KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Germanas Šamrickis

ELEKTRONINIŲ NUSIKALTIMŲ PĖDSAKŲ FIKSAVIMAS DEBESŲ SAUGYKLŲ KOMPIUTERIJOS APLINKOJE

Baigiamasis magistro darbas

Vadovas
Doc. dr. Jevgenijus Toldinas

KAUNAS, 2015

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

TVIRTINU
Katedros vedėjas
(parašas) Prof. dr. Algimantas Venčkauskas
(data)

ELEKTRONINIŲ NUSIKALTIMŲ PĖDSAKŲ FIKSAVIMAS DEBESŲ SAUGYKLŲ KOMPIUTERIJOS APLINKOJE

Baigiamasis magistro darbas

Informacijos ir informacinių technologijų sauga (kodas 621E10003)

Vadovas
(parašas) Doc. dr. Jevgenijus Toldinas
(data)

Recenzentas
(parašas) Doc. dr. Giedrius Ziberkas
(data)

Projektą atliko
(parašas) Germanas Šamrickis
(data)

KAUNAS, 2015

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Germanas Šamrickis

(Studento vardas, pavardė)

Informacijos ir informacinių technologijų sauga (kodas 621E10003)

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Pavadinimas“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

2 0 ___ m. _____ ____ d.

Kaunas

Patvirtinu, kad mano **Germano Šamrickio** baigiamasis projektas tema „*Elektroninių nusikaltimų pėdsakų fiksavimas debesų saugyklų kompiuterijos aplinkoje*“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

_____ _____

(vardą ir pavardę įrašyti ranka) *(parašas)*

Šamrickis, G. Baigiamojo projekto pavadinimas. *Elektroninių nusikaltimų pėdsakų fiksavimas debesų saugyklų kompiuterijos aplinkoje* baigiamasis projektas / vadovas doc. dr. Jevgenijus Toldinas; Kauno technologijos universitetas, informatikos fakultetas, kompiuterių katedra.

Kaunas, 2015. 70 psl.

SANTRAUKA

Debesies technologijos serveriai suteikia galimybę optimizuoti įmonės veiklą bei leidžia naudotis dideliais techniniais ir informaciniais resursais, būtent dėl šių priežasčių ši technologija tampa vis labiau patrauklesnė. Nors ji turi daug privalumų, tačiau yra ir trūkumų. Vienas iš jų – elektroninių nusikaltimų tyrimas debesies tipo serveriuose. Teigiama, kad nesant tinkamam sistemos auditui nusikaltimų ištyrimas tokiose serveriuose tampa ypač sudėtingu procesus, o dažnai net neįmanomu. Spręsti šią problemą dauguma autorių siūlo tokiuose serveriuose diegiant papildomas tinklo ir serverių stebėsenos sistemas, kurios gali analizuoti informacijos srautą bei pranešti apie nesankcionuotus veiksmus. Tačiau tai reškia, kad jokiais kitais būdas nusikalstamos veikos pėdsakai negali būti ištirti. Šis darbas pateikia metodą, kaip galima atkurti veiksmus, vykusius debesies serverio duomenų bazių talpyklose, analizuojant minėtų serverių sisteminius įrašus. Visi be išimties duomenų bazių serveriai generuoja sisteminius įrašus, kurie vėliau paties serverio yra naudojami atstatant duomenų vientisumą, pavyzdžiui, įvykus sistemos gedimui. Panašūs tyrimai buvo atlikti kitų autorių su MySQL serverio failas, tačiau dar niekad nebuvo atliktos studijos, kaip tai galima padaryt DB2 z/OS versijos duomenų bazės serveryje. Eksperimentinio tyrimo metu buvo analizuojami DB2 duomenų bazės serverio sisteminiai failai ir jų dėka buvo rekonstruotos serveryje vykdytos SQL INSERT/UPDATE/DELETE užklausos.

SUMMARY

Could services give companies an opportunity to optimize processes and use wide scope of system and information resources – those are the main reasons why this technology have gain huge popularity. But with those advantages comes several drawbacks. One of those is – digital forensics in could services. This been stated that if set audit policy is wage there is not many chances that it would be possible to conduct digital forensics. Some authors suggest implement additional audit tools on the server side, which would monitor informational traffic on network side and server side and alert in case of any breaches. But this also means that without presence of such system it's not possible to track back any beaches. This thesis is focused on how actions that were taken on could database systems could be backtracked by analyzing servers log files and reconstructing SQL statements from it. Nowadays all database servers logs activities on the server side in system files which then be used in crash recovery scenarios to restore data into consistent state. Some studies were conduct to show how it could be implemented on MySQL server, but none of the studies were done to show how it's possible at DB2 z/OS database server/ in. in experimental studies analysis were conduct on DB2 database log files and SQL INSERT/UPDATE/DELETE queries were successfully reconstructed

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

SANTRUMPŲ IR TERMINŲ ŽODYNAS

log failas – sisteminis serverio failas, kuriame įrašoma informacija apie skirtingus įvykius serveryje.

RBA – (*angl. Relative bite address*) vieta log faile su laiko žyme DB2 serveryje.

LRSN – (*angl. logical record sequance number*) vietal log faile su laiko žyme DB2 serverių grupėje.

LRH – (*angl. log record headar*) log failo antraštė.

URID – (*angl. Unit of recovery*) indentifikatorius, žymintis duomenų atstatymo tašką.

LG – log failo pagrindinė dalis.

RID – (*angl. Row id*) konkretaus įrašo vieta lentelės puslapyje.

DBID - (*angl. Database discriptor*) duomenų bazės diskriptorius.

OBID - (*angl. Object discriptor*) objekto diskriptorius.

z/OS – Mainframe architektūros kompiuterių operacinė sistema.

JCL – (*angl. Job control language*) skriptinė kalba z/OS operacinėje sistemoje.

REXX – skriptinė kalba IBM kompanijos operacinėse sistemose.

COBOL – procedūrine programavimo kalba.

TRACE – DB2 serverio paprogramių tipas, kuris fiksuoja įvykius, atitinkančius tam tikrus kriterijus.

DBVS – duomenų bazių valdymo sistema.

Mainframe – IBM kompanijos kompiuterių tipas, dažniausiai naudojamas didelių korporacijų ir valstybinių institucijų, skirtas laikyti ir apdoroti didelės svarbos ir didelio kiekio informaciją.

Flag – tam tikras bitas, galintis apibrėžti vykdomos operacijos tipą.

SMF – (*angl. Storage managed files*) vienas iš IBM serveriuose naudojamų failų tipų.

ĮVADAS

Šis darbas yra skirtas informatikos studijų programos informacijos ir informacinių technologijų saugos studijų pakraipai.

Informacinių technologijų revoliucija dažnai prilyginama Gutenbergo spausdinimo preso išradimui. Be jokios abejonės, informacinės technologijos kaip ir Gutenbergo išradimas, smarkiai pakeitė žmonių gyvenimo būdą. Kiekvieną dieną naujos technologijos žengia į priekį, siūlydamos naujus, su socialine erdve susietus problemų sprendimo būdus, tinkančius individualiems žmonėms, institucijoms ar verslui.

Nors naujos technologijos atveria mums naujas galimybes, kartu su jomis atsiranda ir naujos grėsmės. Nusikaltimai elektroninėje erdvėje yra nauja nusikalstamos veikos atmaina, kuri išsivystė kartu besivystant informacinėms technologijoms. Tokie nusikaltimai būna nukreipti tiek prieš individualius asmenis, tiek prieš įmones, institucijas ar organizacijas. Nusikaltėlių tikslai gali būti labai įvairus: užvaldyti kompiuterinius resursus, sutrikdyti sistemos veikimą, išgauti informaciją ir t. t. Tokių nusikaltimų tyrimo metodologija yra pakankamai gerai išvystyta, kai informaciniai resursai yra „statiški“, tačiau šiai laikais informacinių technologijų rinkos tendencijos juda link virtualizacijos paremtų sprendimų. Darbo vietos perkeliamos iš įprastų asmeninių kompiuterių į virtualias darbo vietas, panaudojant debesų kompiuterijos (*angl. cloud computing*) sprendimus. Vis dažniau pasirodo virtualizacijos produktai nešiojamiems įrenginiams – delninukams, išmaniesiems telefonams. Didėjant pasaulinei efektyvaus energijos ir infrastruktūros panaudojimo tendencijai, informacinių sistemų diegime vis plačiau naudojamos virtualizacijos technologijos. Vis daugiau įmonių bei individualių žmonių naudoja minėtus technologinius sprendimus, siekiant sumažinti įmonės kaštus. Įmonės linkusios perkelti sistemų ir informacinių išteklių administravimo naštą į debesį. Debesies technologija šiuo metu įgauna vis didesnį populiarumą dėl savo lankstumo ir prieinamumo, tačiau debesis privilioja ir nemažai žmonių, siekiančių pakenkti tokiai sistemai arba užvaldyti informacija joje, nes tokia informacija gali būti įvertinta šimtais milijonų dolerių (žiūrėti lentelėje 1[1]).

1 lentelė. IMS ir DB2 duomenų bazėse saugomų įrašų vertė

Klientų informacija	
Klientų įrašų skaičius	5000000
Procentinė dalis nutekintos klientų informacijos	3,00 %
Bendras nutekintos klientų informacijos kiekis	150000

Tiesiogiainai kaštai susieti su informacijos nutekinimu:	**Vieno įrašo kaštai**	**Bendra suma**
Paslaugų teikimas nemokamai arba su nuolaida	**$26**	**$3900000**
Informaciniai pranešimai, telefoniniai skambučiai, el.laiškai, Web, media	**$14**	**$2100000**
Teisinė gynyba ir kriminalinis tyrimas	**$7**	**$1050000**
Teisinės paslaugos, auditas, administracinė išlaidos	**$4**	**$600000**
Skambučio centro kaštai	**$3**	**$450000**
Santykiai su investuotojas ir visuomene	**$1**	**$150000**
Vidinis tyrimas	**$1**	**$150000**
Bendra suma	**$56**	**$8400000**

Netiesioginiai kaštai susieti su informacijos nutekėjimu (per vieną įrašą):	**Vieno įrašo kaštai**	**Bendra suma**
Darbuotojų produktyvumo praradimas (darbuotojai atitraukiami nuo kitų užduočių)	**$25**	**$3750000**
Projektuojamo pelno sumažėjimas (esamu klientų pasitraukimas, ribotos galimybės pritraukti naujų klientų)	**$50**	**$7500000**
Bendra suma	**$75**	**$11250000**

Bendras išlaidų kiekis:	**Vieno įrašo kaštai**	**Bendra suma**
Tiesioginiai kaštai susieti su informacijos nutekinimu:	**$56**	**$8400000**
Netiesioginiai kaštai susieti su informacijos nutekėjimu (per vieną įrašą):	**$75**	**$11250000**
Bendra suma	**$131**	**$19650000**

Būtent dėl minėtų priežasčių skaitmeninės teismo ekspertizės problema šiandien yra kaip niekada aktuali. Per pastarąjį dešimtmetį stipriai išaugo nusikaltimų, atliekamų elektroninėje erdvėje skaičius. To pasėkoje, įsikūrė daug kompanijų, kurios siūlo produktus, padedančius teisėsaugos organams nustatyti kas, ką, kur, kada ir kaip įvykdė nusikalstamas veikas, naudojantis kompiuterinėmis priemonėmis. Skaitmeninė teismo ekspertizė, kaip sfera, būtent ir išsivystė tuo pagrindu, kad būtų užtikrintas tinkamas elektroninių įkalčių pateikimas teismui. Tačiau kyla daug klausimų, ar apskritai debesies technologija gali būti tinkamai ištirta dabar jau nusistovėjusiais skaitmeninės teismo ekspertizės būdais.

Yra ir daugybė kitų problemų, susijusių su teismo ekspertizės atlikimu debesies tipo serveriuose. Drąsiai galima teigti, kad tai yra vis dar besivystanti sritis, todėl dar aktyviai kuriami įvairūs metodai jai patobulinti. Tad darbo tikslas – pasiūlyti naują metodiką debesies duomenų saugyklų skaitmeninės teismo ekspertizės atlikimo efektyvumui pagerinti. Tam, kad būtų pasiektas šis darbo tikslas, buvo įgyvendinti šie uždaviniai:

- Buvo išnagrinėti skirtingi debesies serverio paslaugos tipai ir jų saugumo problemos;

- Išnagrinėtas skaitmeninės teismo ekspertizės procesas, jo galimybės debesies technologijos aplinkoje;
- Išnagrinėjus esamą įkalčių rinkimo praktiką kompiuterizuotoje aplinkoje, buvo nustatytos silpnosios šios praktikos pusės, kartu pabrėžiant abejotiną tokio proceso priklausomumą debesies serverių technologijoms;
- Pasiūlytas naujas metodas, kaip galima atlikti skaitmeninę teismo ekspertizę debesų saugyklų kompiuterizuotoje aplinkoje;
- Darbe buvo atliktas pasiūlytos metodikos efektyvumo tyrimas.

Įkalčių rinkimas ir fiksavimas yra skirtingas debesies ir ne debesies technologijų serveriuose, dėl šios priežasties šis darbas yra ypač aktualus. Atsižvelgus į skirtingą įkalčių fiksavimo ir tyrimo poreikį debesies serveriuose, šiame darbe pasiūlytas sprendimas, kuris užtikrina, kad įkalčių paėmimo metu serverio veikla nebūtų nutraukiama. Siūlomas metodas – SQL užklausų rekonstravimas iš duomenų bazės serverio fiksuojamų sisteminių įrašų. Šis metodas gali būti taikomas ir tada, kai tokios užklausos buvo atliktos sistemos administratoriaus arba asmens, turinčio tokias teises.

Peter Fruhwirt ir Peter Kieseberg aprašė būdą, kaip galima rekonstruoti SQL užklausas iš serverio generuojamų failų. Minėti autoriai analizavo log failuose laikytą informacija. Tam jie buvo pasitelkę bitų reikšmių analizę. Šis tyrimas buvo skirtas apžvelgi tokio proceso eigą tik MySQL duomenų bazės serveryje.

Analizuojant autorių pateiktą būdą ir jo praktiką, šiame darbe pirmą kartą yra pasiūlomas ir praktiškai įgyvendintas būdas, kaip galima rekonstruoti SQL užklausas IBM DB2 z/OS duomenų bazės serveryje. Šis darbas taip pat unikalus tuo, kad darbe pasiūlytas metodas buvo ištirtas IaaS debesies serverio tipe.

Atliktas eksperimentinis tyrimas parodė, kad taikant failų baitų reikšmių analizę bei žinant duomenų bazėje esančių lentelių struktūrą, galima rekonstruoti vykdytas užklausas, net ir tas, kurios buvo vykdomos sistemos administratoriaus arba tokias teisės turinčio asmens.

Pirmoje darbo dalyje yra apžvelgiamos debesies serverių paslaugos tipai ir vystymosi tendencijos. Aptariamos šių serverių saugumo problemos ir tam tikrų autorių siūlomi šių saugumo problemų sprendimo būdai. Antroje darbo dalyje apžvelgiamas nusikalstamos veikos kompiuterizuotoje aplinkoje tyrimo procesas, jam būdingi bruožai, silpnosios pusės, pasiūlomas naujas tyrimo metodas. Trečioje darbo dalyje nuodugniau aptariamas siūlomas metodas ir nurodoma

kokiu būdu jis gali būti taikomas debesies saugyklų kompiuterizuotoje aplinkoje bei aptariami su šia tematika susieti tyrimai. Ketvirtoje darbo dalyje pateikiama eksperimentinio tyrimo eiga ir rezultatai. Galiausiai darbo pabaigoje pateikiamos išvados. Darbo prieduose yra papildoma informacija, kuri apibrėžia tam tikrų DB2 duomenų bazės serverio log failuose esančių baitų reikšmes bei tyrimo metu naudotus duomenis

1. Kas yra debesies serverio technologija?

Tam, kad būtų galima kalbėti apie debesų duomenų saugyklų saugumo problemas, labai svarbu suprasti, kas apskritai yra laikoma debesies serveriu. Šioje darbo dalyje apžvelgsime pagrindinius debesų kompiuterijos skiriamuosius bruožus.

