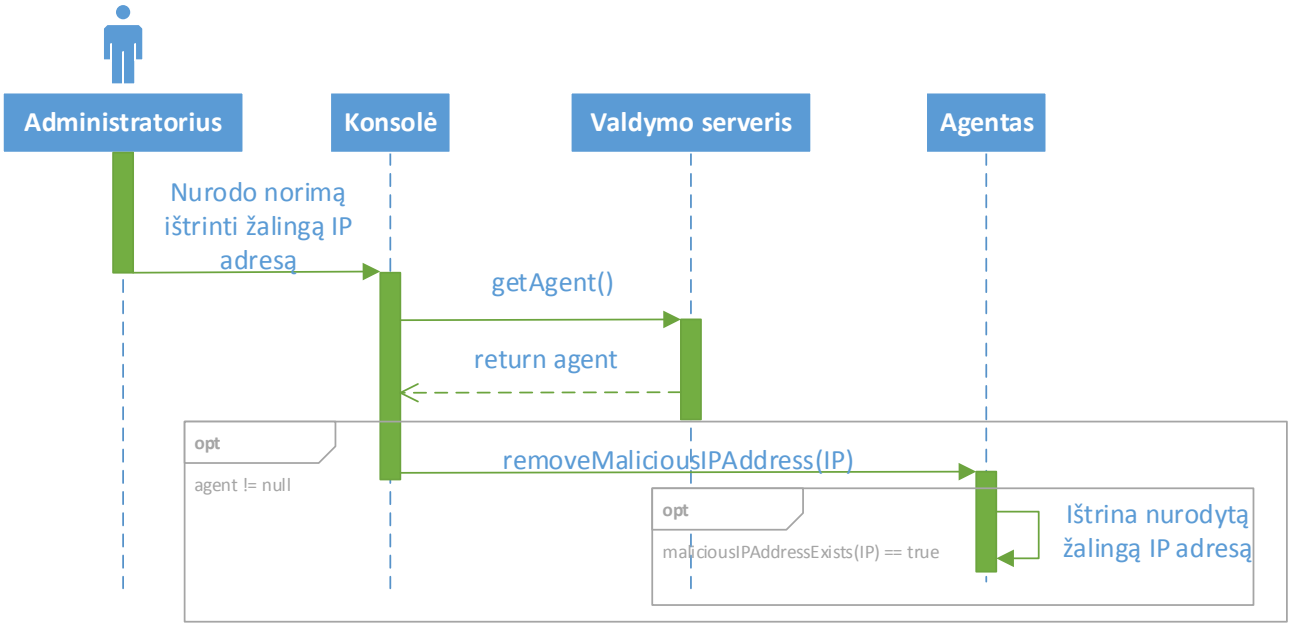
11 pav. Žalingų IP adresų peržiūrėjimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori peržiūrėti esamus žalingus IP adresus. Konsolė iškviečia valdymo serverio getAgent() metodą, kuris grąžina agentą kintamajame agent. Jei grąžintas kintamasis netuščias (jo reikšmė nėra lygi null), tuomet konsolė iškviečia agento getMaliciousIPAddresses() metodą, kuris grąžina visų įvestų žalingų IP adresų sąrašą kintamajame maliciuosIPAddresses. Tuomet konsolė atvaizduoja administratoriui kintamojo maliciousIPAddresses turinį.



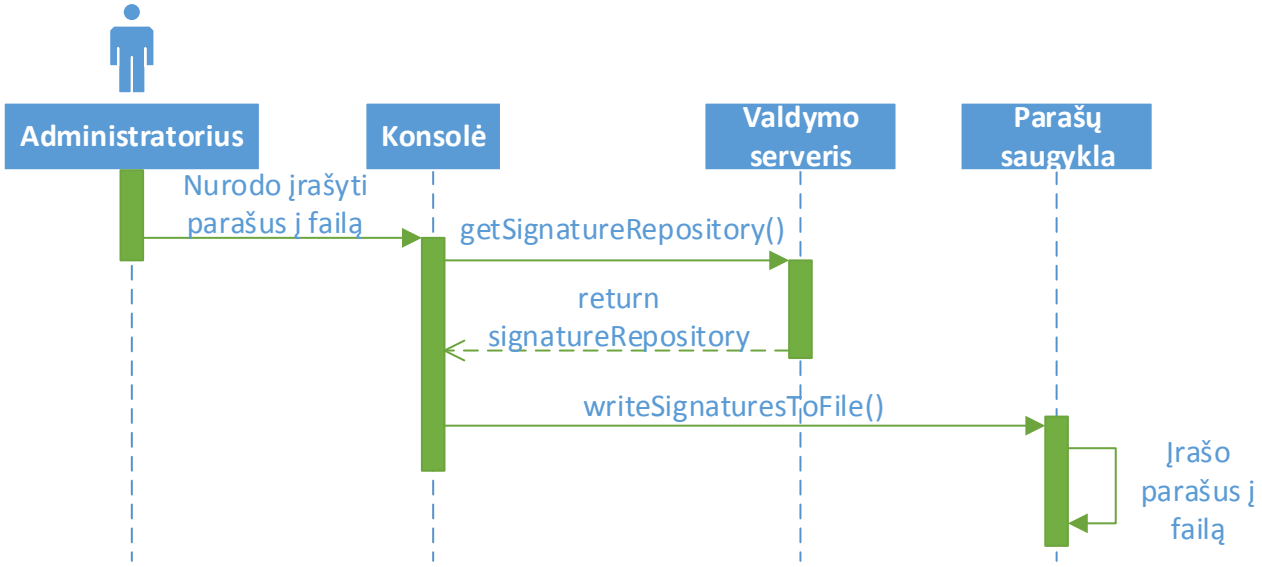
12 pav. Naujo žalingo IP adreso įvedimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori įvesti naują žalingą IP adresą ir įveda naują IP adresą. Konsolė iškviečia valdymo serverio getAgent() metodą, kuris grąžina agentą kintamajame agent. Jei grąžintas kintamasis netuščias (jo reikšmė nėra lygi null), tuomet konsolė iškviečia agento addMaliciousIPAddress(IP) metodą, kur kaip argumentas nurodytas administratoriaus įvestas IP adresas. Agentas patikrina, ar toks IP adresas nėra jau užregistruotas. Jei tokio nėra, tuomet yra įvedamas naujas žalingas IP adresas.



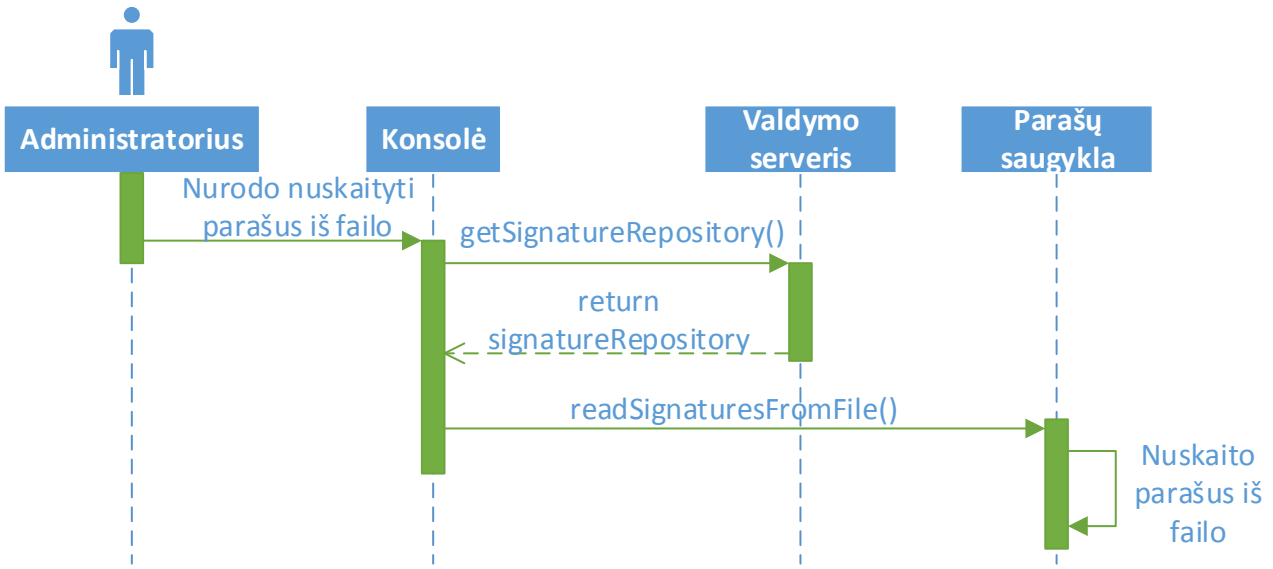
13 pav. Žalingo IP adreso ištrynimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori ištrinti žalingą IP adresą ir įveda IP adresą. Konsolė iškviečia valdymo serverio getAgent() metodą, kuris grąžina agentą kintamajame agent. Jei grąžintas kintamasis netuščias (jo reikšmė nėra lygi null), tuomet konsolė iškviečia agento removeMaliciousIPAddress(IP) metodą, kur kaip argumentas nurodytas administratoriaus įvestas IP adresas. Agentas patikrina, ar toks IP adresas yra užregistruotas. Jei toks yra, tuomet jis yra ištrinamas.