Debesų kompiuterija vis dar yra besivystantis technologinis sprendimas. Ne vienas akademinio pasaulio bei verslo pasaulio atstovas bandė apibrėžti šią sąvoką, tačiau visą apibrėžiančios sąvokos vis dar neegzistuoja. Dažniausiai debesies technologijai apibrėžti naudojama bendrinė sąvoka, kuri buvo pasiūlyta Nacionalinio standartų ir technologijos instituto (*National institute of Standards and Technology*) -„Debesų kompiuterija, tai tam tikras technologinis modelis, kurio pasiekiamumas nepriklauso nuo vietos, turintis patogų priėjimą prie kompiuterinių išteklių, kurių konfigūraciją galima keisti reikalui esant“[2].

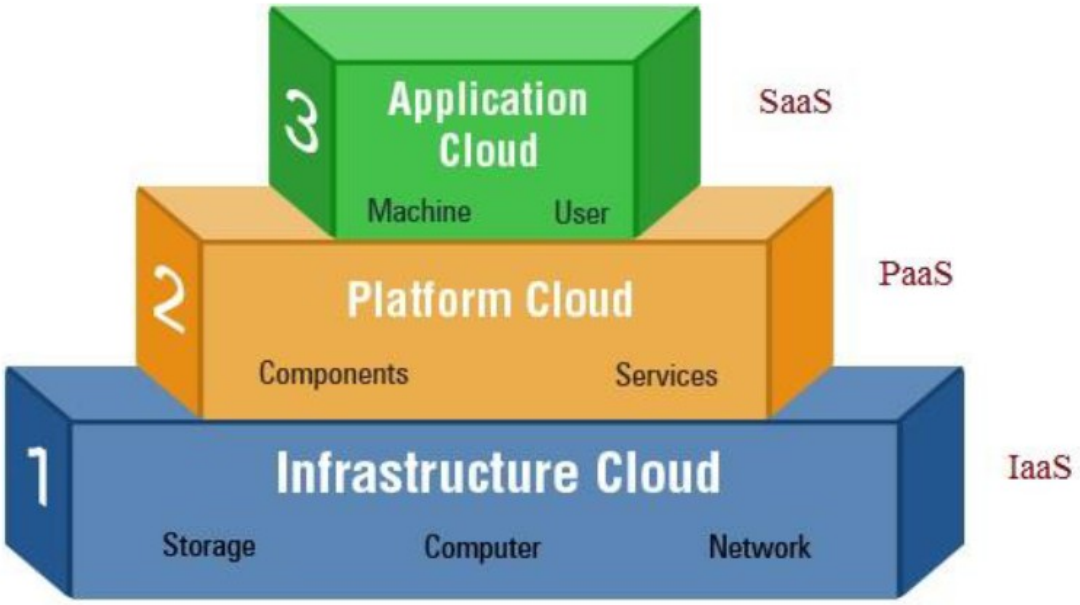
Tam, kad būtų galima paprastai paaiškinti debesies serverio veikimo principus, galima pateikti analogiją su elektros vartojimu. Kai vartotojas įjungia kabelį į rozetę, jis nesirūpina kaip elektra yra pagaminama bei perduodama jam tinklu. Elektra yra virtualizuojama tam konkrečiam vartotojui, kitais žodžiais tariant, elektra vartotojui pasiekiama nuo sienoje esančios rozetės, kurioje slypi visos su elektros perdavimu susietos komplikacijos. Šiame pavyzdyje įmonei priklauso elektros generavimo, gamybos palaikymo ir elektros distribucijos atsakomybė, tuo tarpu vartotojas naudojasi jam suteikiama elektros energija ir neprisiima atsakomybės dėl elektros gamybos ar paskirstymo. Taip pat, kaip ir elektros gamybos atveju, taip ir debesies technologijos atveju, vartotojas naudojasi jam pasiekiamais informacinių technologijų ištekliais (pavyzdžiui, tam tikromis programomis, duomenų saugyklomis) taip išvaduojant save nuo būtinybės turėti visam tam reikiamą infrastruktūrą, suprasti jos veikimą arba rūpintis jos išlaikymu. Už tokią paslaugą vartotojas sumoka atitinkamą kainą, kuri priklauso nuo debesies tipo.

Pagal NIST, debesų kompiuterijos paslauga susideda iš penkių jai būdingų bruožų:

1. Savitarnos paslaugos “pagal pareikalavimą“;
2. „Plati“ tinklo prieiga;
3. Resursų dalinimasis su kitais vartotojais iš bendro fondo;
4. Spartus naudojamų paslaugu masto keitimas (elastiškumas);
5. Išmatuojamumas[2].

Kompanijos, užsisakydamos debesies serverio paslaugas, gali rinktis iš 3-jų paslaugos modelių (žiūrėti pav. 1.1.):

- „Programinė įranga kaip paslauga“ (angl. *Software as a Service* – SaaS): klientas naudojasi programine įranga, kuri veikia tiekėjo debesies infrastruktūroje;
- „Platforma kaip paslauga“ (angl. *Platform as a Service* – PaaS): klientas gali pats įsidiegti reikiamą programinę įrangą debesyje, kuri sukurta naudojantis tokiomis programavimo kalbomis, bibliotekomis, paslaugomis ir įrankiais, kuriuos palaiko paslaugos tiekėjas;
- „Infrastruktūra kaip paslauga“ (angl. *Platform as a Service* – PaaS): suteikiamos galimybės klientams patiems susiformuoti ir naudoti reikiamus procesoriaus, atminties, tinklų ir kitus fundamentalius kompiuterių resursus, kuriuose klientas gali diegti ir leisti programinę įrangą, tame tarpe ir operacines sistemas[2].



1.1. pav. Skirtingų debesies tipų palyginimas[2]

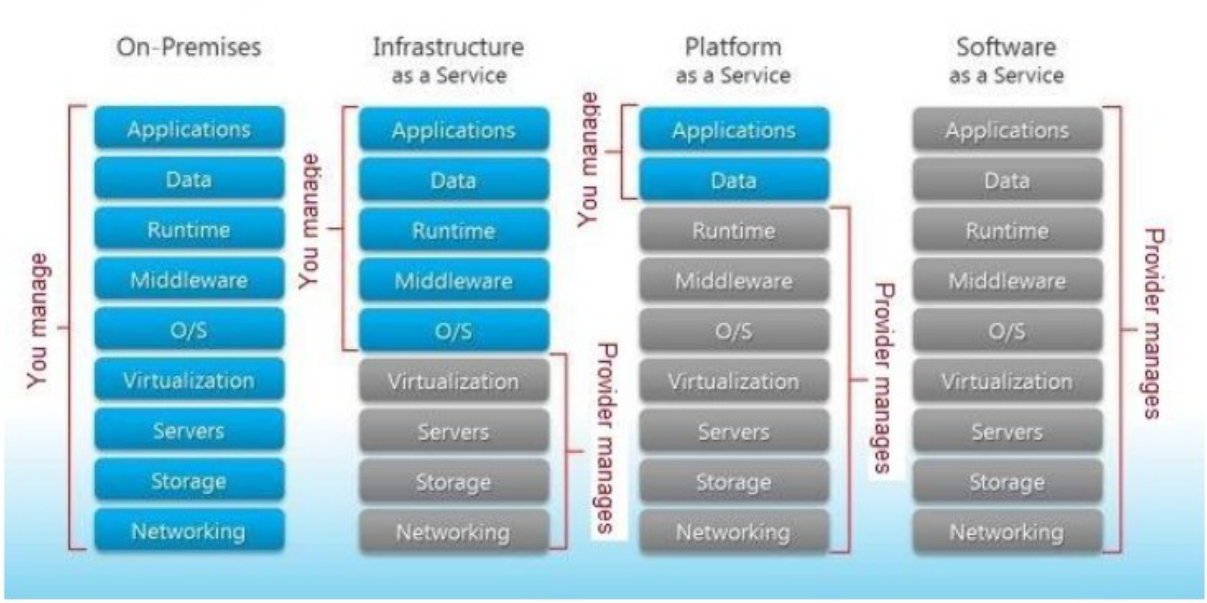
Galiausiai yra galimi keturi debesies tipo serverio diegimo modeliai:

- Privatus, kai debesies infrastruktūra naudojama išskirtinai tik vienintelės organizacijos;
- Bendruomeninis, kai paslaugas naudoja specifinė kelių klientų grupė;
- Viešas, kai debesis yra prieinamas bendrajai publikai;

- Hibridinis, kai apjungiami keli paminėti diegimo modeliai

Su kiekvienu anksčiau paminėtu debesies modeliu susieti ir skirtingi saugumo lygiai, kurie yra prieinami vartotojui arba už kurios vartotojas tiesiogiai yra atsakingas. Iš visų modelių didžiausią saugumo garantiją turi SaaS debesies tipas, tačiau nors šis debesies tipas ir suteikia didesnį saugumą, t. y. saugumo lygį kuriuo vartotojas pats neturi rūpintis, bet praktiškai nepalieka lankstumo, arba galimybės išplėsti paslaugą. Savo ruožtu PaaS suteikia didesnį lankstumą vartotojams. Programų kūrėjai gali kurti taikomąsias programas ant platformos „viršaus“, tačiau šiuo atveju sumažėja integruotų saugumo svertų kiekis teikiamoje paslaugoje. Jei vartotojui reikia papildomų saugumo priemonių, dažniausiai už tai papildomai yra sumokama debesies tiekėjui. IaaS suteikia labai plačias galimybės vartotojui. Šioje aplinkoje jiems leidžiama beveik nevaržomai kurti taikomąsias programas ar kitokius įvesties ir išvesties sprendimus. Tačiau šiuo atveju ženkliai sumažėja integruotų saugumo įrankių debesyje.

Paveiksle žemiau grafiškai atvaizduotas skirtumas tarp skirtingų debesies serverių tipų.



1.2. pav. Saugumo lygių palyginimas pagal debesies tipą[3]

1.1. Duomenų saugykla kaip paslauga (DBaaS)

Šiuo metu gan naujas debesų kompiuterijos paslaugos tipas įgauna vis didesnį populiarumą, tai duomenų saugykla kaip paslauga (angl. *database as a service - DBaaS*). Galima sakyti, kad DBaaS paslaugos tipas yra subkategorija PaaS ir SaaS paslaugų priklausomai nuo to, kokį technologinį sprendimą naudoja paslaugos tiekėjai. DBaaS paslaugos tiekėjai suteikia vartotojams jiems reikiamu metu prieinamos duomenų saugyklos paslaugą, kuri gali būti pasiekiama internetu. Esama paslauga labai palengvina su verslu susietas tam tikras operacijas, nes paslaugos gavėjas neturi rūpintis paslaugos prieinamumu ar su šios duomenų saugyklos susietais administraciniais ir priežiūros iššūkiais, taip pat tokios paslaugos instaliavimu, migravimu iš vienos versijos į kitą, naujų programinių paketų instaliavimu susietais kaštais[4].

Prie debesies duomenų bazės privalumų galima paminėti, jog tokia duomenų bazė yra labai lanksti. Reikalui esant jos dydį galima padidinti be didelių techninių sunkumų, kurie galėtų atsirasti tokią duomenų saugyklą palaikant lokaliai. Šiais laikais vis didėja multimedijos duomenų naudojimas, todėl duomenų bazių dydis gali siekti net kelis terabaitus. Vartotojui tokios lokalios duomenų bazės išlaikymas atsietų labai brangiai, taip pat reikalautų daug žmogiškųjų išteklių, valdant visus procesus, susietus su staigiu duomenų saugyklos augimu - kaip tik šioje vietoje debesų duomenų saugyklos tampa ypač patrauklios. Vartotojui nereikia rūpintis su duomenų saugojimu susietais kaštais. Debesyje vartotojo duomenų saugyklos duomenys paskirstomi po atitinkamus serverius su adekvačiais resursais.

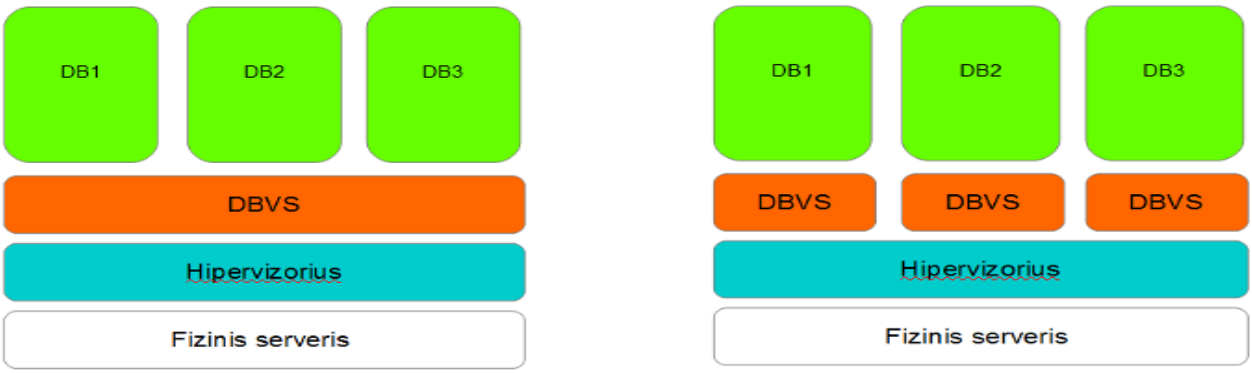
Galimi du duomenų bazės perkėlimo į debesį scenarijai:

- Padaryk tai savarankiškai (angl. *do-it-yourself*). Tokiu atveju vartotojas įdiegia savo duomenų saugyklos valdymo sistemą tiekėjo debesies platformoje. Visus su duomenų saugyklos administravimu, instaliavimu susietus veikslus atlieka pats vartotojas. Šiuo atveju toks sprendimas gana patogus, nes leidžia debesies vartotojui instaliuoti papildomus audito, monitoringo ir kitus produktus jo naudojamame debesyje.
- Pilno funkcionalumo DDaaS. Skirtingai nei anksčiau minėtu atveju, vartotojas perka pilną duomenų saugyklos paslaugą iš jos tiekėjo. Visi su administravimu, palaikymu susietų klausimų sprendimas atitenka paslaugos tiekėjui. Vienintelis dalykas, kuris reikalingas iš vartotojo – jo naudojamų programų prijungimas prie debesies duomenų saugyklos. Kai programos bus prijungtos prie duomenų saugyklos, vartotojas bus apmokestintas pagal tuos

duomenų saugyklos naudojimo kiekį. Jos talpinimas debesyje yra labai patogus vartotojui, tačiau dažnai vartotojas negali instaliuoti papildomų produktų debesyje, tokiu atveju vartotojui telieka pasikliauti paslaugos tiekėju, kad jis tinkamai pasirūpins duomenų apsauga[5].

1.1.1. Vienos instancijos modelis ir VS kelių instancijų modelis

Be paties diegimo debesyje pobūdžio, tokios saugyklos turi ir kitus, ne ką mažiau svarbius tarpusavio skirtumus. DBaaS aplinka gali gerokai skirtis viena nuo kitos. Kai kurie paslaugų tiekėjai siūlo kelių duomenų bazių instancijos modelį per vieną duomenų bazių valdymo sistemą, kiti gi siūlo visą duomenų bazės valdymo sistemą su duomenų bazėm priskirti vienam vartotojui. Kitais žodžiais tariant, pirmuoju atveju prie vienos duomenų saugyklos valdymo sistemos bus prijungtos kelios duomenų saugyklų subsistemos, kurios dalinsis bendrais ištekliais, antruoju atveju prie vienos saugyklos bus prijungta viena duomenų saugyklos subsistema, kuri naudosis tik jai dedikuotais ištekliais. Šiuo atveju vartotojas gali geriau administruoti prieigos autorizaciją, priskirti teises ir roles vartotojams. Tačiau iš kitos pusės vienos instancijos modelis duomenims duomenų bazėse priskiria žymę, kuri yra unikali kiekvienam vartotojui toje subsistemoje. Tokioje aplinkoje prieigos autorizavimu rūpinasi paslaugos tiekėjas. Iš esmės kelių instancijų modelis yra daug patikimesnis ir dažniausiai siūlomas tiekėjų, nes tokiame modelyje yra daug paprasčiau įgyvendinti tam tikrus saugumo sprendimus, pavyzdžiui, duomenų šifravimą (žiūrėti pav. 1.3.).



1.3. pav. Vienos instancijos modelis VS kelių instancijų modelis

1.2. DBaaS saugumo problemos

Nors minėtoje debesies kompiuterijos teikiamoje paslaugoje yra nemažai pliusų, tačiau yra tam tikrų su saugumu susietų spragų, kurios būdingos DBaaS paslaugai. Verta paminėti, kad šios spragos būdingos visoms duomenų bazėms esančioms debesies tipo serveriuose. Taigi, prieš migruojant duomenų saugyklą į debesį turi būti užtikrinama atitinkama šios duomenų saugyklos apsauga. Šios apsaugos priemonės turi apimti duomenų konfidencialumą, vientisumą ir pasiekiamumą. Pagrindiniai saugumo sprendimai turi būti susieti su duomenų apsauga duomenų perdavimo metu, duomenų naudojimo metų ir duomenų statinio buvimo duomenų bazėje metu, taip pat prieiga prie duomenų bazės turi būti kontroliuojama. Todėl galima teigti, jog:

- Tam, kad duomenys nebūtų sugadinti arba perimti trečiųjų asmenų, duomenų perdavimas iš debesies į debesį turi būti vykdomas pagal saugias ir patikimas procedūras;
- Tam, kad būtų užtikrintas duomenų konfidencialumas, duomenys patalpinti debesų duomenų saugykloje visada turi būti šifruojami;
- Tam, kad būtų užtikrintas duomenų vientisumas, prieiga prie duomenų debesyje turi būti prižiūrima debesies duomenų saugyklos tiekėjo platformoje, vykdant atitinkamą sistemos monitoringą bei administravimą duomenų centre.

Tačiau visas šias būtinas sąlygas ne visuomet pavyksta įvykdyti. Neretai pasitaiko problemų susietų su duomenų pasiekiamumu. Tai yra vienas iš kritinių saugumo aspektų, į kuriuos debesies vartotojai turi atsižvelgti. Sutrikus sistemos veiklai, duomenų pasiekiamumas debesyje gali sutrikti laikinai arba galima visam laikui prarasti prieigą prie duomenų. Tam įtakos gali turėti daugybė veiksnių, pavyzdžiui, DOS atakos, įrangos gedimai arba stichinės nelaimės.

Kita saugumo problema susieta su duomenų saugyklomis debesyje – prieigos kontrolė. Neretai perkeliant duomenų saugyklą į debesį prarandama prieigos kontrolė. Vartotojai, kurie perkelia jiems kritinę informaciją į debesį, praranda fizinę, loginę kontrolę, tai pat praranda tų asmenų kontrolę, kurie anksčiau rūpinasi tais duomenimis. Nors išorės atakos kelia didelė grėsmę saugumo užtikrinimui, tačiau neseniai vykdyti tyrimai parodė, kad didžiausia grėsmę duomenų

saugyklų saugumui kyla iš organizacijos vidaus (angl. *Insiders attack*). Tas taikytina kiek debesies vartotojo organizacijai, tiek ir paslaugos tiekėjo organizacijai[6].

Sistemos auditas - tai dar vienas problematiškas saugumo aspektas, su kurio susiduriama perkeliant duomenų saugyklas į debesį. Duomenų saugyklos elastingumas ir lankstumas, t. y. kokiu laipsniu duomenų kiekis debesies saugykloje gali didėti ir kaip lengva tą didėjimą administruoti, yra vienas iš debesies technologijos pliusų. Tačiau tam, kad visada būtų patenkintas toks staigus duomenų saugyklų dydžio kitimas, dažnai tenka tai stebėti ir administruoti debesies tiekėjui. O tai reiškia, kad šių saugyklų prižiūrėjimas ir aprūpinimas dažnai vyksta be vartotojo žinios. Dar daugiau, tam kad būtų patenkinamas nuolatinis sistemos pasiekiamumas, vartotojų duomenis replikuojami tarp skirtingų serverių ir skirtingų lokacijų. Todėl tokia infrastruktūra yra laikoma dinamiška ir vartotojas neturi jokio supratimo ir priėjimo prie fizinės infrastruktūros. Tai kaip gi šiuo atveju visą tai susieta su saugyklos saugumu? Didžioji dauguma tradicinių sistemos monitoringo ir apsaugos priemonių reikalauja pilnos informacijos apie tinklo tipologiją arba reikalinga prieiga prie fizinės duomenų saugojimo vietos. Visais šiais atvejais dinamiškas debesies pobūdis tokius tradicinius įrankius daro nepraktiškais ir visiškai netinkančiais tokiai infrastruktūra stebėti[7]. Tam, kad tokie įrankiai pasiteisintų, nuolatos reikia keisti šių įrankių konfigūraciją, kad jie atitiktų duomenų išdėstymą naujuose serveriuose. Taip pat kažkokios programinės įrangos instaliavimas tiesiogiai į duomenų talpyklas (angli. *storage*) būtų sunkiai įgyvendinamas[6].

Šioje vietoje galima išskirti kelias problemines sritis, kurios kartais labai komplikuoja įsilaužimų tyrimą debesies duomenų bazių serveriuose. Šios problemos neatskiriamos beveik nuo visų tipų debesies tipo serverių:

- Serverio operatyvios atminties dydis;
- Dedikuotos duomenų talpyklos dydis;
- Serverio sisteminių failų replikavimas;
- Serverio failų atsarginių kopijų kiekis.

Vykdant nusikalstamos veikos tyrimą debesies duomenų bazių serveriuose, pagal tradicinę skaitmeninės ekspertizės tyrimo sampratą, būtų daromos serverio operatyvios atminties kopija (angl. *Raw copy*) ir dedikuotos talpyklos kopija. Savo ruožtu tai turėtų neigiamų padarinių sistemos pasiekiamumui, o nepertraukiamos sistemos prieinamumas yra viena iš pagrindinių debesies tipo serverio naudojimo sąlygų. Kitu atveju šios paslaugos naudojimas būtų beprasmis.

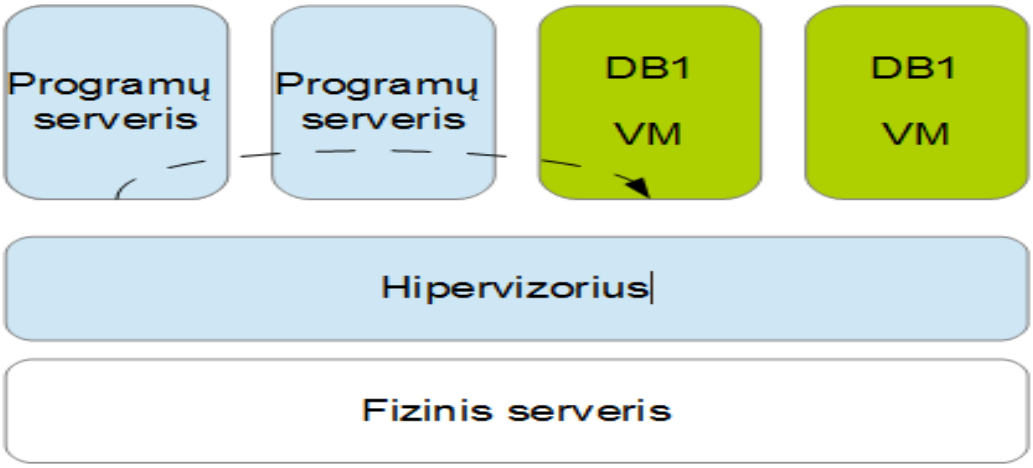
1.2.1. Debesies duomenų saugyklų monitoringas

Duomenų bazių monitoringas ir auditas sąlygojamas galimybės nepertraukiamai fiksuoti visus įvykius, vykstančius duomenų bazės sistemoje. Audito duomenis generuojami remiantis tuo, kas, kaip ir kada buvo priėjęs prie skirtingų duombazės objektų ar buvo juos pakeitęs. Tokio audito įrankio buvimas debesies duomenų bazėje, leidžiantis stebėti ir fiksuoti tokius įvykius, kuriuos vėliau būtų galima panaudoti vykdant teismo ekspertizę, nepriklausomai nuo lokacijos, yra būtinas debesies duomenų bazėms.

Tradiciškai, kai duomenų bazė yra laikoma lokaliame serveryje, tam tikra programinė įranga turėtų būti įdiegta kažkur tinkle, kas leistų stebėti protokolų pažeidimus, fiksuoti galimus virusus ir t.t.. Tačiau tokiais atvejais anksčiau dažnai buvo ignoruojamos arba tiesiog užmirštamos grėsmės, kylančios iš pačios organizacijos vidaus [8]. Šiais laikais dažniausiai pasitaiko hibridiniai saugumo sprendimai, kurie tam tikrą judėjimą lokaliame tinkle siunčia į bendrus tinklo stebėsenos įrankius vykdyti duomenų analizei, kur kiekviena tokia transakcija būtų lyginama su iš anksto nustatyta saugumo politika. Tai nėra pats geriausias sprendimas, bet jis yra dažnai naudojamas įmonėse.

Debesies duomenų saugyklos atveju toks tinklo „klausymasis“ pasidaro neįmanomas dėl techninių iššūkiu, nes įrenginiai yra už organizacijos infrastruktūros ribų. Taip pat patys duomenys debesyje, tenkinant prieinamumo sąlygą, gali atsidurti vis naujoje fizinėje lokacijoje. Todėl tokie tradiciniai sistemos stebėsenos būdai netinka dinamiškai debesies struktūrai.

Kitas argumentas, kodėl tradiciniai tinklo stebėsenos būdai nėra tinkantys debesies duomenų saugykloms, siejamas su virtualizacijos technologija. Anksčiau programos, kurioms buvo reikalinga prieiga prie duomenų saugyklos ,buvo talpinamos atskirame serveryje, kai pati duomenų saugykla buvo talpinama kitame fiziniame serveryje. Tačiau virtualizacijos dėka fiziniais resursais yra dalinamasi debesyje ir dažnai kiek programa, tiek pati duomenų bazė būna patalpinta tame pačiame fiziniame serveryje (žiūrėti pav. 1.4.). Tokiu atveju tinklo monitoringo įrankiai negalėtų aptikti tokių transakcijų, nes komunikacija būtų vykdoma tarp virtualių mašinų[9].



1.4. pav. Duomenų saugyklų komunikavimas virtualizuotose aplinkose

Tam, kad galima būtų vykdyti sistemos monitoringą, galima stebėti per vieną virtualią mašiną visas tinklų siunčiamas transakcijas. Tačiau toks sprendimas smarkia sulėtintų bendrą sistemos veikimą. Todėl dažnai yra siūloma naudoti duomenų bazės agentus, kurie veiktų kartų su duomenų bazių valdymo sistemos vienetu. Minėtieji agentai fiksuoja transakcijų siunčiamas užklausas, esant reikalui teikia informacinius pranešimus apie sistemoje apibrėžtos saugumo politikos pažeidimus. Visą informaciją apie vykdomos procesus duomenų saugykloje jos agentas siunčia į centrinį serverį, kur ši informacija saugoma. Siuntimo metu tokia informacija šifruojama. Vėliau, vykdant teismo ekspertizę, autorizuoti asmenys gali tokią informaciją iššifruoti ir išanalizuoti,ieškant nusiklastomos veikos fakto patvirtinančių įrodymų.

1.2.2. Debesies įsilaužimų monitoringo sistemos tipai ir architektūra

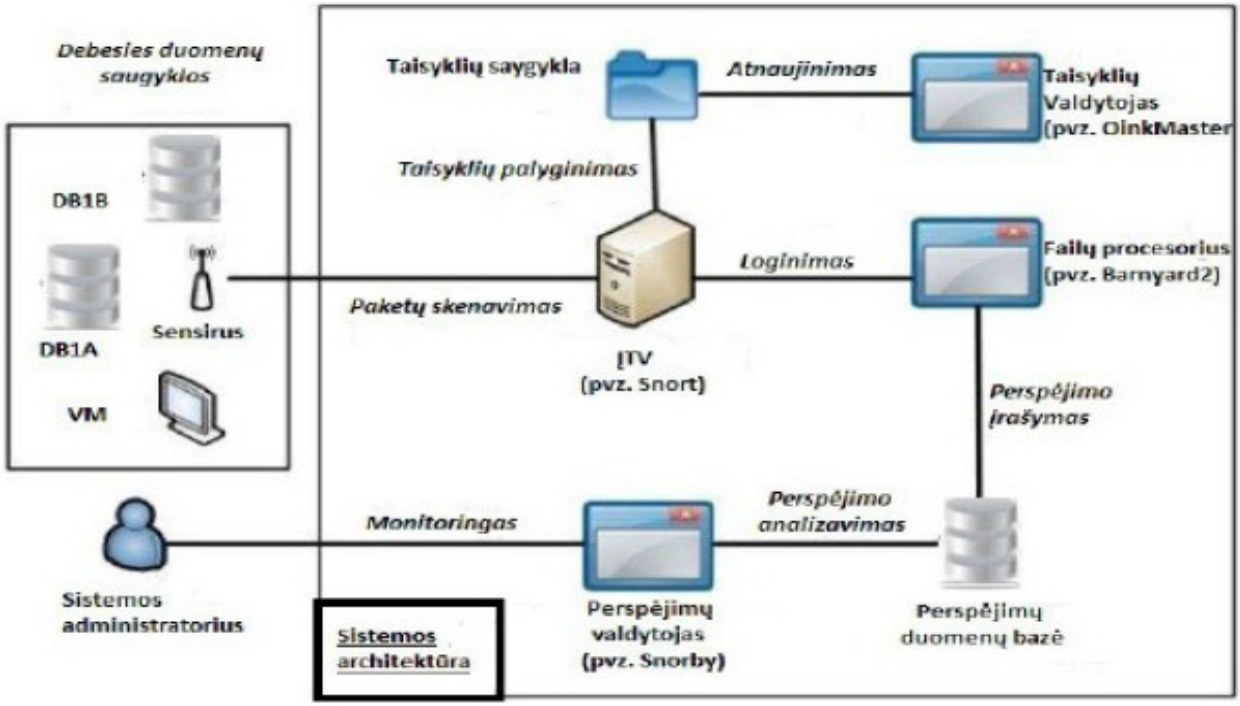
Įsilaužimus fiksuojančių sistemų diegimas (angl. *Intrusion detection system IDS*) debesies technologijose buvo ganėtinai neblogai ištyrinėtas kelių autorių. Tačiau, kaip ir buvo kalbėta ankstesnėse šio darbo dalyse. Debesies paslaugos (angl. *Cloud service*) yra kelių rūšių ir kiekviena iš tų rūšių reikalauja skirtingo priėjimo.

Sistemos skirtos įsilaužimų fiksavimui debesų kompiuterijos architektūroje[10] buvo išvystytos siekiant atlikti globalų tinklų resursų monitoringą. Toks sistemų monitoringas leisdavo atskleisti koordinuotas atakas, nukreiptas prieš lokalius serverius. Tokios dvipolės sistemos architektūra susideda iš dviejų dalių – lokalios ir globalios. Globalios srities, kitais žodžiais tariant, debesies tikslas yra surinkinėti informaciją arba įspėjamuosius pranešimus, sugeneruotus lokaliose serveriuose. Kai galima grėsmė yra identifikuojama globaliame serveryje, apie tai yra įspėjamas lokalaus serverio administratorius, kuris vėliau ir imsis tam tikrų veiksmų, pavyzdžiui, įtrauks į juodąjį sąrašą (angl. *blacklist*) atakos šaltinį (angl. *Source*). Tokia sistema yra labiau tinkama privačių debesų technologijoms, kurios susinchronizuotos privačiai komunikuoti su lokaliais ir nutolusiais serveriais. To pasėkoje lokalių serverių administratoriai yra labai priklausomi nuo debesies paslaugos tiekėjo ir jo galimybių identifikuoti galimus įsilaužimus. Taip pat toks globaliu ir lokaliu serveriu administravimas sukelia kitų, papildomų saugomo problemų.

Mazzariello savo darbe aptarinėja įvairius įsiskverbimo sistemos diegimo metodus atvirojo kodo debesies (*angl. open source cloud*) aplinkoje[11]. Pagal jo siūlomą modelį kelios įsiskverbimo fiksavimo sistemos diegiamos kiekvieno debesies fiziniame kontrolieriuje, kas leidžia fiksuoti tik tam tikrą dalį tinklo srauto, skirto tik tam tikram skaičiui virtualių mašinų. Toks siūlomas modelis reikalauja labai nemažai pastangų konfigūruojant fizinius įrenginius.