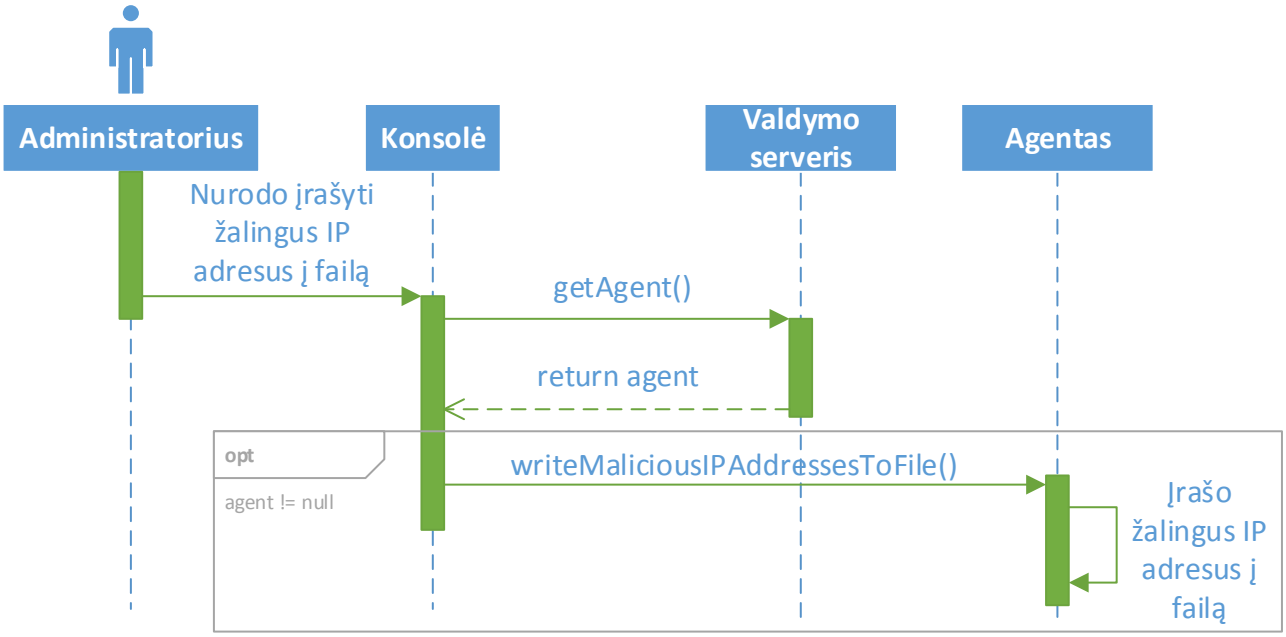
14 pav. Parašų įrašymas į failą

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori įrašyti parašus į failą. Konsolė kreipiasi į valdymo serverio getSignatureRepository() metodą, kuris konsolei grąžina parašų saugyklą kintamajame signatureRepository. Tuomet konsolė kreipiasi į parašų saugyklos writeSignaturesToFile() metodą, kuris įrašo visus esamus parašus į failą.



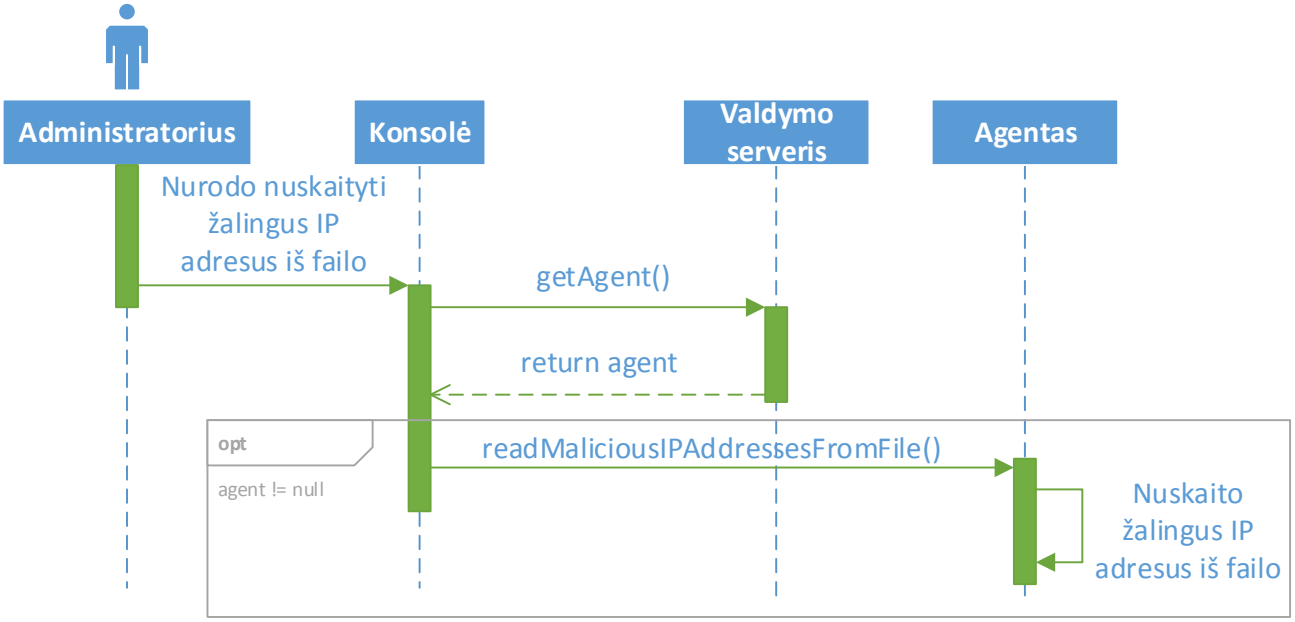
15 pav. Parašų nuskaitymas iš failo

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori nuskaityti parašus iš failo. Konsolė kreipiasi į valdymo serverio getSignatureRepository() metodą, kuris konsolei grąžina parašų saugyklą kintamajame signatureRepository. Tuomet konsolė kreipiasi į parašų saugyklos readSignaturesFromFile() metodą, kuris nuskaito visus parašus iš failo.



16 pav. Žalingų IP adresų įrašymas į failą

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori įrašyti žalingus IP adresus į failą. Konsolė iškviečia valdymo serverio getAgent() metodą, kuris grąžina agentą kintamajame agent. Jei grąžintas kintamasis netuščias (jo reikšmė nėra lygi null), tuomet konsolė iškviečia agento writeMaliciousIPAddressesToFile() metodą, kuris įrašo visus turimus žalingus IP adresus į failą.



17 pav. Žalingų IP adresų nuskaitymas iš failo

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori nuskaityti žalingus IP adresus iš failo. Konsolė iškviečia valdymo serverio getAgent() metodą, kuris grąžina agentą kintamajame agent. Jei grąžintas kintamasis netuščias (jo reikšmė nėra lygi null), tuomet konsolė iškviečia agento readMaliciousIPAddressesFromFile() metodą, kuris nuskaito visus žalingus IP adresus iš failo.

4.2. Įsilaužimo prevencijos sistemos architektūra

Šiame poskyryje bus aprašyta aktyvi įsilaužimo prevencijos sistema: jos veikimas (4.2.1), administratoriaus atliekamos užduotys (4.2.2), sistemos statinis modelis (4.2.3) ir dinaminis modelis (4.2.4).

4.2.1. Veikimo aprašymas

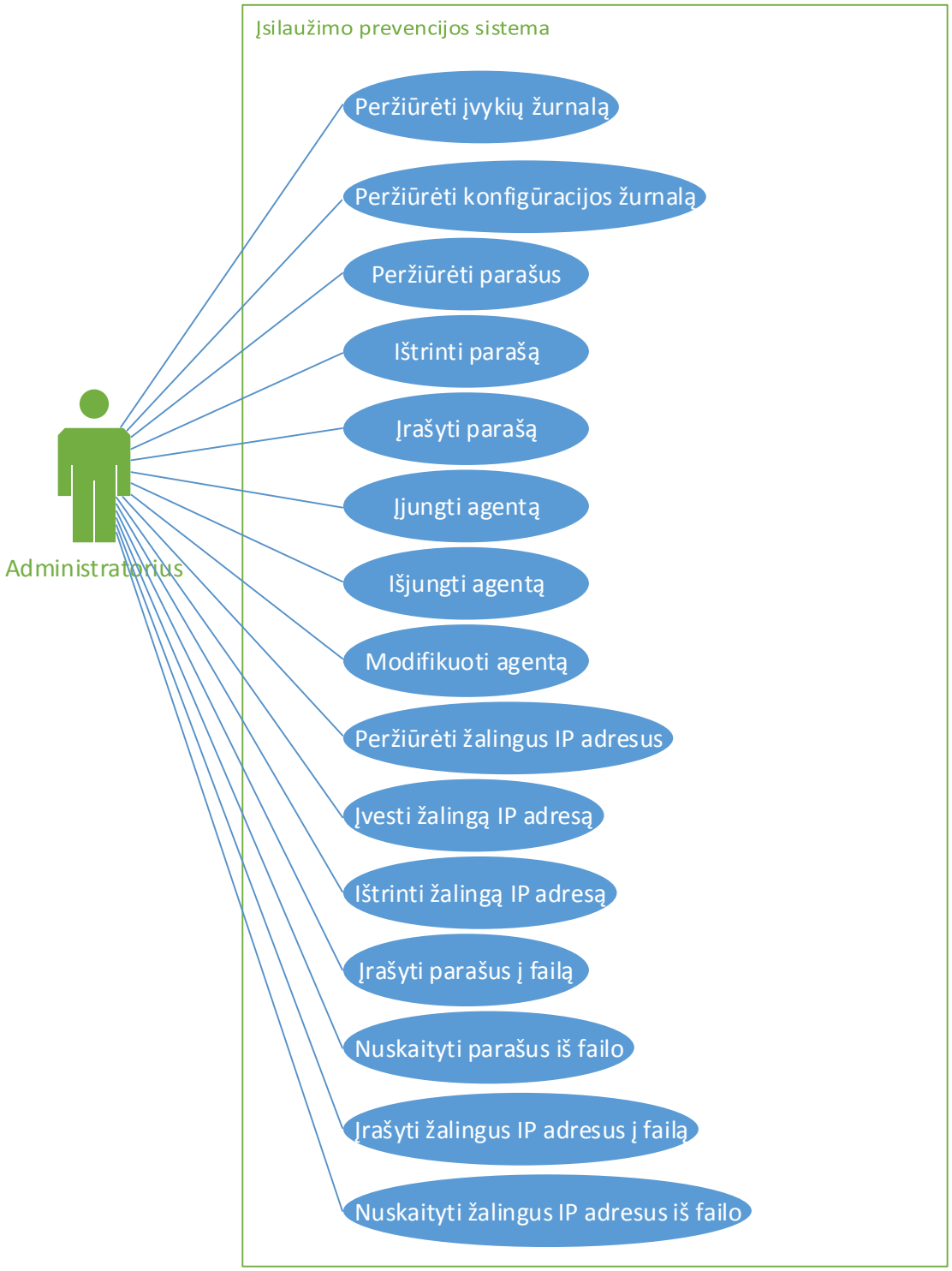
Įsilaužimo prevencijos sistema bus apsaugos sistema aktyviai stebinti ir tirianti ar nėra galimų įsilaužimo bandymų į „Firebird“ duomenų bazių valdymo sistemą. Šios sistemos pagrindiniai komponentai bus:

- Agentas – sistemos dalis, kuri įgaliotojo serverio (angl. Proxy) pagrindu perima tinklo srauto, tenkančio duomenų bazių valdymo sistemai, paketus, tiria, ar nebuvo įsilaužimo bandymų, ir gerybinius paketus perduoda pačiai duomenų bazių valdymo sistemai;
- Valdymo serveris – pagrindinė sistemos dalis, kuri valdys agentą, kontroliuos visu kitų komponentų veikimą;
- Parašų saugykla – saugykla atliekanti parašų saugojimą, naujų pridėjimą ir esamų pašalinimą;
- Žurnalų valdytojas – sistemos dalis valdanti visus jos žurnalus: įvykių, konfigūracijos pakeitimų;
- Konsolė – tekstinė sąsaja administratoriams, skirta valdyti įsilaužimo aptikimo sistemai.