Kiti tyrimai, kalbantys apie debesies įsilaužimo fiksavimo koncepcijas, siūlo dalinai vartotojų valdoma įvykių fiksavimo sistemą. Toks techninis sprendimas susidaro iš kelių sensorių suinstaliuotų debesyje ir centrinio valdymo „pulto“[12]. Toks spendimas gali būti diegiamas visų tipų debesies technologijose. Drąsiai galima teigti, kad toks sprendimas yra daugiasluoksnis, nes jo dėka atliekamas tiek tinklų siunčiamų paketų t.y. prieigos iš kitų debesies sluoksnių, tiek prieigos iš tame pačiame debesyje monitoringas. Tokiame sprendime centrinis įsilaužimo fiksavimo modulis surenka informaciją iš centrinių modulių, ja analizuoja, gali išvesti tam tikras koleriacijas. Taip pat debesies vartotojai gali konfigūruoti prieigos profilius/parametrus. Tokio tipo įsiskverbimus fiksuojančioje sistemoje vartotojai didžiąja dalimi yra priklausomi nuo debesies paslaugos tiekėjo turimos tokios sistemos infrastruktūros, bei nuo to, kaip suinstaliuota tokia sistema ir kurioje vietoje tinkle toks įrenginys yra įdiegtas, taip pat rimtų saugumo problemų kelia įsiskverbimus fiksuojančios sistemos instaliavimas ant kiekvienos iš kliento naudojamos virtualios mašinos.

Pavyzdyje žemiau pateikiamas vizualizuota tokios sistemos struktūra.



1.5. pav. Debesų su įsiskverbimų fiksavimo sistema pavyzdys

1.2.3. IDSaaS sistemos ypatumai

Tokio tipo sistemos egzistavimas debesyje –privačiame arba viešame, vartotojams atneša eilę privalumų, pavyzdžiui:

- Elastingumas – augant infrastruktūroms poreikiams yra gan paprasta padidinti tokių sistemų skaičių;
- Mobilumas – ši sistema, renkanti informaciją apie vykdomus veiksmus vertualioje mašinoje, gali būti suinstaliuota tiek privačiame tinke, tiek viešame debesyje arba skirtinguose regionuose tame pačiame debesyje;
- Pilna kontrolė – šios sistemos administravimas yra pilnai atsietas nuo debesies tiekėjo, pavyzdžiui, administruojant tokią sistema galima deaktyvuoti arba aktyvuoti pavienius suinstaliuotus monitoringo agentus;
- Adaptuoti parašai – šios sistemos vartotojai gali sukurti savus parašus, kurie būtų naudojami

kaip sistemos identifikatoriai, kurie leistų nuspręsti, ar atliekami veiksmai yra teisėti, ir ar užklausos šaltinį reikia blokuoti (žiūrėti pav. 1.6.)

	Signature
Grammar	[action] [protocol] [src_ip] [src_port] [direction] [dist_ip] [dist_port] [option]
Application Level	alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (uricontent:".php";pcre:"/(\%27)|(\')|(\-\-)|(%23)|(#)/"; msg:"SQL Injection Attack"; sid:1000005;rev:1;)
System Level	activate tcp any any -> $HOME_NET 22 (content:"/bin/sh"; activates:1; msg:"Possible SSH buffer overflow"; sid:1000023;) dynamic tcp any any -> HOME_NET 22 (activated_by:1; count:10;)
Network Level	drop tcp $EXTERNAL_NET any -> $DB_NET 3306 (msg:"Unauthorized Access to DB server"; content:"mysql -p"; nocase; sid:1000019;)

1.6. pav. Parašų pavyzdžiai

- Sistemos patikimumas - visi sistemos veiksmai, kurių monitoringas yra atliekamas, yra įrašomi į pasirinktą saugią vietą, daromos atsarginės kopijos.

Akivaizdu, kad tokia sistema turi daug privalumų. Jos dėka ne tik galima stebėti veiksmus, vykdomus sistemoje, bet užkirsti kelia piktavališkiems veiksmams. Tačiau esminis šios sistemos trūkumas, kaip paradoksaliai tai beskambėtų - šios sistemos buvimas arba nebuvimas. Tam, kad ši sistema neštų naudą, ją reikia instaliuoti debesies serveryje, ją konfigūruoti, apdoroti surinktą informacija, kitas minusas – beveik visos tokios sistemos nėra skirtos įkalčių surinkimui, t. y. nėra sertifikuotos vykdyti tokia funkciją. Įkalčių rinkimo procesas yra labai svarbus procesas ikiteisminiame procese, nes šio proceso metu surinkti įkalčiai galės/negalės būti panaudojami teisme.

Taigi, koks gi yra tas įkalčių rinkimo procesas? Kaip tie įkalčiai yra surenkami? Kaip atrodo elektroninių nusikaltimų tyrimo procesas ir kuo jis skiriasi/nesiskiria nuo stacionaraus tyrimo ir tyrimo debesies tipo serveriuose. Visa tai bus aptariamo kitoje šio darbo dalyje, kurioje taip pat

bus pateikta metodologija, skirta nusikaltimų tyrimui debesies duomenų bazių serveriuose.

1.3. Išvados

Išanalizavus skirtingus debesies tipus ir būdus, kaip galima atlikti tokių sistemų monitoringą, bei kokios yra tokių sistemų saugomo spragos, galima daryti šias išvadas:

- Debesies technologija suteikė įmonėms naujų būdų kaip galima sumažinti savo veiklos kaštus, padidinti įmonės veiklos efektyvumą ir našumą, naudojantis debesies serverių ištekliais, todėl ateityje įmonių, naudojančių šį technologinį sprendimą tik daugės;

- Debesies tipo serveriai turi savo reikalavimus iš vartotojų pusės. Įmonė, teikianti tokias paslaugas, turi užtikrinti, kad serveris visada bus pasiekiamas paslaugos gavėjams, kitu atveju tai grėstų reputacijos ir pelno praradimu. Vykdant nusikalstamos veikos tyrimą tokio tipo serveryje tradiciniais būdais neįmanoma užtikrinti nepertraukiamo jo pasiekiamumo;

- Daugelis autorių susidariusią saugumo spragą siūlo spręsti įdiegiant papildomas monitoringo sistemas tokiuose serveriuose;

- Papildomos monitoringo sistemos yra gana kompleksinės ir reikalauja papildomų serverio išteklių. Jos nors ir geba atkurti nusikalstamos veikos pėdsakus, tačiau jei jų serveryje neegzistuoja, įkalčių praktiškai nėra galimybės atkurti;

- Vienas iš pagrindinių nusikalstamos veikos tyrimo debesies tipo serveriuose trūkumas – didelis neapibrėžtumas, kitais žodžiais tariant – informacijos apie sistemos išteklius turėjimas.

Kitoje darbo dalyje apžvelgiama kibernetinių nusikaltimų samprata, kokios yra tokio tipo nusikalstamos veikos tyrimo gairės. Taip pat bus įvertinta esama kibernetinių nusikaltimų įkalčių paėmimo metodika, tuo siekiama parodyti, kokios yra šios metodikos silpnosios vietos, taikant ją nusikalstamos veikos tyrimams debesies tipo serveriuose. Išsiaiškinus minėtas silpnąsias vietas, bus pasiūlytas naujas metodas, leidžiantis ištirti minėtąsias nusikalstamas veikas.

2. Elektroninių nusikaltimų pėdsakų fiksavimas debesų saugyklų kompiuterijos aplinkoje

2.1. Kibernetinių nusikaltimų klasifikacija

Prieš pradedant kalbėti apie kibernetinius nusikaltimus (angl. *cyber crime*), reikėtų pirma šiek tiek apžvelgti, ką apskritai reiškia toks terminas. Gali pasirodyti, kad tai nėra labai svarbus aspektas mūsų tyrime, tačiau nusiklastomos veikos kibernetinėje aplinkoje apibrėžimas turi labai didelę svarbą nusikalstamos veikos tyrimui (angl. *cyber forensics*) ir galutiniam juridiškai pagrįstam nuosprendžiui. Esant nusikalstamos veikos faktui, nustatomos visuotinai priimtos metodologijos, kurios bus naudojamos atliekant tyrimą, taip pat pasirenkami sertifikuoti įrankiai tam tyrimui atlikti. Jei nusikalstamos veikos tyrimas bus atliekamas naudojant visuotinai nepripažintą metodą arba nesertifikuotą įrankį, tokiu atveju nuteisti asmenį, įvykdžiusį tokio pobūdžio veiklą, yra neįmanoma[13].

Taigi, kas yra kibernetinis nusikaltimas? Vystantis technologijoms vystėsi ir paties kibernetinio nusikaltimo apibrėžimas. Vienas iš pirmųjų tokių apibrėžimų atsirado dar septyniasdešimtaisiais metais. Tais laikais kibernetinis nusikalstamumas buvo apibrėžiamas ir suvokiamas kaip:

- Kompiuteris yra objektas, arba duomenys kompiuteryje yra nusikalstamos veikos objektas;
- Kompiuteris sukuria unikalia aplinką arba turtą;
- Kompiuteris yra nusikalstamos veikos instrumentas arba priemonė;
- Kompiuteris veikia kaip simbolis, siekiant padaryti įtaką arba įbauginti[14].

Kiek vėlesniu laiku Robert Taylor pateikė savo, labiau apibendrintą apibrėžimą. R.Taylor išskyrė keturis kibernetinio nusikalstamumo dėmenis:

- Kompiuteris kaip atakos taikinys. Atakuojantis asmuo siekia užkirsti kelią teisėtiems vartotojams prieiti prie kompiuterinių išteklių. Prie tokio kibernetinio nusikalstamumo galima priskirti tokias kibernetines atakas kaip DOS atakas;
- Kompiuteris kaip nusikalstamos veikos įrankis. Tokios nusikalstamos veikos tikslas yra užvaldyti informaciją, kurią vėliau galima būtų panaudoti kitoms nusikalstamoms veikoms vykdyti kaip, pavyzdžiui, identiteto vagystės;

- Kompiuteris kaip netiesiogiai susietas objektas vykdant nusikalstama veiką. Prie tokių veikų galima priskirti su pinigų plovimu susietas nusikalstamas veikas.
- Nusikalstamos veikos susietos su kompiuterinių ir/arba kompiuterinės technikos paplitimu. Tokios nusikalstamos veikos gali būti nukreiptos prieš kompiuterių industrijoje veikiančias įmones, nusikaltimai susieti su intelektualia nuosavybe, piratiniu programų platinimu[15] .

Atsižvelgiant į nusikalstoms veikos pobūdi yra pasirenkamas teisminės ekspertizės metodas arba apibendrintos gairės, kurios padėtų tinkamai surinkti duomenis apie nusikalstama veiką ir tinkamai ją išanalizuoti ir interpretuoti.

2.2. Skaitmeninė teismo ekspertizė

Kada kalbame apie nusikalstamą veiką ir jos atkūrimą elektroninėje erdvėje, susiduriame su skaitmeninės teismo ekspertizės sfera (dar žinoma kaip kompiuterių ir jų tinklų teismo ekspertizė).

Skaitmeninė teismo ekspertizė turi daug apibrėžimų, bet plačiąja prasme tai yra „mokslo panaudojimas, siekiant identifikuoti, surinkti, ištirti ir išanalizuoti duomenis išlaikant jų vientisumą ir išsaugant griežtą „arešto grandinę“ (angl. *chain of custody*)[16].

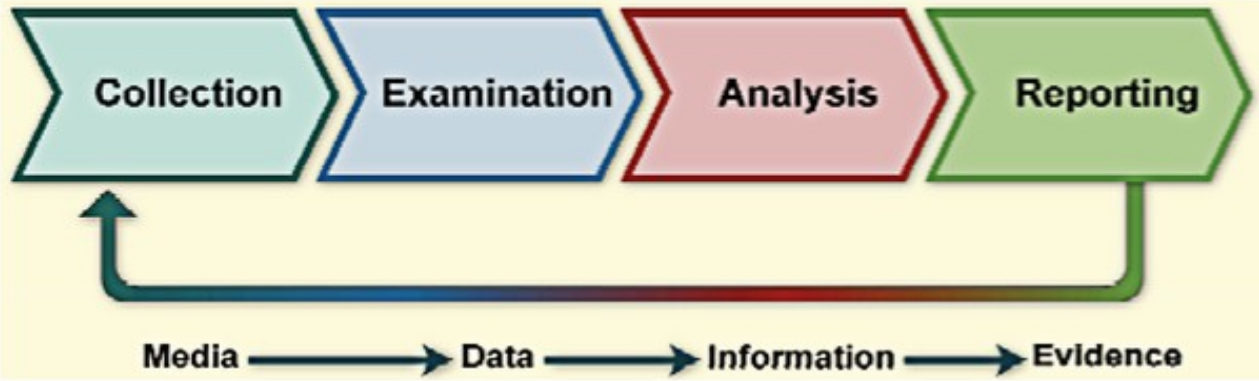
Per pastarąjį dešimtmetį stipriai išaugo nusikaltimų, atliekamų elektroninėje erdvėje, skaičius. To pasėkoje, įsikūrė daug kompanijų, kurios siūlo produktus, padedančius teisėsaugos institucijoms nustatyti kas, ką, kur, kada ir kaip padarė nusikalstamas veikas naudojantis kompiuterinėmis priemonėmis. Skaitmeninė teismo ekspertizė, kaip sfera, būtent ir išsivystė tuo pagrindu, kad būtų užtikrintas tinkamas elektroninių įkalčių pateikimas teismui. Dar 2001-ais metais pirmą karta buvo pasiūlyta galima metodologija, nusakanti kaip įkalčiai kibernetinėje erdvėje turėtų būti fiksuojami ir tiriami (žiūrėti pav. 2.1.).

2.1. pav. „Digital forensic research workshop 2001“ pasiūlytas tyrimo modelis[16]

Vėlesniais metais ši metodologinė schema buvo supaprastinta, ir dabar imta laikyti, kad skaitmeninės teismo ekspertizės tyrimas turėtų būti grindžiamas keturiomis pagrindinėmis fazėmis (žiūrėti pav. 2.2.)[13]:

- Surinkimas (angl. *collection*). Pirmiausia iš potencialių šaltinių turi būti nustatyti, užvadinti, surinkti ir išsaugoti visi nusikaltimo įkalčiai. Tai daroma vadovaujantis nustatytomis procedūromis ir rekomendacijomis, kurios išsaugo duomenų vientisumą. Paprastai labai svarbus yra laiko faktorius, nes pavėlavus galima prarasti dinaminius duomenis, tokius kaip esamus tinklo susijungimus ar įrašus iš įrangos, naudojančios baterijų energiją, pavyzdžiui, mobiliųjų telefonų, „išmaniųjų“ prietaisų ir kitų;
- Ištyrimas (angl. *examination*). Pasinaudojant automatizuotais ir rankiniais metodais, surinkti dideli duomenų kiekiai yra teisiškai apdorojami siekiant įvertinti ir išrinkti ypatingo reikšmingumo informaciją. Tai atliekant turi būti išlaikomas duomenų vientisumas;
- Analizė (angl. *analysis*). Šioje fazėje iš informacijos, surinktos ištyrimo fazėje, yra padaromos pagrįstos, nusikalstamus veiksmus apibūdinančios išvados, kurios pačios savaime ir yra analizės tikslas. Tai daroma vadovaujantis teisiškai pagrįstais metodais ir būdais;
- Atskaitomybė (angl. *reporting*). Galiausiai ataskaitose yra pateikiami analizės rezultatai, kurie papildomi tokia informacija kaip: kokie žingsniai buvo atlikti, kaip buvo pasirinkti įrankiai ir procedūros, kokie kiti veiksmai yra reikalingi (pavyzdžiui, papildomų duomenų šaltinių teismo ekspertizės atlikimas, nustatytų pažeidžiamumų apsaugojimas, esamų

saugumo valdymo įrankių patobulinimas). Be to, neretai pateikiamos rekomendacijos ir pačios teismo ekspertizės proceso patobulinimui: gairių, įrankių, politikos, procedūrų. Paprastai esama situacija apsprendžia, kokio formalumo ataskaitos yra reikalingos kiekvienu atveju atskirai[13].



2.2. pav. Kibernetinių nusikaltimų tyrimo eiga[13]

2.2.1. Skaitmeninė teismo ekspertizė debesies tipo serveriuose

Šioje darbo dalyje aptarsime, kuo skiriasi duomenų bazių tyrimas nuo tradicinės, pavyzdžiui, Windows failų sistemos tyrimo. Tradicinis Windows failų sistemos tyrimas savo dėmesį fokusuoja ties pastovios ir nepastovios informacijos rinkimu, kuri priklauso tam tikrai operacinei sistemai ir programai. Pavyzdžiui „Internet Exploler“ naršyklė arba „Microsoft Office“ programa. Didžiojoje dalyje atvejų šių sistemų tyrime duomenų bazė paliekama nuošalyje. Tokiais atvejais yra sudėtinga, o tam tikrai atvejais, praktiškai neįmanoma nustatyti, ar informacija esanti duomenų bazėje buvo sukompromituota atakos metu ar ne.

Tarkime, sistemos administratorius identifikuoja tam tikrus keistus įrašus WEB programos serverio log failuose. Po tam tikrų log failų tyrinėjimų, sistemos administratorius nustatė, kad buvo bandoma vykdyti SQL injekcijos ataką. Nustačius faktą, jog buvo vykdoma ataka, pradedamas šio fakto tyrimas. Tyrėjas ima nagrinėti operacinės sistemos failus, galimas tam tikras SQL injekcijos liekanas iš sistemos operatyvios atminties, kurios ten atsirado iš WEB programos serverio ir yra susietos su tam tikrais WEB programos serverio log įrašais. Vėliau tyrėjas nustato, kad buvo bandoma vykdyti SQL injekcijos ataka, kurios metu kenkėjišką SQL kodą buvo bandoma

„praleisti“ per Web puslapio interfaisą. Dėka WEB serverio log failų tyrėjas gali nustatyti iš kokio IP adreso buvo bandoma vykdyti šią ataką, tačiau tyrėjas negali pasakyti ar ši vykdoma ataka buvo sėkminga ar ne. Jei ji buvo sėkminga, ar informacija laikoma duomenų bazėje buvo sukompromituota ar ne. Tokiu atveju tyrėjas negali pasakyti kokio dydžio žala buvo padaryta[17].

Vykdant duomenų bazės serverio tyrimą, tyrėjas turėtų, be informacijos, kurią jis surinko iš WEB programos serverio, surinkti informaciją iš duomenų bazės serverio ir ją palyginti su įrašais WEB programos serverio log failuose. Tik vykdant Duomenų bazės serverio tyrimą galima pasakyti, ar SQL užklausa buvo sėkmingai priimta DB serverio, apdorota ir įvykdyta serveryje. Taip pat, tiriant DB serverį, galima pasakyti, ar kokia nors informacija laikoma DB serveryje buvo sukompromituota ar ne.

Kitas pavyzdys. Įsivaizduokime kitą scenarijų – tarkim, koks nors nesąžiningas įmonės darbuotojas padarė neautorizuotus pakeitimus 500GB duomenų bazėje. Tarkime, kad tais pakeitimas buvo keičiamos klientų mokėtinos sumos „Online“ pardavimų duomenų bazėje. Tradicinis tyrimas pradėdamas nuo to, kad yra padaroma operatyvios atminties kopija ir kietojo disko kopija. Yra nustatoma, kad darbuotojas interaktyviai buvo prisijungęs prie sistemos ne darbo metu, tačiau negalima nustatyti, ar vykdomos transakcijos/užklausos į duomenų bazės serverį iš tikrųjų pakeitė joje esančią informaciją ar ne[17].

Vykdant tikrą DB serverio tyrimą galima būtų nustatyti, kokia informacija buvo modifikuota prisijungimo metu, kokios transakcijos vyko ir kada vyko šios transakcijos. Turint omenyje, kad duomenų bazė yra gan didelė, galima būtų sumažinti paieškos kriterijus, pavyzdžiui, ieškoti tik tam tikrų DML užklausų, kurios buvo skirtos tam tikriems klientų įrašams arba tam tikroms duomenų bazių lentelėms. Taip pat vykdant duomenų bazės serverio tyrimą galima būtų nustatyto, kokios buvo pirminės klientų įrašų reikšmės ir į kokias jos buvo pakeistos.

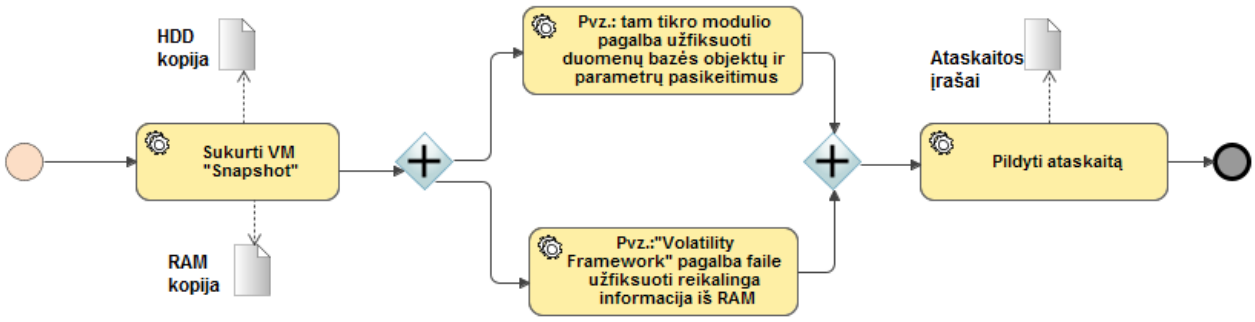
2.2.2. Kodėl debesies duomenų bazėms reikia kitokios tyrimo metodologijos?

Įsivaizduokite, jūs esate didelės kompanijos savininkas, kuri turi galbūt 10 tūks. klientų, o gal net ir 10 mln. klientų. Akivaizdu, kad tokio dydžio įmonei reikia labai didelio IT ūkio, t. y. keliolikos serverių, duomenų saugyklų, sistemos atstatymo planų po netikėto gedimo, tokių planų testavimo, naujų IT produktų kūrimo ir ,aišku, IT personalo, kuris vykdytų šias ir kitas veiklas, susietas su įmonės darbo palaikymu. Dažnai didelės įmonės renkasi debesies kompiuterinius

sprendimus, dėl priežasčių, kurios jau buvo išvardintos šiame darbe - mažesnių valdymo kaštų, nes nebereikia rūpintis serverių priežiūra, keisti sugedusius komponentus, vykdyti sistemos atnaujinimus. Tačiau iš šios sistemos yra reikalaujama, kad ji būtų visada pasiekiama, jos darbas būtų nepertraukiamas. Pavyzdžiui, įmonėms, kurios vykdo elektroninius atsiskaitymus, yra ypač aktualu, kad sistemos veikla niekad nesutriktų, nes kitokių atveju nebūtų galimybės vykdyti atsiskaitymus, o tai reikštų klientų pasitikėjimo praradimą bei pelno mažėjimą.

Atsižvelgiant į argumentus, kurie buvo paminėti aukščiau, galima apibūdinti silpnąsias tradicinio nusikalstamos veikos tyrimo būdo vietas (žiūrėti pav. 2.3.). Pirma, vykdant tokį tyrimą, pagal nusistovėjusią praktiką, konfiskuojamas kompiuteris, kuriuo buvo vykdoma nusiklastoma veika. Serveris, kuriuo naudojasi įmonė, nebūtinai yra tame pačiame pastate, mieste ar net šalyje, todėl fizikai įgyvendinti šio dalyko neįmanoma. Jei neegzistuoja galimybės konfiskuoti debesies serverio kompiuterinę įrangą, tai pat nėra galimybės tiesiogiai pasidaryti kompiuterio kietojo disko ir operatyvios atminties kopijas. Tarkime, operatyvios atminties kopiją ir kietojo disko kopiją galima pasidaryt naudojant sertifikuotas programas nuotoliniu būdu, tačiau šioje vietoje taip pat atsiranda eilė problemų. Pirma, kokio dydžio yra kompiuterio, o kalbant apie debesies technologiją, debesies serverio operatyvioji atmintis? Taip pat kokio dydžio yra dedikuota duomenų saugykla? Ar duomenis įrašomi tik į vienos saugyklos diskus ar į kelių saugyklų diskus? Kiek laiko užtruks padaryt šių komponentų kopijas? Bet kokiu atveju, ar būtų kopijuojama operatyvioji atmintis, ar duomenys esantys diskuose, būtų susidurta su sistemos pasiekiamumo trukdžiais. Sistemom, kurios turi būti pasiekiamos 24/7 tokie trukdžiai yra nedovanotini, nes tai reiškia pajamų ir klientų praradimą[18].

Iš išvardintų specifinių dalykų, kurie būdingi surenkant nusikalstamos veikos įkalčius duomenų bazėse, ir debesų kompiuterijos ypatybių, galima teigti, kad tai reikalauja netradicinės metodologijos. Šiame darbe yra siūloma nauja metodologija, kuri apsiribotų įkalčių rinkimų debesies duomenų bazėse naudojant šių duomenų bazių serverių sukuriamus sisteminius įrašus. Kiekvienas DBVS turi jai skirtas programas, kurios gali padaryti momentines kopijas veiksmų, vykstančių arba jau vykusių serveryje. Naudojant minėtus įrašus galima rekonstruoti vykdytas SQL užklausas, nustatyti nusiklastomos veikos vykdytoją bei vėliau palyginti duomenų bazės serveryje surinktą informaciją su, pavyzdžiui, programų serverio įrašais.

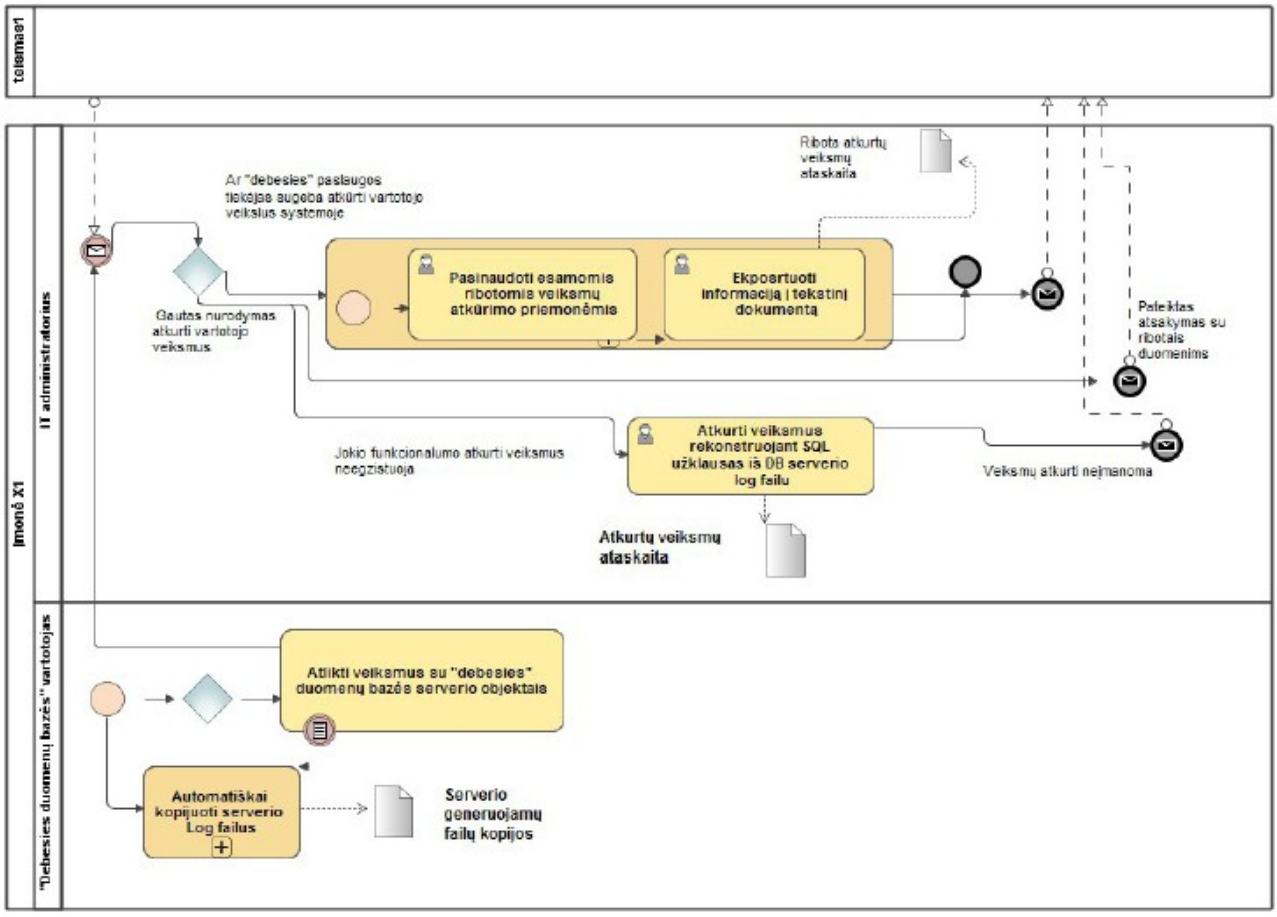


2.3. pav. Tradicinė nusikalstamos veikos kompiuterizuotoje aplinkoje tyrimo eiga (BPNM proceso diagrama)

2.2.3. Nusikaltimų pėdsakų fiksavimas debesies duomenų bazių serveriuose

Prieš tai buvusiame darbo skyriuje buvo išanalizuotas ir aiškiai nustatytas tokios paslaugos poreikis šių dienų teismo procesuose, kuriuose susiduriama su debesies duomenų bazių serveriuose galimai esamais įkalčiais. Problema iškyla todėl, kad nėra vieningų reikalavimų ar standartų, apibrėžiančių asmenų, besinaudojančių debesies serveriu, veiksmų fiksavimo būtinumą ir konkrečias šio funkcionalumo įgyvendinimo gaires. Vadinasi, įvykus nusikaltimui ir prireikus tokius veiksmus atkurti, teismas turi pasikliauti tokios paslaugos tiekėjų suteiktu vidiniu tam tikslui skirtu funkcionalumu, kuris ne tik kad gali būti nepakankamas arba dažniausiai neegzistuoja išvis.

Siūlomas metodas nagrinėjamos problemos kontekste pavaizduotas apačioje (žiūrėti pav. 2.4.).

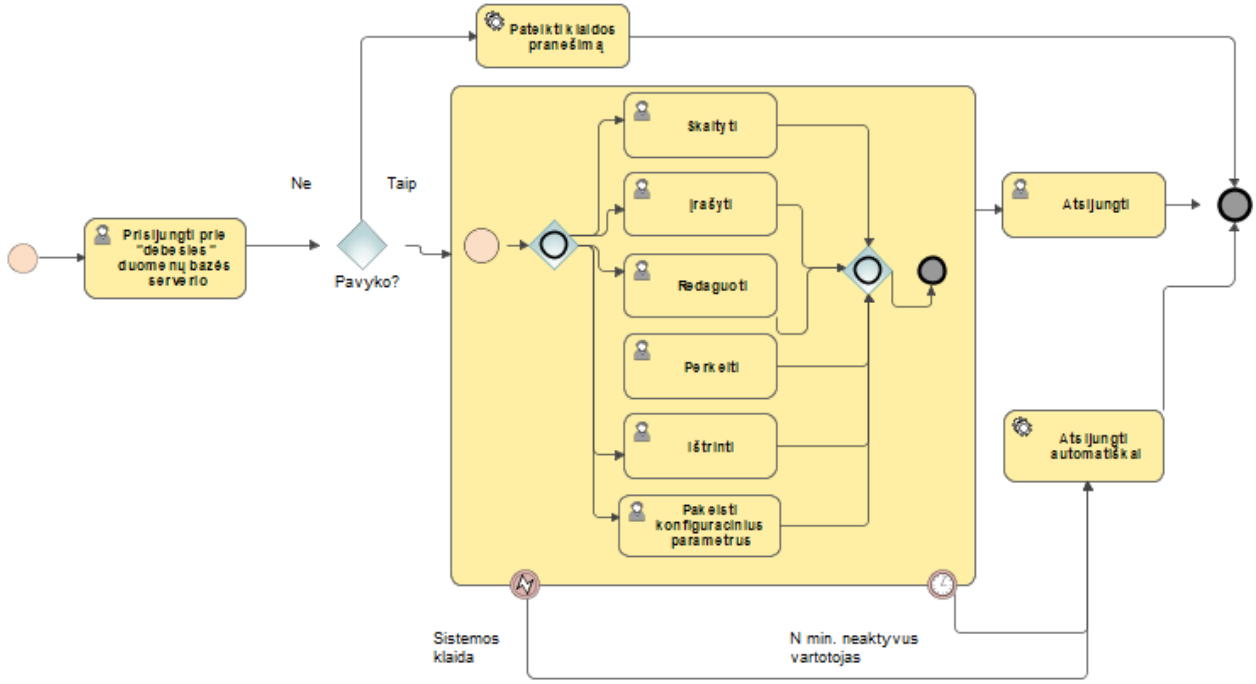


2.4. pav. Nusikalstamos veikos tyrimo procesas kompiuterizuotoje aplinkoje

Taigi šiame skyriuje yra pateikiamas metodas, kaip galima būtų atkurti nusikalstamos veikos pėdsakus, naudojant duomenų bazės serverio užfiksuotą informaciją, kai neegzistuoja jokio papildomo funkcionalumo. Šioje vietoje reikia patikslinti, kad šis metodas suteikia galimybę tokią informaciją surinkti iš duomenų bazės serverio, vėliau tokia informacija gali būti palyginama su Web serverio ar programų serverio informacija, apie kurią šiame darbe nėra kalbama.