Agentas įgaliotojo serverio principu priims ir analizuos paketus tinklo srauto, skirto duomenų bazių valdymo sistemai. Agentas bus aktyvus, kadangi jis priims visą tinklo srautą skirtą duomenų bazių valdymo sistemai, gerybinius paketus siųs duomenų bazių valdymo sistemai, o žalingus atmes. Agentas analizuos gautus paketus naudodamas parašu paremtą aptikimą pagal parašus, saugomus parašų saugykloje. Aptikus galimą įsilaužimą, agentas išsaugos detalią informaciją apie jį įvykių žurnale ir atmes paketą, susijusį su tuo įsilaužimo bandymu. Konsolė leis administratoriui valdyti, kada agentas turi būti įjungtas, kokio prievado (angl. Port) klausyti, valdyti naudojamus parašus, valdyti žalingų IP adresų sąrašą, peržiūrėti žurnalus. Žurnalų valdytojas registruos įvairius konfigūracijos pasikeitimus į konfigūracijos žurnalą.

4.2.2. Užduotys ir jų vykdymo scenarijai

Žemiau pateikta įsilaužimo prevencijos sistemos panaudojimo atvejų diagrama (18 pav. Įsilaužimo prevencijos sistemos panaudojimo atvejų diagrama).



18 pav. Įsilaužimo prevencijos sistemos panaudojimo atvejų diagrama

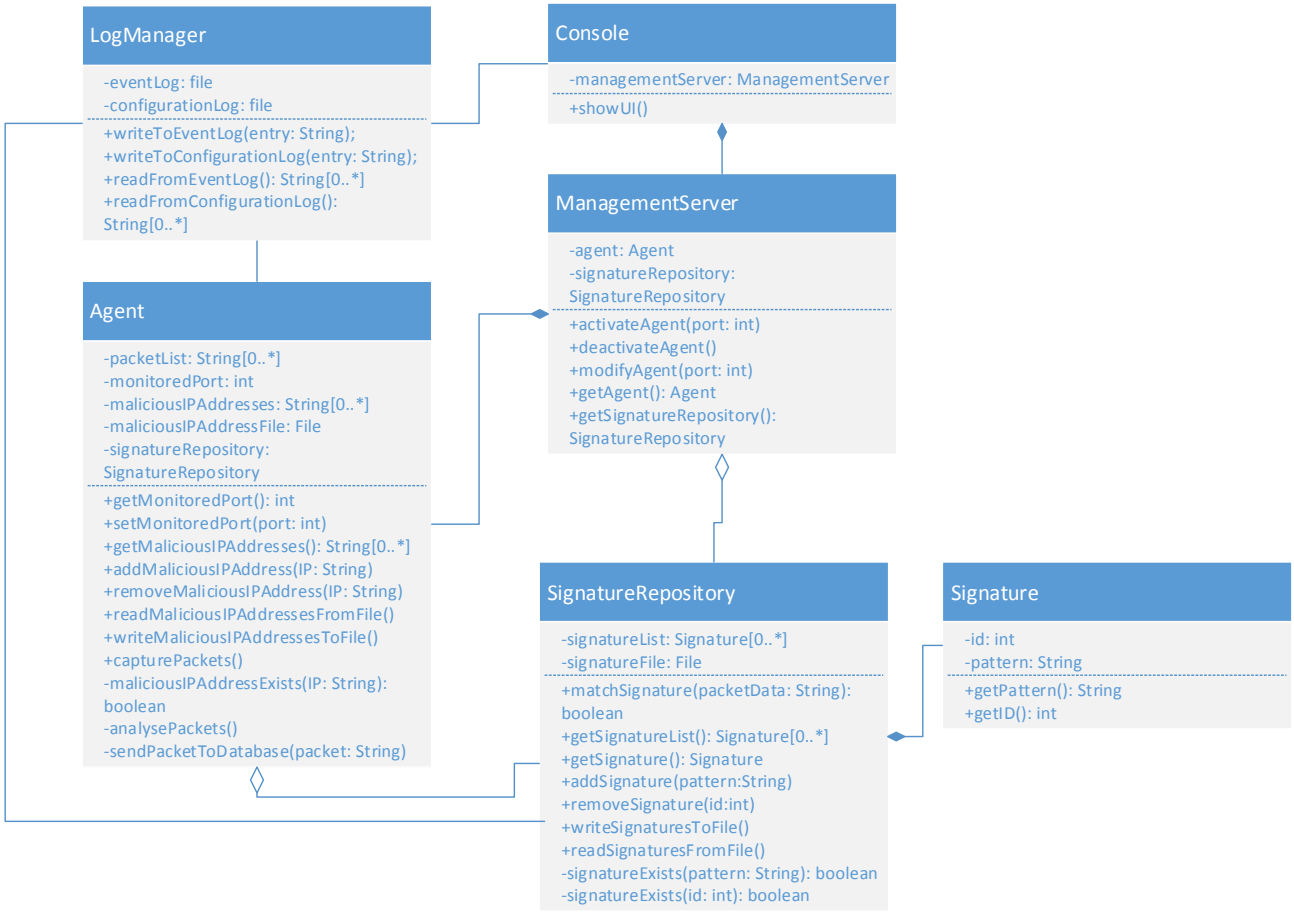
Pagrindinės administratoriaus atliekamos užduotys:

1. Peržiūrėti įvykių žurnalą. Administratorius naudodamasis konsole peržvelgia įvykių žurnalą, kuriame registruojami visi aptikti įsilaužimo bandymai su informacija apie juos (iš kokio adreso atėjo paketas, kokią komandą/duomenis laikė paketas ir t.t.).

2. Peržiūrėti konfigūracijos žurnalą. Administratorius naudodamasis konsole peržvelgia konfigūracijos žurnalą, kuriame saugomi agento nustatymų bei parašų saugyklos pasikeitimai.
3. Peržiūrėti parašus. Administratorius naudodamasis konsole peržvelgia parašų saugykloje esančius parašus.
4. Ištrinti parašą. Administratorius naudodamasis konsole ištriną parašą su nurodytu identifikacijos numeriu.
5. Įrašyti parašą. Administratorius naudodamasis konsole sukuria naują parašą, kuriam automatiškai sugeneruojamas identifikacijos numeris.
6. Įjungti agentą. Administratorius naudodamasis konsole įjungia agentą, kad šis skaitytų paketus, skirtus duomenų bazių valdymo sistemai.
7. Išjungti agentą. Administratorius naudodamasis konsole išjungia agentą, kad šis nebeskaitytų jokių paketų.
8. Modifikuoti agentą. Administratorius naudodamasis konsole pakeičia prievadą, kurio klausosi agentas.
9. Peržiūrėti žalingus IP adresus. Administratorius naudodamasis konsole peržiūri sąrašą užregistruotų žalingų IP adresų.
10. Įvesti žalingą IP adresą. Administratorius naudodamasis konsole įveda naują žalingą IP adresą.
11. Ištrinti žalingą IP adresą. Administratorius naudodamasis konsole ištrina žalingą IP adresą.
12. Įrašyti parašus į failą. Administratorius naudodamasis konsole nurodo parašų saugyklai visus turimus parašus įrašyti į failą.
13. Nuskaityti parašus iš failo. Administratorius naudodamasis konsole nurodo parašų saugyklai nuskaityti parašus iš failo.
14. Įrašyti žalingus IP adresus į failą. Administratorius naudodamasis konsole nurodo agentui visus turimus žalingus IP adresus įrašyti į failą.
15. Nuskaityti žalingus IP adresus iš failo. Administratorius naudodamasis konsole nurodo agentui nuskaityti žalingus IP adresus iš failo.

4.2.3. Statinis sistemos modelis

Žemiau pateikiama įsilaužimo prevencijos sistemos klasių diagrama (19 pav. Įsilaužimo prevencijos sistemos klasių diagrama).



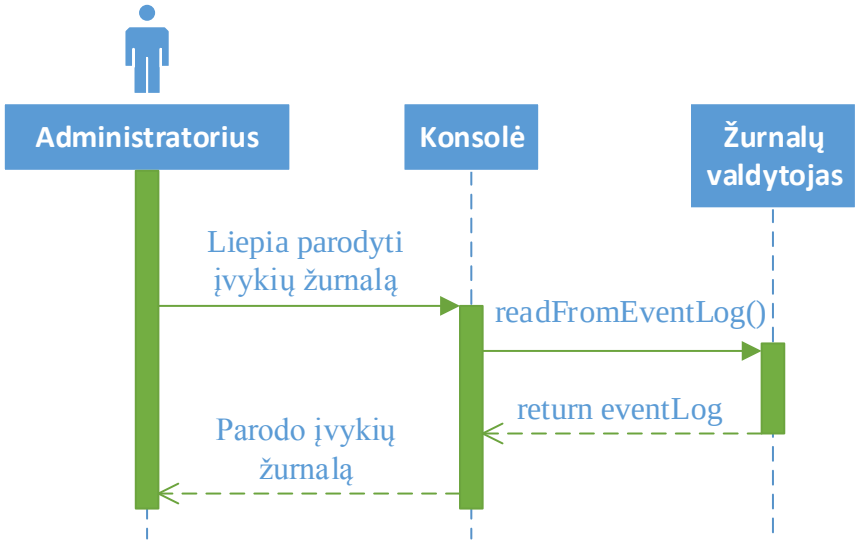
19 pav. Įsilaužimo prevencijos sistemos klasių diagrama

4.2.4. Dinaminis sistemos modelis

Žemiau pateikiamos sekų diagramos, atitinkančios kiekvieną panaudojimo scenarijų:

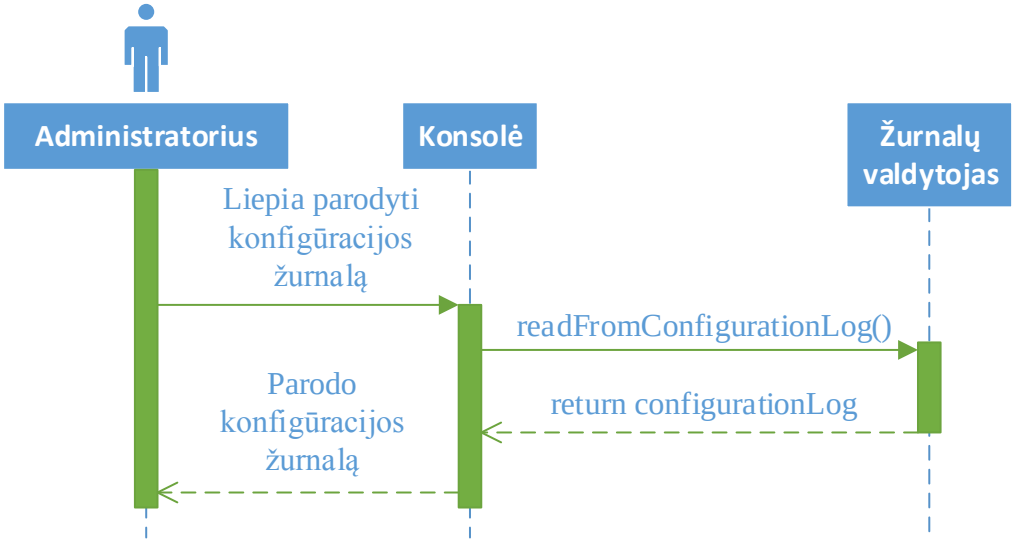
1. Peržiūrėti įvykių žurnalą (20 pav. Įvykių žurnalo peržiūrėjimas);
2. Peržiūrėti konfigūracijos žurnalą (21 pav. Konfigūracijos žurnalo peržiūrėjimas);
3. Peržiūrėti parašus (22 pav. Parašų peržiūrėjimas);
4. Įrašyti parašą (23 pav. Naujo parašo įvedimas);
5. Ištrinti parašą (24 pav. Parašo ištrynimas);
6. Įjungti agentą (25 pav. Agento įjungimas);

7. Išjungti agentą (26 pav. Agento išjungimas);
8. Modifikuoti agentą (27 pav. Agento modifikavimas);
9. Peržiūrėti žalingus IP adresus (28 pav. Žalingų IP adresų peržiūrėjimas);
10. Įvesti žalingą IP adresą (29 pav. Naujo žalingo IP adreso įvedimas);
11. Ištrinti žalingą IP adresą (30 pav. Žalingo IP adreso ištrynimas);
12. Įrašyti parašus į failą (31 pav. Parašų įrašymas į failą);
13. Nuskaityti parašus iš failo (32 pav. Parašų nuskaitymas iš failo);
14. Įrašyti žalingus IP adresus į failą (33 pav. Žalingų IP adresų įrašymas į failą);
15. Nuskaityti žalingus IP adresus iš failo (34 pav. Žalingų IP adresų nuskaitymas iš failo).



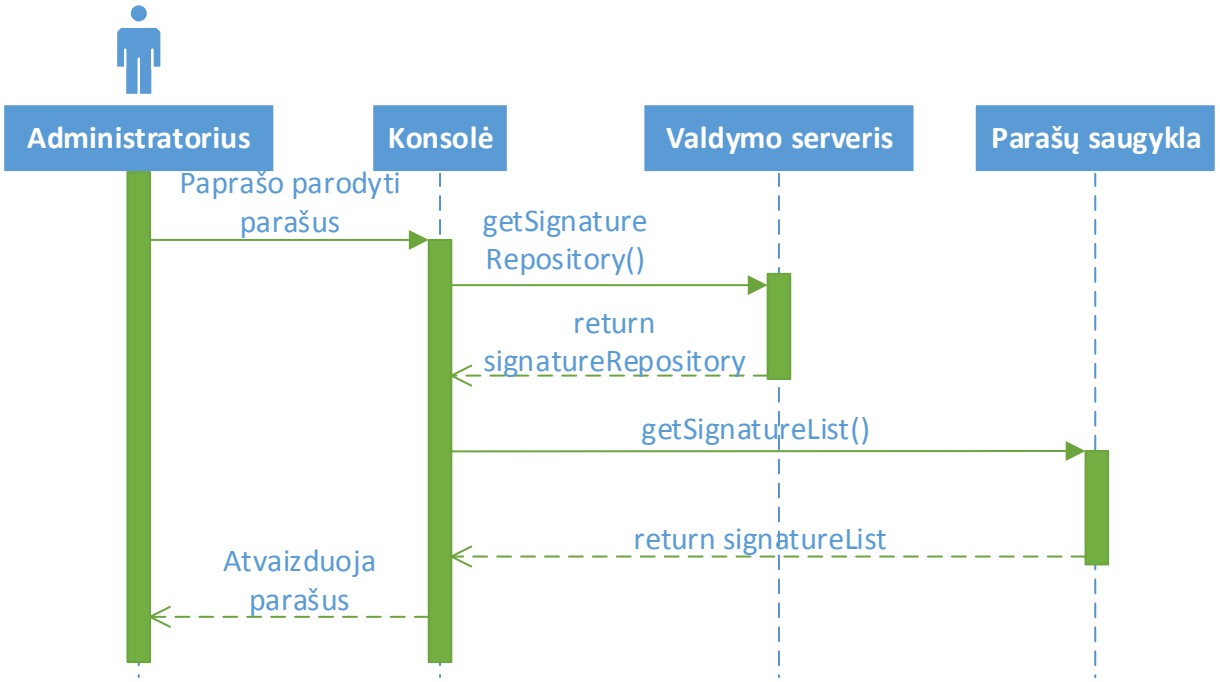
20 pav. Įvykių žurnalo peržiūrėjimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori peržiūrėti įvykių žurnalą. Konsolė kreipiasi į žurnalų valdytojo readFromEventLog() metodą, kuris konsolei grąžina iš failo nuskaitytą įvykių žurnalą kintamajame eventLog. Konsolė tada atvaizduoja administratoriui kintamojo eventLog turinį.



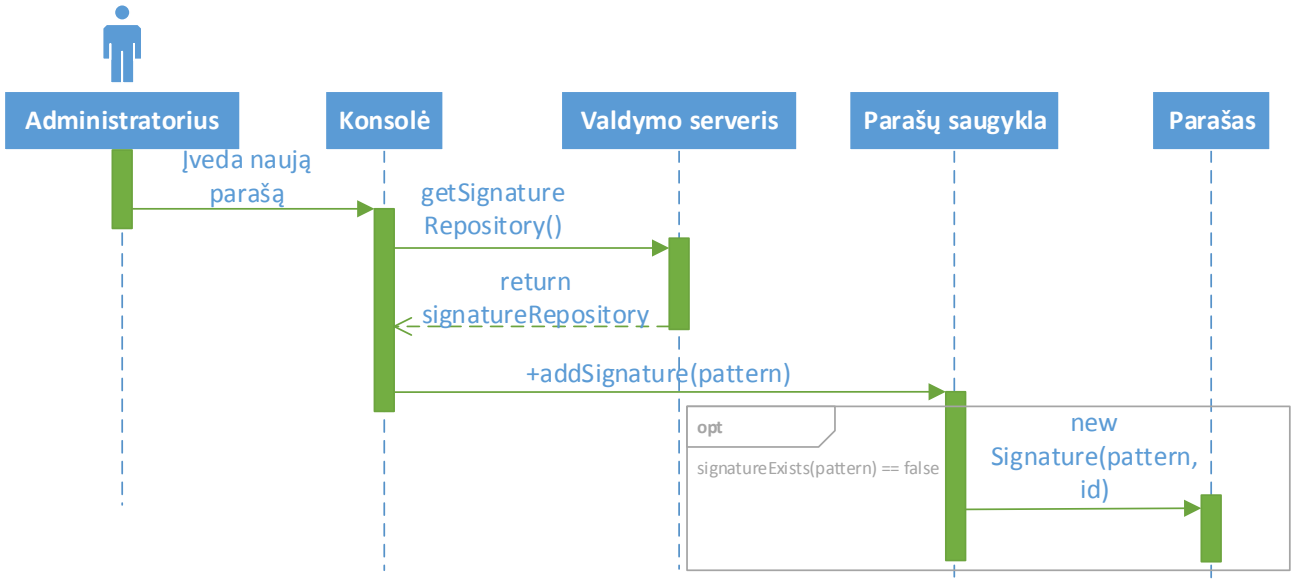
21 pav. Konfigūracijos žurnalo peržiūrėjimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori peržiūrėti konfigūracijos žurnalą. Konsolė kreipiasi į žurnalų valdytojo readFromConfigurationLog() metodą, kuris konsolei grąžina iš failo nuskaitytą konfigūracijos žurnalą kintamajame configurationLog. Konsolė tada atvaizduoja administratoriui kintamojo configurationLog turinį.



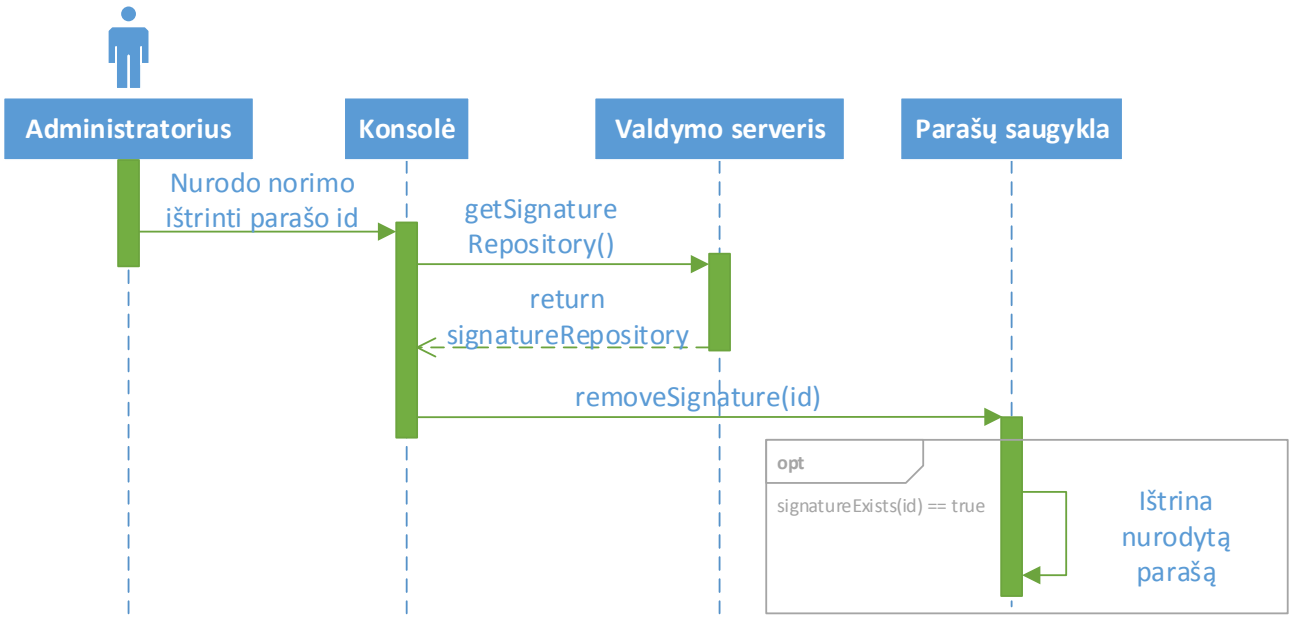
22 pav. Parašų peržiūrėjimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori peržiūrėti įvestus parašus. Konsolė kreipiasi į valdymo serverio getSignatureRepository() metodą, kuris konsolei grąžina parašų saugyklą kintamajame signatureRepository. Tuomet konsolė kreipiasi į parašų saugyklos getSignatureList() metodą, kuris konsolei grąžiną sąrašą visų įvestų parašų kintamajame signatureList. Galiausiai konsolė administratoriui atvaizduoja kintamojo signatureList turinį.