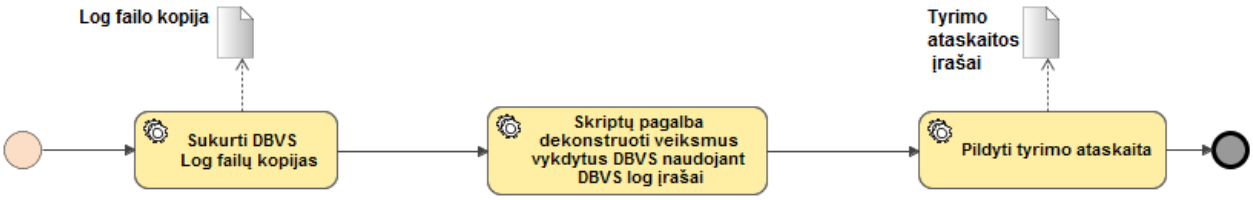
Taigi, principinė šio metodo eiga būtų – pirma, reiktų pasidaryti duomenų bazės serverio sisteminių (log) failų kopijas. Tas kopijas galima pasidaryti naudojant duomenų bazės serveryje esančias paprogrames. Kiekvienas tokio tipo serveris turi tam tikras dedikuotas programas, kurios skirtos duomenų bazės serverio infrastruktūrai palaikyti, procesams analizuoti ir t.t. Tų programų dėka, kartu su duomenų bazės serverio sisteminiais įrašais galima pasidaryti ir momentines visos šios sistemos kopijas, t. y. kokie procesai esamu metu vyksta serverio operatyvioje aplinkoje[19]. Dažnai šios programos yra naudojamos tam, kad serverio programinės įrangos tiekėjui butų pateikta informacija apie galimas programinės įrangos kodo klaidas, už kurių taisymą atsako tos programos

gamintojas. Tokia sistemos kopija, skirta programinės įrangos tiekėjui, yra pateikiama dvejetainiu formatu, nes sistemos gamintojui taip pat yra aktualu, kad tokia informacija būtų nuosekli ir jos kopijavimo metu ji nebūtų pakeista, nes tai gali daryti įtaką atsiradusių klaidų taisymui[20]. Kokie vartotojo veiksmai yra fiksuojami duomenų bazės serverio yra grafiškai atvaizduota BPMN proceso diagramoje (žiūrėti pav. 2.5.).



2.5. pav. Duomenų bazės serverio fiksuojami veiksmai (BPNM proceso diagrama)

Dažnais atvejais serverio sugeneruotų failų pakanka tam, kad būtų galima nustatyti nusikalstamos veikos pobūdi ir galimai padarytą žalą. Dažnai skirtinguose duomenų bazės serveriuose duomenys būna įrašomi į skirtingo tipo sisteminius failus (log failai). Tada kiekvieno tokio failo kopija turi būti daroma atskirai ir kartais naudojant skirtingas serverio paprogrames[21].



2.6. pav. Įkalčių surinkimas debesies tipo serveryje (BPNM proceso diagrama)

Siūlomu metodo-būdo pagalba siekiama parodyti, kaip galima rekonstruoti debesies duomenų bazės serveryje vykusius veiksmus iš naudojamu sisteminių failų, turint omenyje, kad minėtame serveryje neegzistuoja įsiskverbimo tikrinimo variklio arba kito sisteminio sprendimo, kuris galėtų papildomai fiksuoti įvykius duomenų bazės serveryje (žiūrėti pav. 12). Šiam tikslui pasiekti šiame darbe analizuojama kelių autorių atlikta studija, kuri buvo skirta MySQL serverio log failų analizavimui ir SQL užklausų rekonstravimui. Taip pat bus nagrinėjama šio serverio struktūra, failų tipai, pagalbinės programos. Visa tai yra reikalinga tam, kad būtų sudarytas bendras vaizdas apie šio duomenų bazės serverio veikimo principus, fiksuojamus įvykius ir kaip visą šia informaciją galima apdoroti. Tad turint kiek detalesnį supratimą apie MySQL galima bus lengviau perprasti kitų autorių aprašytą praktiką, t. y. kaip naudojantis esamais informaciniais ištekliais galima rekonstruoti DBVS vykdytas užklausas. Įgytos teorinės žinios bus taikomas analizuojant DB2 serverio veikimo principus ir kaip šio tipo serveryje galima rekonstruoti vykdytas užklausas.

2.2.4. Duomenų bazių serverių architektūra

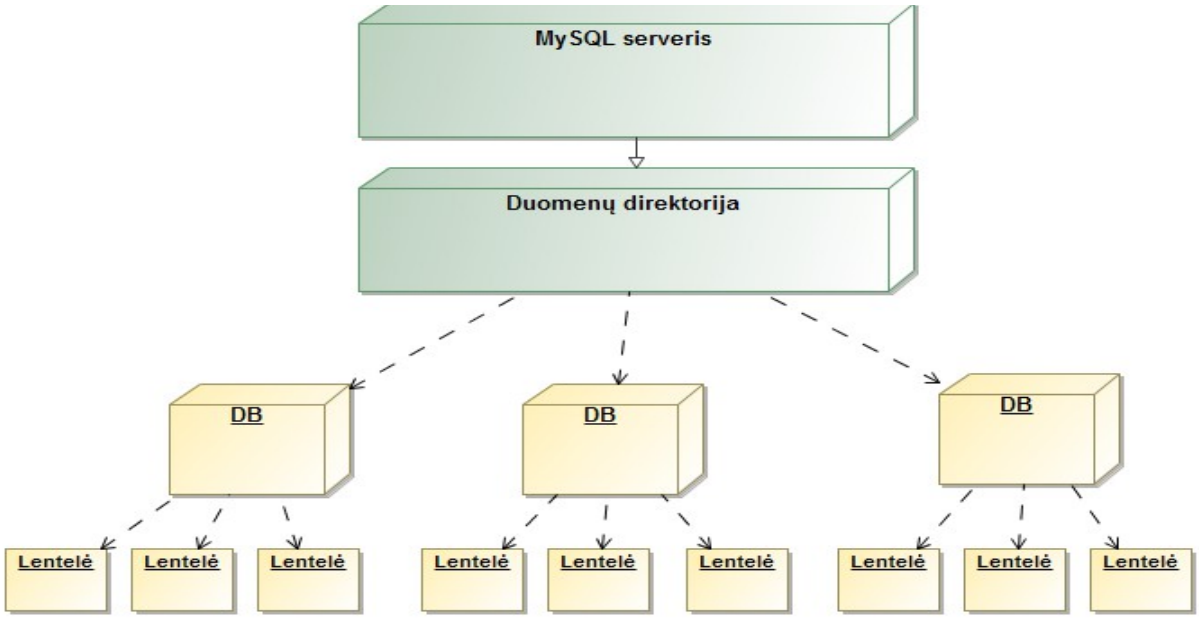
2.2.4.1. MySQL duomenų bazės serverio struktūra

Šioje dalyje apžvelgsime esamus šios serverio komponentus, kurie turi būti žinomi tyrėjui atliekant nusikalstamos veikos įkalčių rinkimą.

Pagrindinis MySQL serverio saugyklos valdymo variklis - InnoDB. Jis saugo visas serveryje esančias duomenų bazes, objektų statusus, failus ir sistemos log failus. Taip pat tam tikra informacija yra laikoma globaliame (angl. *Global*) ir Saugyklos (angl. *Storage*) varikliui specifinėje operatyvios atminties vietoje. Šie komponentai sudaro visą pagrindą atliekant nusikalstamos veikos tyrimus.

2.2.4.2. Duomenų direktorijos struktūra

Duomenų direktorijoje MySQL serveris saugo duomenų bazes ir sistemos statusų failus. Kiekviena serveryje esanti duomenų bazė turi savo įrašą duomenų direktorijoje. Kiekvienas su duomenų baze susietas objektas turi įrašą apie save duomenų bazės direktorijos faile[22] (žiūrėti pav. 2.7.).



2.7. pav. MySQL duomenų direktorijos pavyzdys

Duomenų direktorijoje taip pat saugomi ir kiti failai:

- Serverio procesų ID (PID) failas. Kai serveris yra paleidžiamas jis įrašo visus procesų ID į minėtąjį failą. Kitos programos vadovaujantis šio failo informacija gali rasti sekančio proceso reikšmę;
- Serverio generuojami log failai. Šie failai yra ypatingos svarbos, nes gali leisti atsekti informaciją apie neleistiną veiką arba sistemos veiklos problemas.

2.2.4.3. MySQL sisteminis katalogas, statuso ir log failai

Sisteminiu katalogu MySQL duomenų bazių serveryje vadinama Information_schema duomenų bazė. Joje laikomi metaduomenys apie visus serveryje esančius objektus. Kiekvienas serverio vartotojas gali prieiti prie informacijos šioje duomenų bazėje, tačiau vartotojams bus matoma tik ta informacija, prie kurios jie turi atitinkamas prieigos teises.

Sistemos log failai generuoja gana didelį informacijos kiekį, kuris įrašomas į informacijos laikmenas. Įrašomos informacijos kiekis priklauso nuo serverio apkrovimo. Už failų saugojimą atsakingas duomenų saugyklos variklis, tačiau papildomai jis gali generuoti ir loginius log failų įrašus (tai yra papildoma kiekvieno duomenų saugyklos variklio funkcija). Žemiau pateiktoje lentelėje yra pateiktas serverio lygio statuso ir log failų sąrašas. Atskiri duomenų saugyklos varikliai gali turėti savo atskirus log failus[23].

2 lentelė: MySQL serverio failų tipai

Failo tipas	**Vardas pagal nutylėjimą**	**Failuose saugoma informacija, kuri gali būti naudojama veiksmų atkūrimui**
Process ID failas	HOSTNAME.pid	Serverio proceso ID
Error log	HOSTNAME.err	Procesų pradžios ir pabaigos ir jų klaidos
General query log	HOSTNAME.log	Prisijungimų/atsijungimų prie serverio informacija
Binary log	HOSTNAME-bin.nnnnnn	SQL užklausų atvaizdavimas dvejetainės kodu
Binary log index	HOSTNAME-bin.index	Dvejetainės log failų sąrašas
Relay log	HOSTNAME-relay-bin.nnnnnn	Bet kokios informacijos modifikacijos užklausa gauta iš kito serverio
Relay log index	HOSTNAME-relay-bin.index	Iš kito serverio gautų modifikacijos užklauso log failų sąrašas
Master info file	Master.info	Parametrų sąrašas skirtu prisijungimui prie pagrindinio serverio
Relay info file	Relay-log.info	Saugo informacija apie Relay log procesinimą
Slow-query log	HOSTNAME-slow.log	Neefektyvių SQL užklausų log failas

InnoDB duomenų saugyklos variklis turi dvejų tipų log failus – „undo log“ ir „redo log“. Vienas jų skirtas gražinti informacija į prieš tai buvusį COMMIT tašką, kitas skirtas atkurti paskutinius darytus pakeitimus, kai įvyko sistemos nesankcionuotas atjungimas.

log failai – svarbiausias informacijos šaltinis, norint sužinoti kokie veiksmai buvo vykdomi duomenų bazės serveryje. Pavyzdžiui, „General query log“ faile galima atrasti tokia informaciją (žiūrėti pav. 2.8.). Pagal nutylėjimą ši informacija įrašoma į duomenų direktoriją.

```
1  080412 16:47:24 44 Query SET PASSWORD FOR
2  'root'@'localhost'=PASSWORD('secret')
```

2.8. pav. „General log“ failo įrašo pavyzdys

2.2.4.4. MySQL pagalbinės programos

MySQL serveris turi eilę programų, kurios gali būti naudojamos atkuriant ir tiriant nusikalstamas veikas serveryje.

Mysąldump – duomenų bazės atsarginės kopijos generavimo programa. Ši serverio programa yra naudojama generuojant duomenų bazės struktūrą ir joje laikoma informaciją, kuri vėliau gali būti perkeliama į kitą serverį (žiūrėti pav. 2.9.). Taip pat ši programa gali sugeneruoti tokį dokumentą XML formatu.

```
shell> mysqldump --xml -u root TESTING1 City
<?xml version="1.0"?>
<mysqldump xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<database name="TESTING1">
<table_structure name="City">
<field Field="ID" Type="int(11)" Null="NO" Key="PRI" Extra="auto_increment" />
<field Field="Name" Type="char(35)" Null="NO" Key="" Default="" Extra="" />
<field Field="CountryCode" Type="char(3)" Null="NO" Key="" Default="" Extra=""
/>
<key Table="City" Non_unique="0" Key_name="PRIMARY" Seq_in_index="1"
Column_name="ID"
Collation="A" Cardinality="4079" Null="" Index_type="BTREE" Comment="" />
<options Name="City" Engine="MyISAM" Version="10" Row_format="Fixed"
Rows="4079"
Avg_row_length="67" Data_length="273293"
Max_data_length="18858823439613951"
Index_length="43008" Data_free="0" Auto_increment="4080"
Create_time="2007-03-31 01:47:01" Update_time="2007-03-31 01:47:02"
Collation="latin1_swedish_ci" Create_options="" Comment="" />
</table_structure>
<table_data name="City">
<row>
<field name="ID">1</field>
<field name="Name">Kabul</field>
<field name="CountryCode">AFG</field>
</row>
...
<row>
</table_data>
</database>
</mysqldump>
```

2.9. pav. MySQLdump programos sukurto failo pavyzdys

Mysqlaccess – serverio klientinė programa tikrinanti prieigos teises. Šios programos pagalba galima gauti informaciją kokias prieigos teises turi tam tikri sistemos vartotojai ir/arba iš išorės prisijungę vartotojai.

Myisamlog – programa, išvedanti MyISAM log failo turinį. Šios programos dėka galima atkurti objekto statusą į tam tikrą vietą laike, vykdyti UPDATE operacijas bei objektų versijavimą.

Myisamchk – programa skirta sistemos palaikymui. Jos pagalba gali reorganizuoti serverio objektus taip juos optimizuojant.

Mysqlbinlog – programa apdorojanti dvejetainio formato log failus (žiūrėti pav. 2.10.). MySQL serveris informacija apie vykdytus veiksmus sistemoje įrašoma dvejetainės formatu, tam, kad šią informacija galima būtų pateikti tekstiniu formatu yra naudojama mysqlbinlog programa.

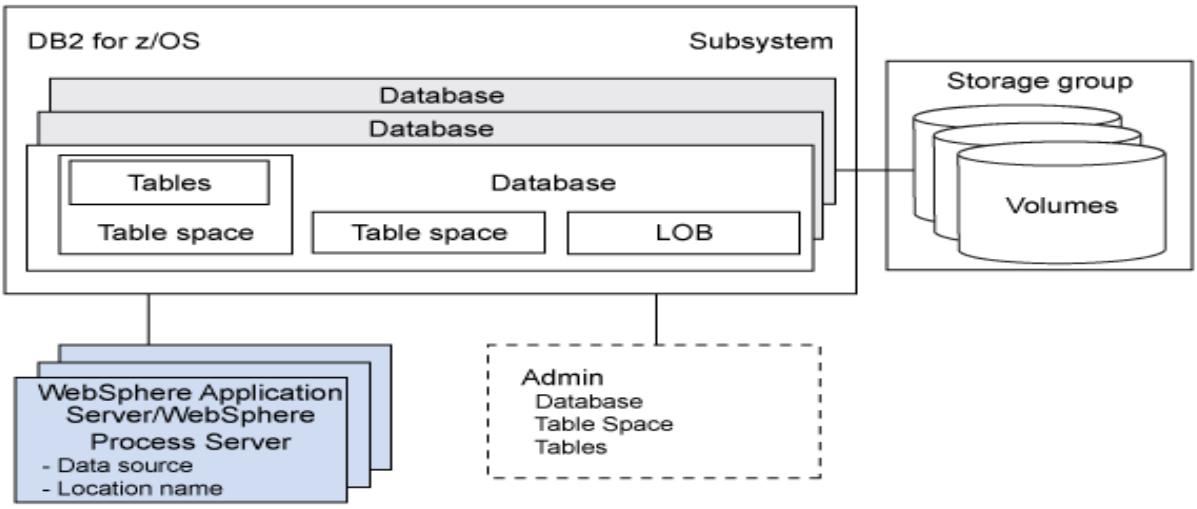
Taip pat ši programa gali būti naudojama sukuriant šešioliktainės formato dump failo sukūrimą[24].

```
# at 4
#051024 17:24:13 server id 1 end_log_pos 98
# Position Timestamp Type Master ID Size Master Pos Flags
# 00000004 9d fc 5c 43 0f 01 00 00 00 5e 00 00 00 62 00 00 00 00 00
# 00000017 04 00 35 2e 30 2e 31 35 2d 64 65 62 75 67 2d 6c |..5.0.15.debug.l|
# 00000027 6f 67 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |og..............|
# 00000037 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
# 00000047 00 00 00 00 9d fc 5c 43 13 38 0d 00 08 00 12 00 |.......C.8......|
# 00000057 04 04 04 04 12 00 00 4b 00 04 1a |.......K...|
# Start: binlog v 4, server v 5.0.15-debug-log created 051024 17:24:13
# at startup
ROLLBACK;
```

2.10. pav. MySQLbinlog progrmos sukurto failo pavyzdys

2.2.4.5. DB2 duomenų bazės struktūra

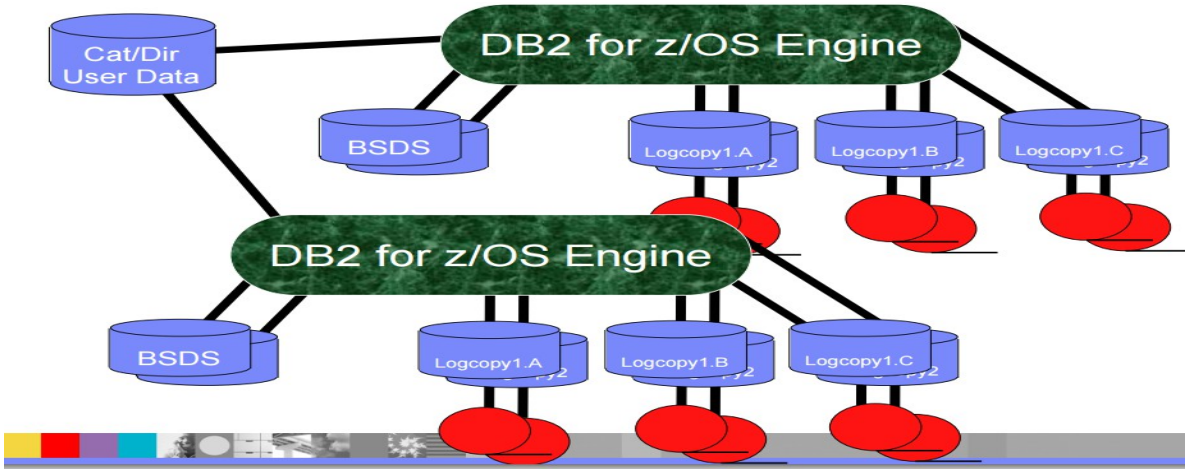
DB2 yra IBM reliacinių duomenų bazių valdymo sistema skirta naudoti su įvairia aparatine įranga. Pagal pajėgumus ir funkcionalumą galima įsigyti įvairių DB2 versijų: nuo nemokamų Express-C iki didelėms įmonėms pritaikytų duomenų bazių valdymo sistemų. Komercinės DB2 versijos turi daug saugos ir aukšto patikimumo savybių. Šiame darbe bus naudojama DB2 z/OS versija, skirta Mainfraime architektūros serveriams. Kaip ir visi duomenų bazės serveriai DB2 duomenų bazės serveris visus jame vykstančius procesus įrašo į log failus (analogiškai MySQL), kurie gali būti naudojama grąžinant duomenis į ankstesnę būseną įvykus sistemos gedimui arba kitiems nenumatytiems įvykiams [25].



2.11. pav. DB2 duomenų bazės architektūra[25]

DB2 žargonu kalbant, šie log failai yra skirstomi į aktyvius ir archyvuotus. Aktyvus log failai yra tie failai, į kuriuos esamu momentu vyksta įrašymas ir jie bus naudojami duomenų atstatymui, pavyzdžiui, poros minučių tarpe. Įsivaizduokime tokią situaciją. Programa Prisijungė prie duomenų bazės BASE1 ir bandė atnaujinti 10 dienų senumo įrašus lentelėse. Vykdant DML užklausą nebuvo vykdoma jokia COMMIT operacija ir po poros minučių veikimo programa buvo „jėga“ sustabdyta, tokiu atvejų visi daryti pakeitimai turi būti gražinami į ankstesnę būklę, tokiu atveju ši informacija duomenų bazės serverio bus atstatoma iš aktyvių log įrašų, kitaip vadinamų „Active logs“ (žiūrėt pav. 2.12.) .

Kitas pavyzdys būtų, tarkime, prieš savaitę duomenų bazėje buvo daryti atnaujinimai, ir visiems klientas, gimusiems šį mėnesį, per klaidą buvo paskaičiuoti papildomi mokesčiai. Tokiai informacijai atstatyti iki pakeitimo bus naudojamas suarchyvuotas DB2 serverio log failas, kitaip vaidinamas Archive log. Visa informacija apie esamus log failus, jų buvimo vietą bei laiką, kada jie buvo sukurti, kokio laikotarpio informacija yra saugoma ir įrašoma į taip vadinamus BSDS failus. DB2 serverio įrašų kūrimo architektūrą galima butų pavaizduoti šiuo paveikslu[26].



2.12. pav. DB2 duomenų bazės architektūra su sistemos log failais[26]

Į log failus įrašytą informaciją galima išgauti naudojant DB2 serverio pagalbinę programą – DSN1LOGP. Šios programos dėka sistema sugeneruoja tokią sistemos kopiją:

```
00006BFBE999  LRSN(C6CD403EB3AF)  TYPE(SYSTEM EVENT)
              SUBTYPE(TRACE RECORD)

*LRH* 01400034 00100041 10800000 00000000 00000000 00000726 00000000 00000000 0000C6CD  *                          F
      403EB3AF 0000                                                                       *
0000  011A0000 00000028 00F20001 00000014 00130001 000B60E2 E3D6D740 E3D6D740 C4C2F216  *        2           -STOP DB2
0020  81AB2000 00000040 00560117 005A02A1 16180930 C4E2D5C1 C6CD403E C6CD403E B392DDEE  *a          !      DSNAF    k
0040  00000006 00000006 00000000 E2E3D3C5 C3F14040 40404040 40404040 40404040 C4E2D5C1  *           STLEC1          DSNA
0060  40404040 E2E8C5C3 F1C4C2F2 C6CD403E B3770001 00000000 0000F3F0 0000F3F0 F9F0009C  *   SYEC1DB2F            3090
0080  0200E2E8 E2D6D7D9 4040F0F2 F34BC7C3 E2C3D5F6 F0F2E5C1 F1C14040 F1C14040 40404040  * SYSOPR  023.GCSCN602VA1A
00A0  40404040 4040E2E8 E2D6D7D9 40400000 00000000 00000000 00000000 00000000 00000000  *     SYSOPR
00C0  00000000 00000000 00004040 40404040 40404040 40404040 40404040 40404040 40404040  *
00E0  40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040  *
0100  40404040 40404040 40404040 00000000 00000000 00000000 0000
```

2.13. pav. DB2 log failo informacija surinkta naudojant DSN1LOGP programą

Be DSN1LOGP paprogrames egzistuoja DSN1DMP pagalbinė programa, kurios pagalba galima pasidaryt kopiją visų procesų, vykstančių duomenų bazės serveryje (žiūrėt pav. 19). Ši programa sukuria SMF failus, kuriuos naudojant tam tikrus skriptus įmanoma apdoroti, taip išgaunant informaciją apie potencialius įsilaužimus.

Kiekviename DB2 duomenų bazės serveryje automatiškai, vos tik suinstaliavus patį DBVS serverį, paleidžiamos įvykių klausymosi paprogramės (angl. *trace*). Jų skaičius ir tipas priklauso nuo instaliacinių parametrų. DB2 klausymo paprogramės skirstosi į kelis tipus[27]:

- *Apskaitos paprogramė (angl .Accounting trace)* – pateikia informaciją apie transakcijų lygio įrašus, kuri yra įrašoma, kai baigiama transakcija. Šią informaciją naudojame serverio apkrovos planavimui ir programų efektyvumui gerinti;
- *Nauduojamumo paprogramė (angl. Performance trace)* – skirta DBVS veikimo analizei ir optimizavimui. Jos dėka galima surinkti informaciją apie tam tikrus įvykius sistemoje, pavyzdžiui: prisijungimo informacija iš nutolusių serverių; vykdytų užklausų kiekį ir jų tipą. Šią informaciją galima panaudoti optimizuojant programos, sistemos resursų, vartotojo ar duomenų bazės serverio optimizavimui;
- *Monitprinimo popragramė (angl. Monitoring trace)* - atlieka monitoringą vykdomų SQL užklausų ir kitų sisteminių pasikeitimų.
- *Audito paprogramė (angl. Audit trace)* – surenka informaciją apie serverio saugos „variklio“ darbą ir jo sugeneruotus pranešimus;
- *Statistinė paprogramė (angl. Statistic trace)* – surenka statistinę informaciją apie tai, kiek yra naudojami duomenų bazės serverio sisteminiai resursai.

2.3. Išvados

Išanalizavus nusikalstamos veikos kompiuterizuotoje aplinkoje tyrimo būdus ir skirtingus duomenų bazės serverius, galima daryti šias išvadas:

- Tradicinės nusikalstamos veikos tyrimo metodologija, kurios metu yra daromos operatyvios atminties ir kietojo disko kopijos, nėra tinkama atlikti tyrimą debesies tipo serveriuose;
- Dėl savo ypatumų debesies duomenų bazių serveriai turi turėti naują metodą, kuris leistų ištirti atliktus veiksmus serveryje;
- Šiuo metu naudojami duomenų bazių serveriai gali generuoti ataskaitas apie vykdytus serveryje veiksmus, dažniausiai šių atskaitų pakanka, kad būtų galima rekonstruoti SQL užklausas iš serverio sisteminių failų;
- Reikalui esant, duomenų bazių serverio pagalbinėmis programomis galima

sugeneruoti serverio operatyvios atminties kopiją, kartu su sisteminių failų momentine kopija. Tokiais būdais sugeneruoti failai gali būti analizuojami ir naudojami kaip įkalčiai teisme.

Kitoje šio darbo dalyje bus teoriškai paaiškinamas naujo metodo veikimas, kuris praktiškai yra įgyvendintas eksperimentiniame tyrime. Šio metodo dėka galima atkurti veiksmus, vykdytus DBVS, nesant specialiai tokių įvykių fiksavimo ir atkūrimo sistemai.

3. Artefaktų surinkimas ir analizavimas

Šioje darbo dalyje praktiškai apžvelgsime kokiais būdais galima rekonstruoti serveryje vykdytas užklausas, kokia informacija tam turi būti surinkta ir kokia yra tokio proceso eiga.

MySQL severio atveju, pirmiausia, prieš pradedant nusikaltimo pėdsakų fiksavimą ir atkūrimą debesies duomenų bazių serveryje, reikia surinkti serveryje saugotus artefaktus – log failus, dvejetainius failus, tekstinius failus (.MYD, .MYI, .FRM), naudojantis MySQL serveryje įdiegtomis programomis. Vėliau šie duomenys yra apdorojami tam skirtais skriptais taip suformuojant metaduomenis. Analizuojant metaduomenis bandoma rasti ir atkurti nusikalstamos veikos pėdsakus. Galiausiai yra generuojama šios veiklos ataskaita.

DB2 duomenų bazės serverio atveju principinė proceso diagrama praktiškai nesikeičia. Vienintelis dalykas, kuris šiek tiek skiriasi – log failų įvairovė. Visą reikiamą informaciją apie vykdytus procesus serveryje galima atkurti iš „Active log“ failų bei „Archive log“ failų bei SMF failų, kurie generuojami TRACE paprogramėmis.

3.1. MySQL ir DB2 log failų formatai

Kaip iliustruojančiu pavyzdžiu, šiame darbe remiamasi keliomis studijoms, kurios buvo atliktos Peter Fruhwirt, Peter Kieseberg[28]. Jų tyrimas buvo atliktas analizuojant MySQL serverio log failus ir jų dėka rekonstruojant vykdytas užklausas. Taigi šioje darbo dalyje apžvelgsime autorių išnagrinėta MySQL failų struktūrą ir būdą, kaip iš tokių failų yra rekonstruojamos užklausos, vėliau šios įgautos teorinės žinios leis mums atitinkamu būdu išanalizuoti DB2 duomenų bazės serverio failų struktūra bei rekonstruoti jame buvusias užklausas. Savo log failų struktūra abi DBVS yra gan panašios. Šio darbo tyrime bus naudojamos MySQL .FRM log failai ir DB2 Active log failai. Šių dviejų log failų struktūrą galima suskirstyti į kelias dalis:

- failo santraukos informacija (būdinga tik DB2 paprogramės sugeneruotam failui);
- log failo antraštė;
- log failo subantraštė (MySQL vadinama atskaitos tašku);
- įrašyta informacija apie vykdytus veiksmus.

MySQL duomenų bazė turi kelis įrašymo variklius (angl.*storage engine),* todėl papildomos informacijos nuo 0x0000 poslinkio kiekis gali būti kintamas.

Tam, kad būtų galima sėkmingai rekonstruoti SQL užklausas, autoriai naudoja kelių tipų failus:ib_logfile0 , ib_logfile1 ir .FRM tipo failus, kurie nusako kokia yra lentelės struktūra. Šiuos visus minėtus failus galima išgauti MySQL serveryje, naudojant anksčiau šiame darbe minėtas serverio paprogramės. b_logfile0 , ib_logfile1, kurios saugo savyje informaciją, kuri reikalinga serveriui atstatant darbą, t. y. duomenų vientisumą, pavyzdžiui, po neplanuoto sistemos veikimo sutrikdymo.

Informacija apie log failo antraštėje saugoma informaciją galima išskaidyti į kelias dalis (žiūrėti lentelę 3)[26].

3 lentelė: Log failo atneštinės baitų informacija

Poslinkis	Ilgis	Interpretacija
0x00	4	Log antraštes numeris
0x02	2	Įrašytų baitų kiekis bloke
0x04	2	Poslinkis nuo pirmų log grupės įrašų
0x05	4	Atskaitos taškų kiekis
0x06	2	Antraštės dydis

Pats log failas taip pat turi savo loginę struktūra, kurios analizės dėka galima išgauti daug vertingos informacijos (žiūrėti lentelę 4,5)[26].