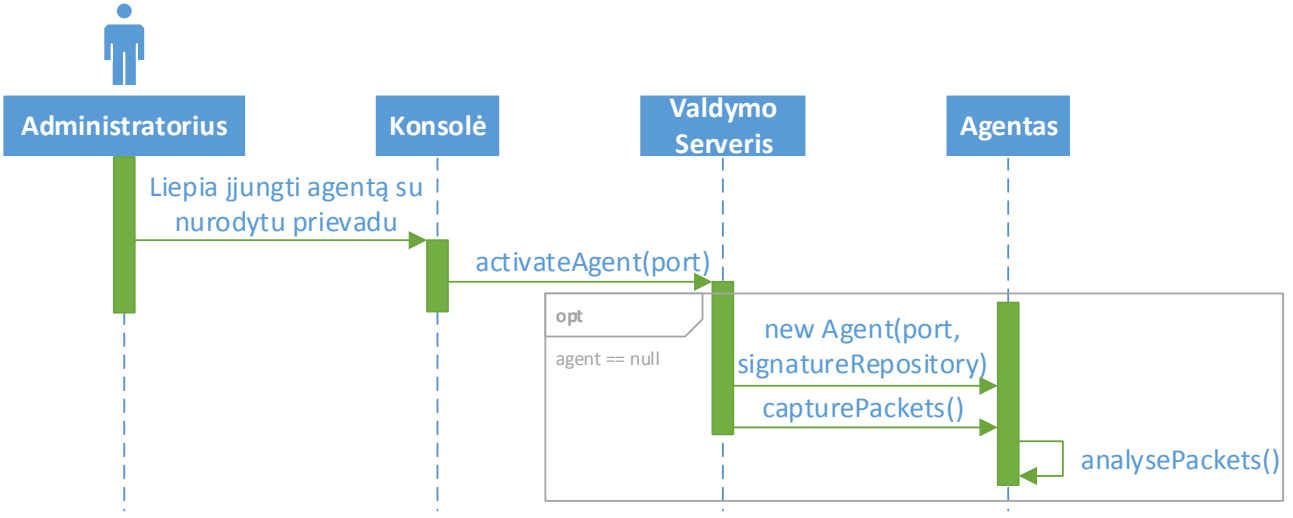
23 pav. Naujo parašo įvedimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori įvesti naują parašą ir įveda jo šabloną. Konsolė kreipiasi į valdymo serverio getSignatureRepository() metodą, kuris konsolei grąžina parašų saugyklą kintamajame signatureRepository. Tuomet konsolė kreipiasi į parašų saugyklos addSignature(pattern) metodą, kur kaip parametras pattern pateikiamas administratoriaus įvestas šablonas. Parašų saugykla patikrina, ar nėra parašo su pateiktu šablonu. Jei tokio parašo nėra, tuomet sukuriamas naujas parašas su pateiktu šablonu ir jam sugeneruojamas naujas identifikacijos numeris.



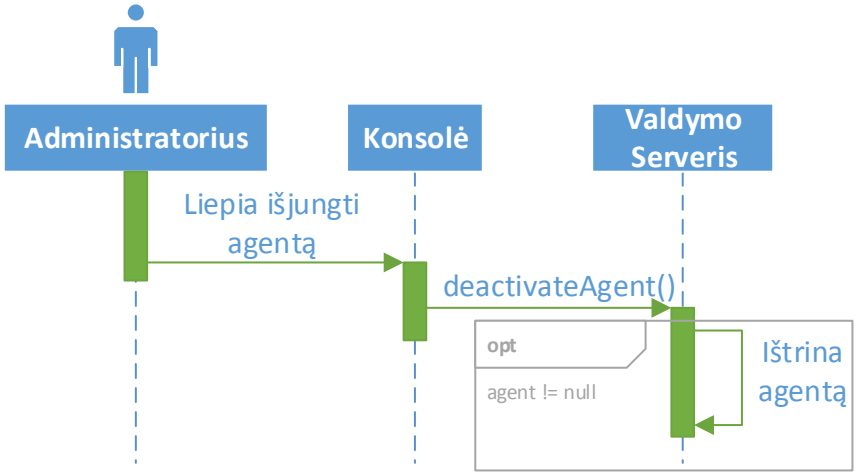
24 pav. Parašo ištrynimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori ištrinti esamą parašą ir įveda parašo identifikacijos numerį. Konsolė kreipiasi į valdymo serverio getSignatureRepository() metodą, kuris konsolei grąžina parašų saugyklą kintamajame signatureRepository. Tuomet konsolė kreipiasi į parašų saugyklos removeSignature(id) metodą, kur kaip parametras id pateikiamas administratoriaus įvestas identifikacijos numeris. Parašų saugykla patikrina, ar egzistuoja parašas su nurodytu identifikacijos numeriu. Jei toks parašas egzistuoja, tuomet nurodytas parašas yra ištrinamas.



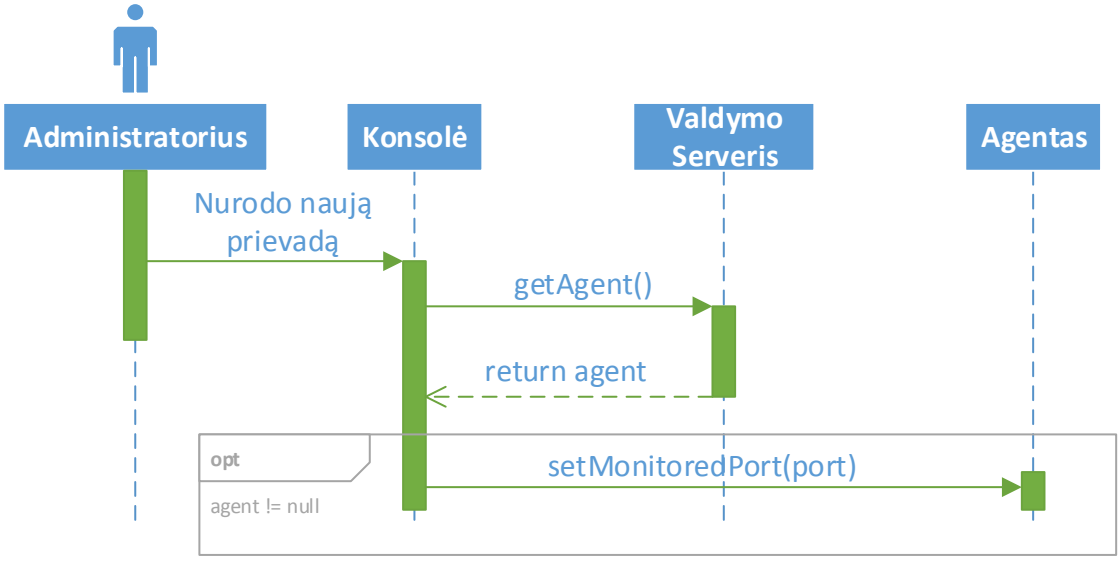
25 pav. Agento įjungimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori įjungti agentą ir įveda prievadą, kurio nori, kad agentas klausytųsi. Konsolė iškviečia valdymo serverio activateAgent(port) metodą, kur kaip argumentas port pateikiamas administratoriaus įvestas prievadas. Valdymo serveris patikrina ar agentas egzistuoja. Jei agentas neegzistuoja, tuomet valdymo serveris sukuria naują agentas su nurodytu prievadu ir aktyvuoja agento capturePackets() metodą, kuris aktyvuoja agento paketų klausymą ir iškviečia analysePackets() metodą, kuris vykdo pastebėtų paketų analizę.



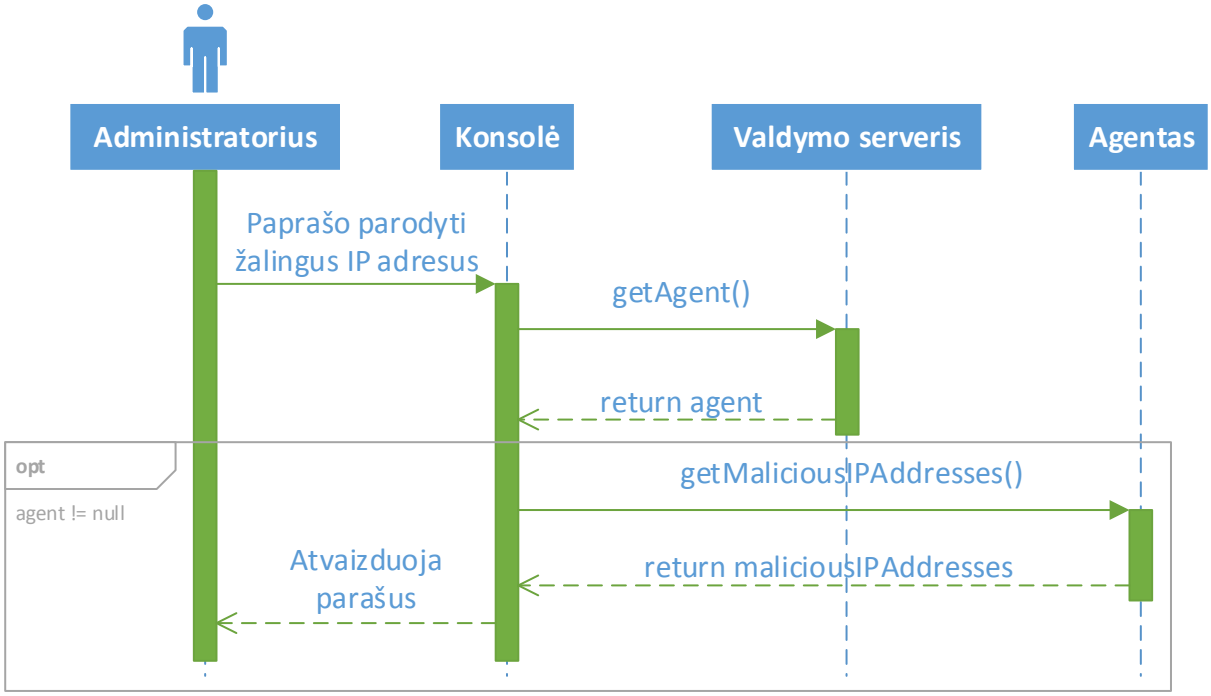
26 pav. Agento išjungimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori išjungti agentą. Konsolė iškviečia valdymo serverio deactivateAgent() metodą, kuris, jei agentas egzistuoja, ištrina agentą.



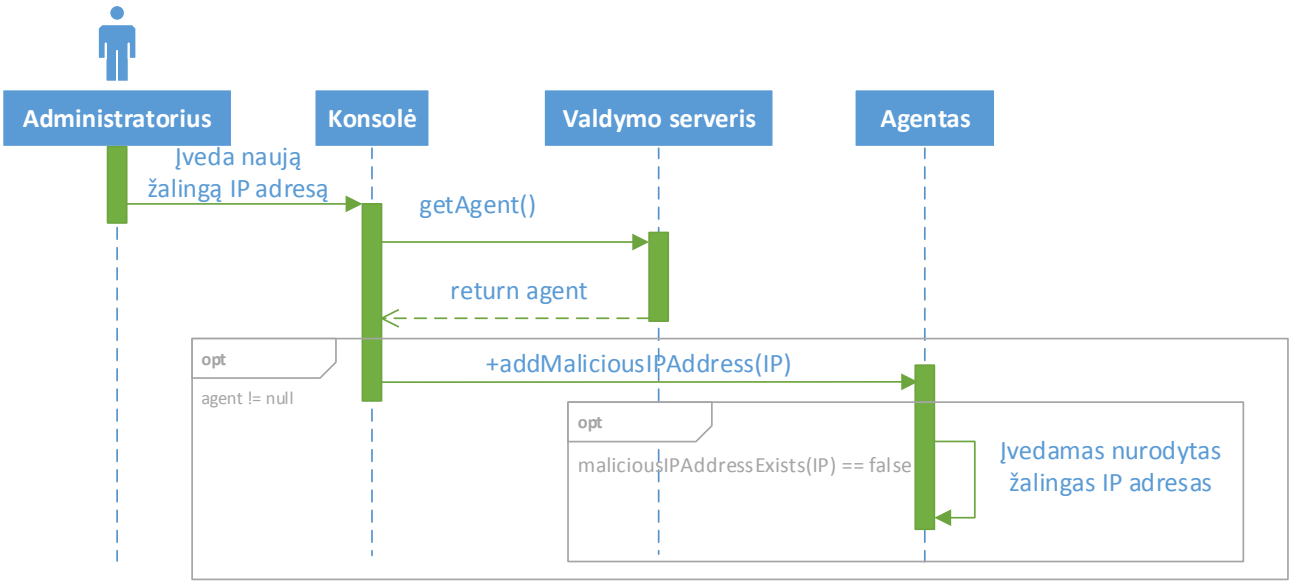
27 pav. Agento modifikavimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori modifikuoti agentą ir nurodo naują prievadą. Konsolė iškviečia valdymo serverio getAgent() metodą, kuris grąžina agentą kintamajame agent. Jei grąžintas kintamasis netuščias (jo reikšmė nėra lygi null), tuomet konsolė iškviečia agento setMonitoredPort(port) metodą, kur kaip argumentas port pateikiamas administratoriaus nurodytas naujas prievadas. Agentui tuomet nustatomas naujas prievadas.



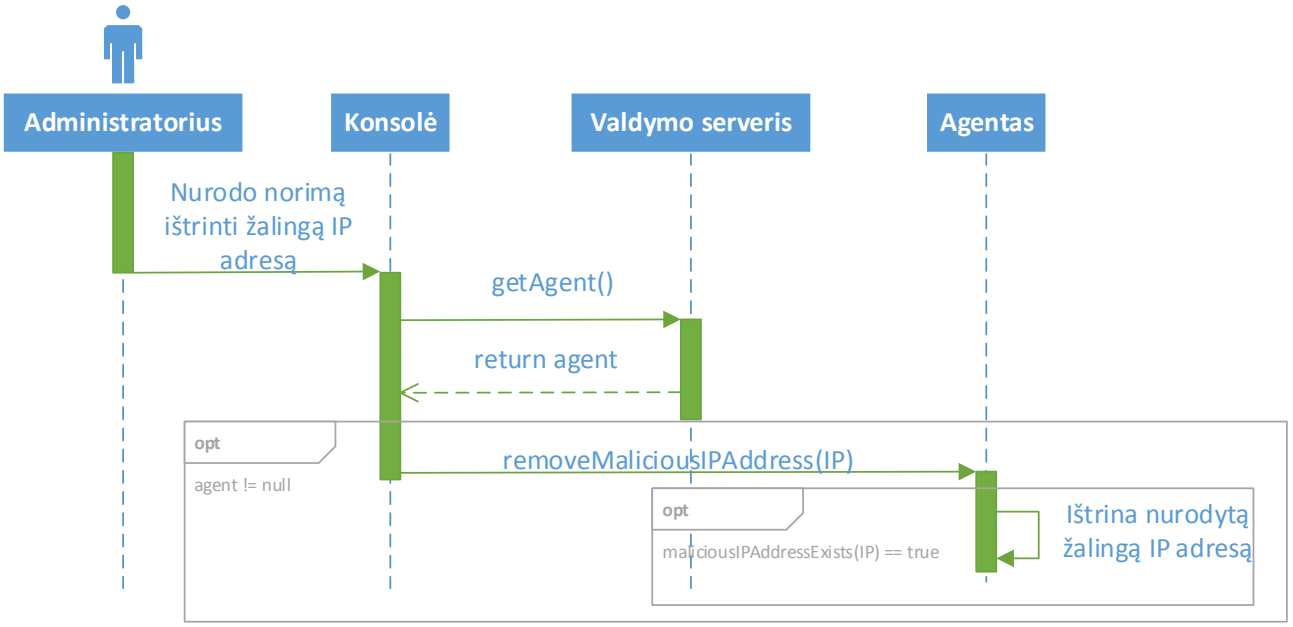
28 pav. Žalingų IP adresų peržiūrėjimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori peržiūrėti esamus žalingus IP adresus. Konsolė iškviečia valdymo serverio getAgent() metodą, kuris grąžina agentą kintamajame agent. Jei grąžintas kintamasis netuščias (jo reikšmė nėra lygi null), tuomet konsolė iškviečia agento getMaliciousIPAddresses() metodą, kuris grąžina visų įvestų žalingų IP adresų sąrašą kintamajame maliciuosIPAddresses. Tuomet konsolė atvaizduoja administratoriui kintamojo maliciousIPAddresses turinį.



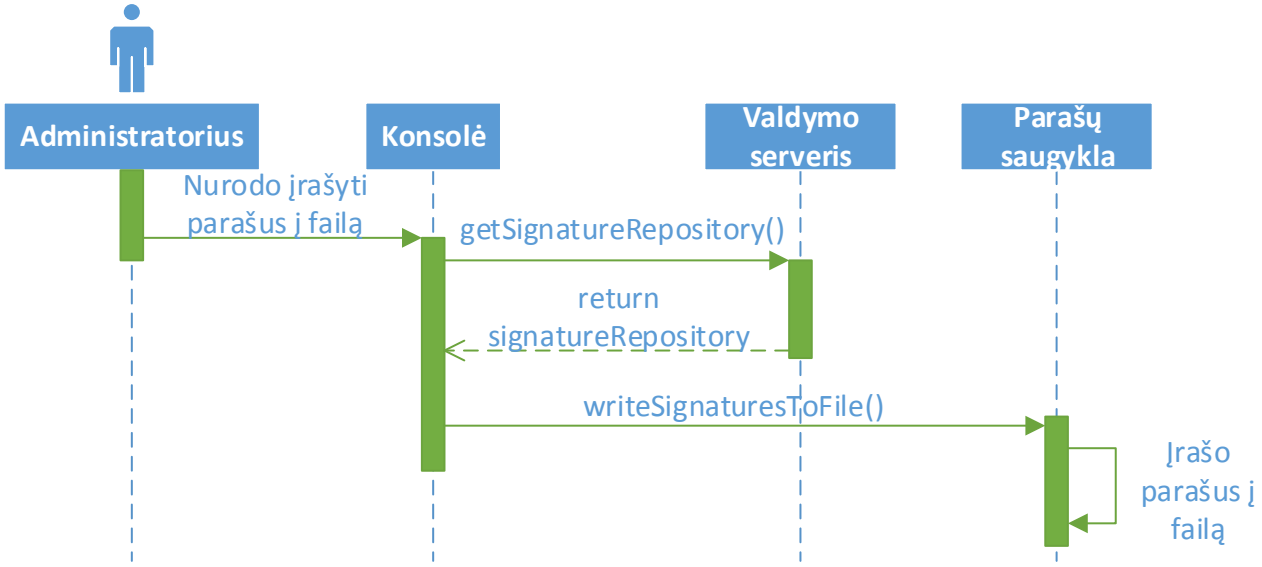
29 pav. Naujo žalingo IP adreso įvedimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori įvesti naują žalingą IP adresą ir įveda naują IP adresą. Konsolė iškviečia valdymo serverio getAgent() metodą, kuris grąžina agentą kintamajame agent. Jei grąžintas kintamasis netuščias (jo reikšmė nėra lygi null), tuomet konsolė iškviečia agento addMaliciousIPAddress(IP) metodą, kur kaip argumentas nurodytas administratoriaus įvestas IP adresas. Agentas patikrina, ar toks IP adresas nėra jau užregistruotas. Jei tokio nėra, tuomet yra įvedamas naujas žalingas IP adresas.



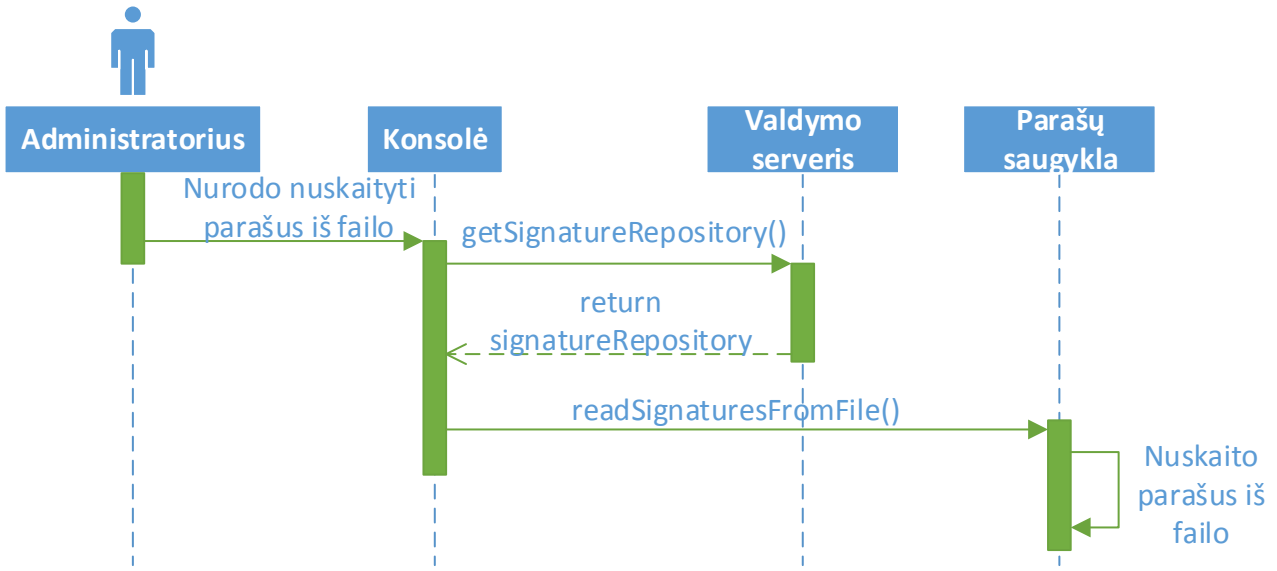
30 pav. Žalingo IP adreso ištrynimas

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori ištrinti žalingą IP adresą ir įveda IP adresą. Konsolė iškviečia valdymo serverio getAgent() metodą, kuris grąžina agentą kintamajame agent. Jei grąžintas kintamasis netuščias (jo reikšmė nėra lygi null), tuomet konsolė iškviečia agento removeMaliciousIPAddress(IP) metodą, kur kaip argumentas nurodytas administratoriaus įvestas IP adresas. Agentas patikrina, ar toks IP adresas nėra jau užregistruotas. Jei toks yra, tuomet jis yra ištrinamas.



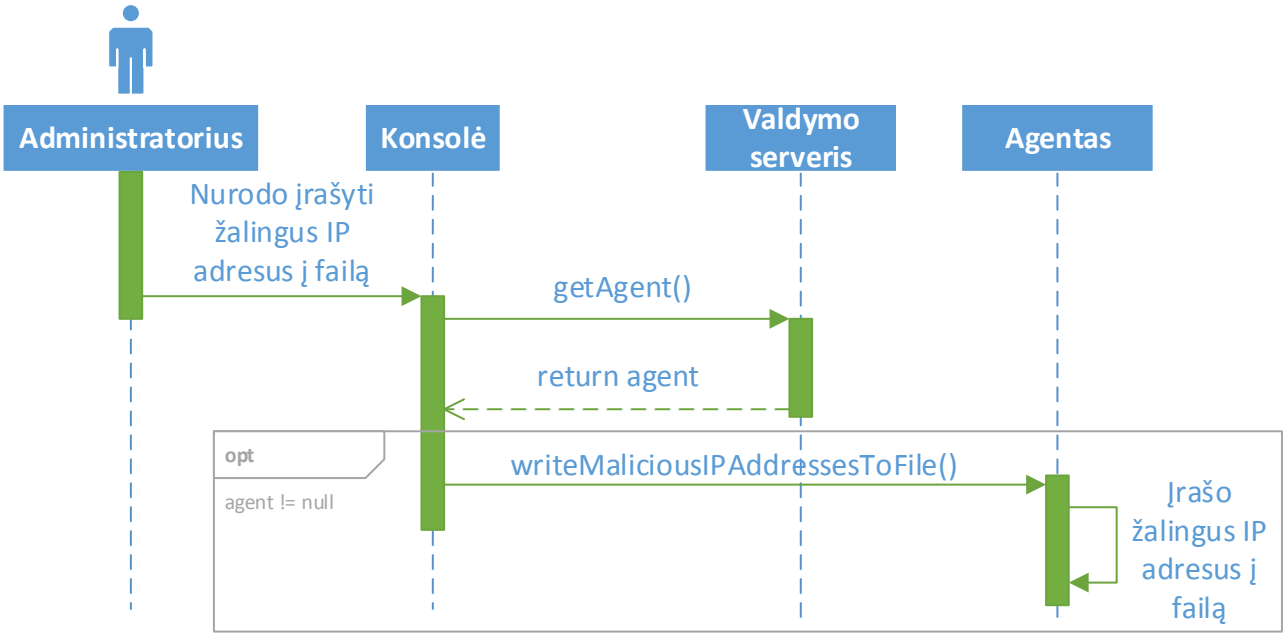
31 pav. Parašų įrašymas į failą

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori įrašyti parašus į failą. Konsolė kreipiasi į valdymo serverio getSignatureRepository() metodą, kuris konsolei grąžina parašų saugyklą kintamajame signatureRepository. Tuomet konsolė kreipiasi į parašų saugyklos writeSignaturesToFile() metodą, kuris įrašo visus esamus parašus į failą.



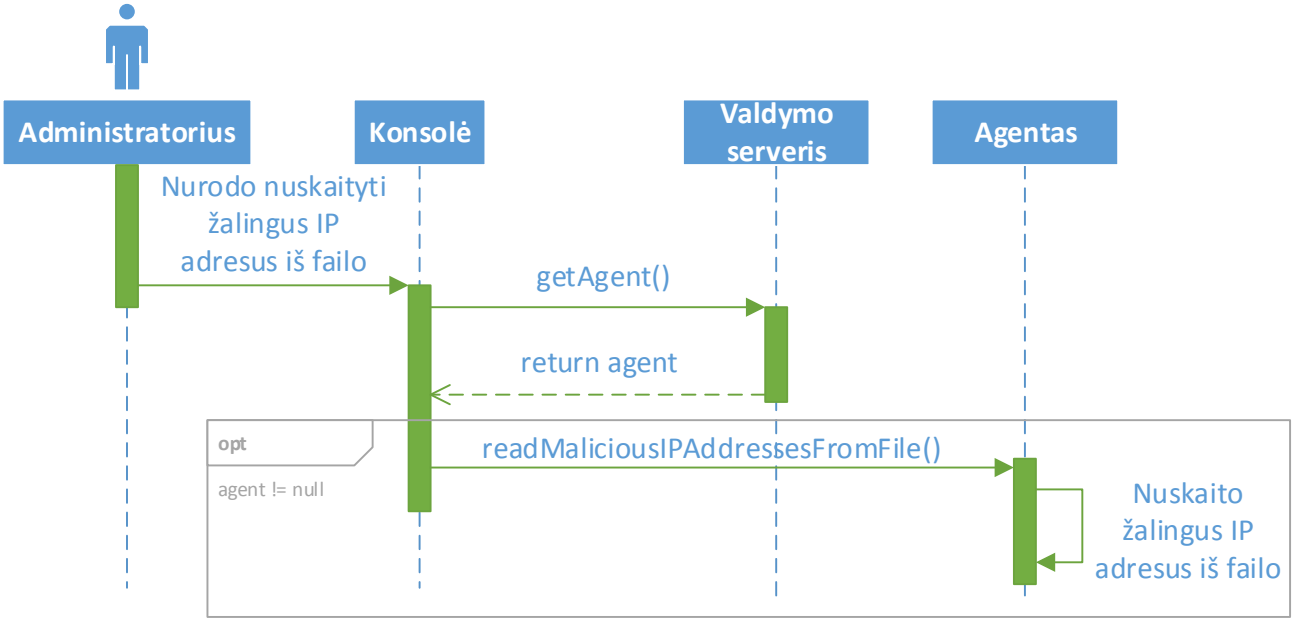
32 pav. Parašų nuskaitymas iš failo

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori nuskaityti parašus iš failo. Konsolė kreipiasi į valdymo serverio getSignatureRepository() metodą, kuris konsolei grąžina parašų saugyklą kintamajame signatureRepository. Tuomet konsolė kreipiasi į parašų saugyklos readSignaturesFromFile() metodą, kuris nuskaito visus parašus iš failo.



33 pav. Žalingų IP adresų įrašymas į failą

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori įrašyti žalingus IP adresus į failą. Konsolė iškviečia valdymo serverio getAgent() metodą, kuris grąžina agentą kintamajame agent. Jei grąžintas kintamasis netuščias (jo reikšmė nėra lygi null), tuomet konsolė iškviečia agento writeMaliciousIPAddressesToFile() metodą, kuris įrašo visus turimus žalingus IP adresus į failą.



34 pav. Žalingų IP adresų nuskaitymas iš failo

Šiame scenarijuje administratorius naudodamasis konsolės pateikiamu meniu padaro pasirinkimą, jog nori nuskaityti žalingus IP adresus iš failo. Konsolė iškviečia valdymo serverio getAgent() metodą, kuris grąžina agentą kintamajame agent. Jei grąžintas kintamasis netuščias (jo reikšmė nėra lygi null), tuomet konsolė iškviečia agento readMaliciousIPAddressesFromFile() metodą, kuris nuskaito visus žalingus IP adresus iš failo.

4.3. Architektūrų palyginimas

Aukščiau aprašytų įsilaužimo aptikimo ir įsilaužimo prevencijos sistemų architektūros yra gana panašios. Pagrindiniai jų skirtumai yra tai, kokią įtaką agentas turi tinklo srautui, skirtam duomenų bazių valdymo sistemai, kaip elgiasi su aptiktais įsilaužimais bei kaip realizuotas agento bendras veikimas. Bus palyginti keli architektūrų realizacijos ir veikimo aspektai.

Įsilaužimo aptikimo sistemos agentas vykdys pasyvią veiklą, kadangi jis analizuos kopijas paketų, skirtų duomenų bazių valdymo sistemai ir tik užfiksuos galimus įsilaužimus, bet neturės galimybių užkirsti jiems kelio. Dėl agento pasyvumo, ši architektūra labiau pasikliauja serverio, kuriame yra duomenų bazių valdymo sistema, pačios duomenų bazių valdymo sistemos turimomis apsaugos priemonėmis bei administratoriaus gebėjimu laiku ir efektyviai atsakyti į aptiktą įsilaužimo bandymą. Taip pat agentas nedarys jokio poveikio duomenų bazių valdymo sistemos veikimo spartai. Realizacija tokio agento yra paprastesnė nei aktyvaus.

Įsilaužimo prevencijos sistemos agentas vykdys aktyvią veiklą, kadangi jis kaip įgaliotasis serveris priims visus paketus, skirtus duomenų bazių valdymo sistemai, ir galės atmesti visus, kuriuos atpažins kaip įsilaužimo bandymus. Dėl agento aktyvumo, ši architektūra mažiau remiasi pašalinėmis apsaugos galimybėmis. Tačiau dėl to, kad agentas perima tinklo srautą, skirtą duomenų bazių valdymo sistemai, priklausomai nuo srauto intensyvumo, agentas gali paveikti duomenų bazių valdymo sistemos veikimo efektyvumą. Jei duomenų bazė yra intensyviai naudojama, t.y. paketų kiekis tenkantis jai yra labai didelis, agentas gali per ilgai užlaikyti analizuojamus paketus, taip pailginant duomenų bazių valdymo sistemos atsako laiką jos naudotojams. Tai apsunkina agento realizaciją, jei norima kiek galima išvengti ilgo paketų užlaikymo.

Įsilaužimo aptikimo sistema yra tinkamesnė, jei pačioje duomenų bazių valdymo sistemoje ir serveryje yra apibrėžtos geros apsaugos priemonės ir norima tik jas papildyti/pastiprinti, bei dėl paprastesnio realizavimo. Įsilaužimo prevencijos sistema yra tinkamesnė, jei norima labiau sustiprinti duomenų bazių valdymo sistemą nuo įsilaužimų ir duomenų bazių valdymo sistemos veikimo efektyvumas (atsako laikas) nėra toks kritinis.

REZULTATAI IR IŠVADOS

Gauti darbo rezultatai:

1. Išnagrinėtos įsilaužimo aptikimo/prevencijos sistemos, pagrindinės įsilaužimo aptikimo metodologijos, įsilaužimo prevencijos būdai, jų panaudojimo scenarijai, pagrindinės technologijos, pagrindiniai jų tipai ir tipiniai komponentai;
2. Išanalizuota, kaip įsilaužimo aptikimo/prevencijos sistemos pritaikomos duomenų bazėms, kokios technikos naudojamos aptikti įsilaužimams duomenų bazėse;
3. Išnagrinėta „Firebird“ duomenų bazių valdymo sistema, jos techninės galimybės, ribojimai ir galimai esami pažeidžiamumai;
4. Suprojektuotos dvi galimos architektūros:
    a. Įsilaužimo aptikimo sistema – pasyvios apsaugos sistema, fiksuojanti aptiktus įsilaužimus;
    b. Įsilaužimo prevencijos sistema – aktyvios apsaugos sistema, fiksuojanti ir blokuojanti aptiktus įsilaužimus.

Gautos darbo išvados:

1. Parašu paremtas aptikimas efektyviausiai tinka žinomų įsilaužimų aptikimui;
2. Anomalijomis paremtas aptikimas efektyviausiai tinka naujų, nežinomų įsilaužimų aptikimui;
3. Norint pagerinti aptikimų tikslumą ir gebėti aptikti didesnį kiekį galimų įsilaužimų, įsilaužimo aptikimo/prevencijos sistemos turėtų naudoti daugiau nei vieną įsilaužimo aptikimo būdą;
4. Įsilaužimo prevencijos sistemos gali turėti įtakos apsaugomų sistemų veikimui/atsako laikui, kadangi dalis galimų prevencijos priemonių reikalauja, jog jų naudojami agentai arba sensoriai būtų tarp saugomų sistemų ir išorinio tinklo, užlaikantys sistemoms skirtą tinklo srautą iki nusprendžiama, jog jis gerybinis;
5. Dėl SQL kalbos ekspresyvumo yra sudėtinga tinkamai ir greitai įvertinti, ar vartotojo komanda yra gerybinė ar žalinga;
6. Duomenų sandėlių veiklos nenuspėjamumas ir platus spektras apsunkina žalingų komandų atskyrimą nuo gerybinių;

7. Suprojektuota įsilaužimo aptikimo sistema nepaveiks duomenų bazės valdymo sistemos efektyvumo, tačiau remsis išorinėmis (serverio, duomenų bazių valdymo sistemos, administratoriaus) apsaugos priemonėmis duomenų bazių valdymo sistemos apsaugoje;
8. Suprojektuota įsilaužimo prevencijos sistema gebės apsaugoti duomenų bazių valdymo sistemą nuo įsilaužimų, kurie atitiks turimus parašus, tačiau ji gali neigiamai paveikti duomenų bazių valdymo sistemos veikimo efektyvumą priklausomai nuo to, kaip sparčiai paketai yra patikrinami ir gerybiniai paketai persiunčiami pačiai duomenų bazių valdymo sistemai.

Tolesnio darbo plėtojimo galimybės:

1. Realizuoti abu architektūrų variantus, atlikti testavimą su jais, palyginti veikimą;
2. Suprojektuoti ir realizuoti pasyvią ir aktyvią architektūras, naudojančias anomalijomis paremtą aptikimą. Tam kartu sukurti efektyvų ir įmanomą realizuoti anomalijų aptikimo algoritmą;
3. Suprojektuoti ir realizuoti pasyvią ir aktyvią architektūras, naudojančias daugiau nei vieną įsilaužimo aptikimo metodą, efektyviai ir tinkamai integruoti įsilaužimo aptikimo metodus.

ŠALTINIAI

[BAM09] Christian Bockermann, Martin Apel, Michael Meier. *Learning SQL for Database Intrusion Detection using Context-Sensitive Modelling.* [žiūrėta 2016-04-05]. Prieiga per internetą: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.552.4993&rep=rep1&type=pdf

[BKT+05] Elisa Bertino, Ashish Kamra, Evimaria Terzi, Athena Vakali. *Intrusion Detection in RBAC-administered Databases*. [žiūrėta 2016-04-05]. Prieiga per internetą: https://www.acsac.org/2005/papers/127.pdf

[CGL00] Christina Yip Chung, Michael Gertz, Karl Levitt. *DEMIDS: A Misuse Detection System for Database Systems*. [žiūrėta 2016-04-05]. Prieiga per internetą: http://seclab.cs.ucdavis.edu/papers/IFIP99.pdf

[FOW] Firebird official website. [žiūrėta 2016-04-20]. Prieiga per internetą: http://www.firebirdsql.org/

[FSW] CVE Details. *Firebird: Security Vulnerabilities*. [žiūrėta 2016-04-20]. Prieiga per internetą: https://www.cvedetails.com/vulnerability-list/vendor_id-669/product_id-1138/Firebirdsql-Firebird.html

[Jac01] Gary Manuel Jackson. *Intrusion prevention system US 7458094 B2*. [žiūrėta 2016-04-05]. Prieiga per internetą: https://www.google.com/patents/US7458094

[KTB08] Ashish Kamra, Evimaria Terzi, Elisa Bertino. *Detecting anomalous access patterns in relational databases*. [žiūrėta 2016-04-05]. Prieiga per internetą: http://cs-people.bu.edu/evimaria/papers/vldb-journal.pdf

[Mor06] Douglas B. Moran. *Extensible intrusion detection system US 7065657 B1*. [žiūrėta 2016-04-05].
Prieiga per internetą: https://www.google.com/patents/US7065657

[MPN+10] Sunu Mathew, Michalis Petropoulos, Hung Q. Ngo, Shambhu Upadhyaya. *A Data-Centric Approach to Insider Attack Detection in Database Systems*. [žiūrėta 2016-04-05].
Prieiga per internetą: http://www.cse.buffalo.edu/~mpetropo/pubs/insider.pdf

[Row02] Craig H. Rowland. *Intrusion detection system US 6405318 B1*. [žiūrėta 2016-04-05].
Prieiga per internetą: https://www.google.com/patents/US6405318

[Roz01] Danny Rozenblum. *Understanding Intrusion Detection Systems*. [žiūrėta 2016-04-05].
Prieiga per internetą:
https://www.sans.org/reading-room/whitepapers/detection/understanding-intrusion-detection-systems-337

[RS05] Kevin Rowett, Somsubhra Sikdar. *Intrusion detection system US 20050216770 A1*. [žiūrėta 2016-04-05].
Prieiga per internetą: https://www.google.com/patents/US20050216770

[SBV14] Ricardo Jorge Santos, Jorge Bernardino, Marco Vieira. *Approaches and Challenges in Database Intrusion Detection*. [žiūrėta 2016-04-05].
Prieiga per internetą:
https://pdfs.semanticscholar.org/2eb4/e0dccae7de1b348f0795fb7d15c185a344ba.pdf

[SL05] Adrian Spalka, Jan Lehnhardt. *A Comprehensive Approach to Anomaly Detection in Relational Databases*. [žiūrėta 2016-04-05].
Prieiga per internetą: http://link.springer.com/chapter/10.1007%2F11535706_16

[SM07] Karen Scarfone, Peter Mell. *Guide to Intrusion Detection and Prevention Systems (IDPS)*. [žiūrėta 2016-04-05].
Prieiga per internetą:
http://ecinetworks.com/wp-content/uploads/bsk-files-manager/86_SP800-94.pdf

[SSM06a] Abhinav Srivastava, Shamik Sural, A.K. Majumdar. *Database Intrusion Detection using Weighted Sequence Mining*. [žiūrėta 2016-04-05].
Prieiga per internetą: http://www.jcomputers.us/vol1/jcp0104-02.pdf

[SSM06b] Abhinav Srivastava, Shamik Sural, A.K. Majumdar. *Weighted Intra-transactional Rule Mining for Database Intrusion Detection*. [žiūrėta 2016-04-05].
Prieiga per internetą:
https://pdfs.semanticscholar.org/b882/d0e710797befa8dccc08940605d2414ff225.pdf

[Sul05] Chad Sullivan. *Host Intrusion Prevention Systems: Defense-in-Depth's Best Friend.* [žiūrėta 2016-04-05].
Prieiga per internetą: http://www.ciscopress.com/articles/article.asp?p=397973

SANTRUMPOS

ANSI – American National Standards Institute.

CTE – Common Table Expression.

DBMS – Database Management System.

DBVS – Duomenų bazių valdymo sistema.

DDoS – Distributed Denial of Service.

DEMIDS – Detection of Misuse in Database Systems.

FTP – File Transfer Protocol.

IDS – Intrusion Detection System.

IETF – Internet Engineering Task Force.

IPS – Intrusion Prevention System.

MAD – Median Absolute Deviation.

NBC – Naïve Bayes Classifier.

OLAP – Online Analytical Processing.

OLTP – Online Transaction Processing.

RFC – Requests for Comments.

SIEM – Security Information and Event Management.

SNMP – Simple Network Management Protocol.

SQL – Structured Query Language.

SSO – Single Sign-On.

TCP – Transmission Control Protocol.

UDP – User Datagram Protocol.

VLAN – Virtual Local Area Network.

VPN – Virtual Private Network.

WLAN – Wireless Local Area Network.