4 lentelė: MySQL log failo pagrindinės dalies baitų informacija[1]

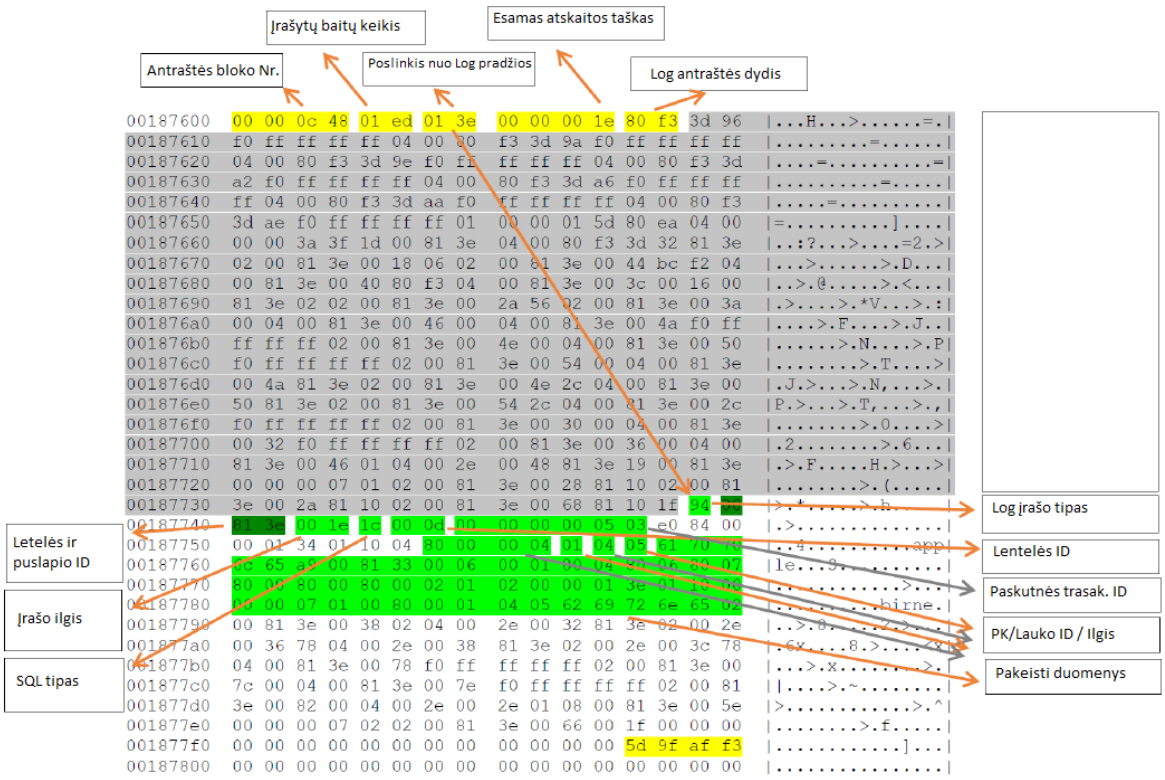
Pozicijos numeris	Ilgis	Interpretacija
1	1	Log įrašo tipas (fiksuota reikšmė: 0x94)
2	o	Lentelės ID (ang. Tablespace)
3	o	Puslapio ID
4	2	Log įraš0 ilgos
5	1	SQL tipas
6	2	Lentelės ID (ang. table)
7	6	Paskutinės transakcijas ID šiame lauke
8	o	Paskutinis grįžimo taškas
9	1	PK rakto ilgis
10	9*	Paveikto PK raktas
11	1	Pakeistų laukų kiekis
12	1	Pirmo pakeisto lauko vieta
13	1	Pirmo pakeisto lauko ilgis
14	13*	Perrašytų duomenų reikšmė

5 lentelė: MySQL SQL užklausų reprezentavimas log failuose

SQL užklausos tipas	Interpretacija
0x0B	INSERT
0x1C	UPDATE
0x0E	DELETE

Autoriai savo darbe, pasinaudojant Mysqlbinlog MySQL serverio pagalbine programa, išgavo serverio log failą, kurį vėliau rekonstravo naudojant iš lentelės pateiktą baitų informaciją. Kaip vizualiai atrodo toks procesas, pavaizduota paveiksle apačioje[29] (žiūrėti pav. 3.1.).

1 Simbolis o reiškia kad tai kintamas dydis, kuris priklauso nuo vykdomos operacijos tipo. Simbolis * reiškia, kad dydis gali kisti, jei duomenys lentelėje yra suspausto formato



3.1. pav. MySQL log failo baitų informacijos reikšmės

Visa detali informacija apie baitų informaciją, saugomą log faile, yra išdėstyta minėtų autorių tyrime. Šio darbo tikslas – pritaikyti minėtąjį teorinį modelį analizuojant DB2 duomenų bazės serverio log failus ir rekonstruoti šiame serveryje vykdytas užklausas.

3.2. DB2 duomenų bazės serverio log failų struktūra

Kaip buvo minėta anksčiau šiame darbe, DB2 duomenų bazės serverio log failas susideda iš subantraštės, kurioje galima rasti informaciją apie šioje failo dalyje saugomą log informaciją ir jos tipą (žiūrėti pav. 3.2.).

```
0000DAA1B03D  TYPE( UNDO  REDO )  URID(0000DAA1AF0A)
              LRSN(C139E7FB8096)  DBID(0104)  OBID(0006)  PAGE(00000002)       10:5
              SUBTYPE(UPDATE IN-PLACE IN A DATA PAGE)  CLR(NO)
              PROCNAME(DSNIREPR)

     *LRH* 0048003D 06000001 0E800000 DAA1AF0A 0000DAA1 B0000626 0000DAA1 B000C139
           E7FB8096 0001
     *LG** 80010400 06000000 0200C12E 79FA6FDE 2D00
     0000  00100102 00388200 004A6652 58565258
```

3.2. Pav. DB2 log failo pavyzdys

DB2 log failo pradžioje yra įrašyta informacija apie įrašus, kurie yra saugomi šioje failo dalyje, t. y. kokią informaciją ji reprezentuoja, ar tai informacija apie vykdytas SQL užklausas, ar tai informacija apie sisteminius pakeitimus ir pan.

log failo antraštė (LRH) yra fiksuoto 36 baitų ilgio. Iš šios failo dalies DB2 serveris gali suprasti, į kokį pradinį tašką reikėtų grąžinti pakeitimus, jei įvyko nenumatytas sistemos sutrikdymas (žiūrėti lentelę 6).

6 lentelė: DB2 log failo antraštės baitų reikšmių reprezentavimas

Poslinkis	Ilgis	Interpretacija
0x00	2	Log įrašo ilgis
0x02	2	Prieš tai buvusio log įrašo ilgis
0x04	2	Įrašo tipas
0x06	2	Subtipas
0x08	1	Resurso valdytoja ID
0x09	1	Flag
0x10	8	URID
0x18	6	Prieš tai buvusio įrašo RBA reikšmė
0x24	1	DB2 versija
0x25	1	HDR ilgis
0x26	6	UNDO sekantis ilgis
0x32	6	Laiko žymė

Informacija saugoma log failo subantraštėje (žiūrėti lentelę 7) nusako objektą, su kurio buvo atlikti tam tikri veiksmai, ir kurioje objekto vietoje (puslapyje) buvo atlikti tie veiksmai.

7 lentelė: DB2 log failo subantraštės baitų reikšmių reprezentavimas

Poslinkis	Ilgis	Interpretacija
0x00	1	Flag
0x01	2	DBID reikšmė
0x03	2	PSID reikšmė
0x05	4	Puslapio numeris
0x09	1	Flag (galimos reikmės : 00/01/02)
0x10	6	RBA reikšmė prieš pakeitimą
0x16	1	Papildomi Flag

Informacija, kuri yra surašyta po subantraštės, vadinama pagrindine log informacija. Būtent šioje dalyje aprašoma informacija apie SQL užklausos tipą, kokia reikšmė buvo pakeista lentelėje ir į kokią reikšmę ji buvo pakeista (žiūrėti lentelę 8),.

8 lentelė: DB2 log failo pagrindine dalies baitų reikšmių reprezentavimas

Poslinkis	Ilgis	Interpretacija
0x00	2	Poslinkis
0x02	2	Bendras Log įrašo ilgis
0x04	1	Flag
0x05	1	ID
0x06	2	OBID
0x08	1	Flag
0x09	1	Flag
0x10	2	Pirmo pakeisti baito poslinkis
0x12	6	Sena reikšmė
0x18	6	Nauja reikšmė

Kaip iliustruojantį pavyzdį panagrinėkime SQL UPDATE užklausą pateikta apačioje.

UPDATE EMP

SET SALARY = SALARY + 1000

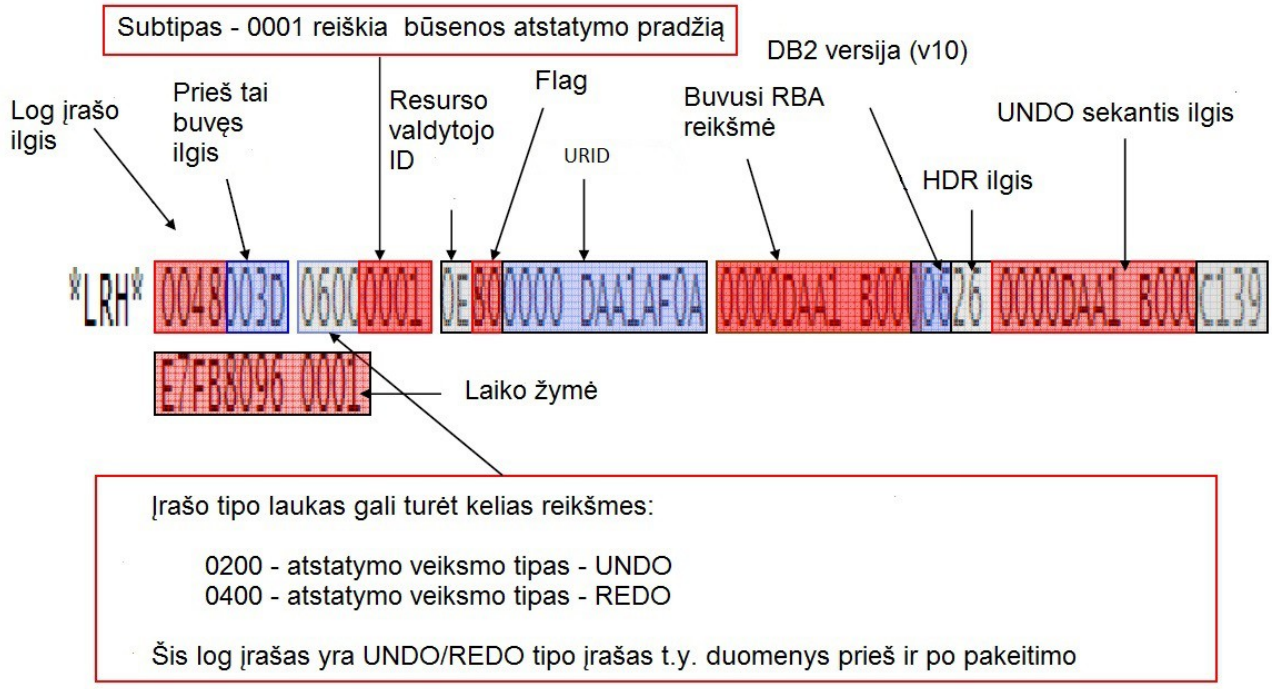
WHERE SALARY = 95652,58

Minėtoji užklausa atlyginimo eilutės reikšmę 95652,58 padidina 1000 vienetų. Šis veiksmas log faile atvaizduojamas taip (žiūrėti pav. 3.3.).

```
TYPE( UNDO  REDO )  URID(0000DAA1AF0A)
LRSN(C139E7FB8096)  DBID(0104)  OBID(0006)  PAGE(00000002)
SUBTYPE(UPDATE IN-PLACE IN A DATA PAGE)  CLR(NO)
PROCNAME(DSNIREPR)
```

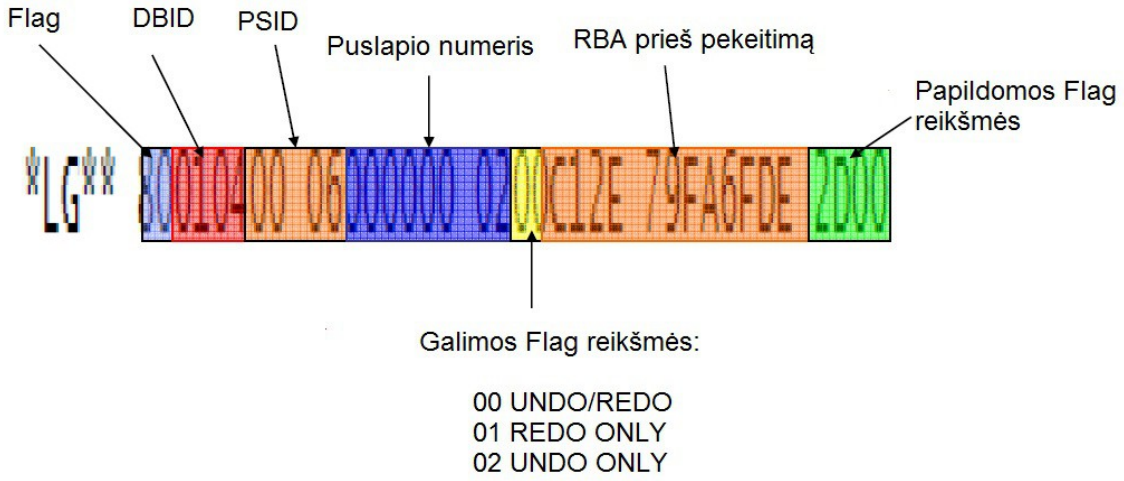
3.3. pav. DB2 log failo pradžios informacijos pavyzdys

log Failo LRH dalyje mes matome informaciją apie laiką, kada buvo užfiksuotas šis veiksmas. Vieta log įraše, kuri apibrėžia objektą iki pakeitimų (RBA), visą log informacijos apie šį veiksmą ilgį, vartotoją, kuris atliko šį veiksmą ir t. t. (žiūrėti pav. 3.4.).



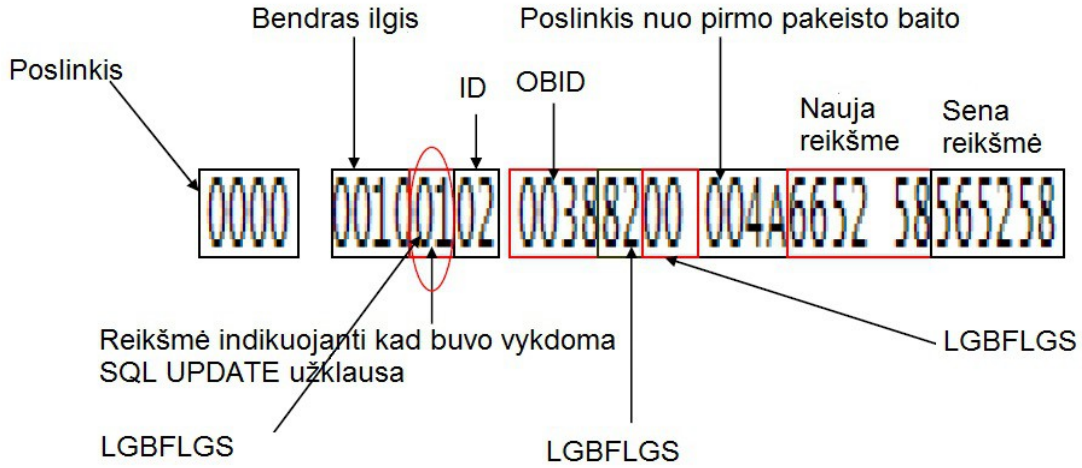
3.4. pav. DB2 log failo antraštės baitų reikšmių reprezentavimo pavyzdys

Failo subantraštėje matome informaciją, nurodančią kokiame duomenų bazės objekte buvo vykdomas pakeitimas ir kurioje objekto vietoje. Šiuo atveju, lenteles antrame puslapyje (žiūrėti pav. 3.5.).



3.5. pav. DB2 log failo subantraštės baitų reikšmių reprezentavimo pavyzdys

Paveikslas 3.6., nurodo kaip atrodo log failo pagrindinė dalis vykdant ankščiau minėtą užklausą.



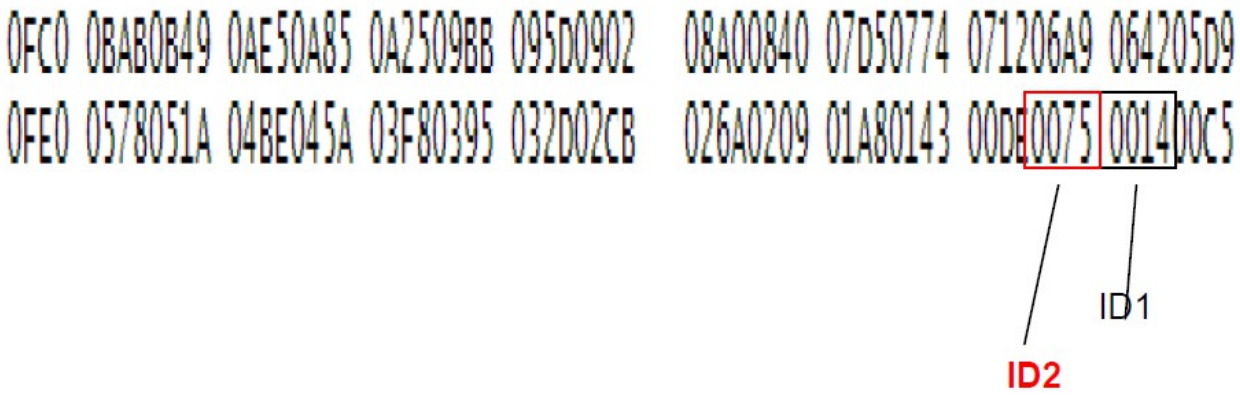
3.6. pav. DB2 log failo pagrindinės dalies baitų reikšmių reprezentavimo pavyzdys

Taigi, ką mes žinome apie įvykdytą užklausą, analizuojant DB2 serverio log failą:

- Duomenys, kurie buvo pakeisti lentelėje yra saugomi jos antrame puslapyje;
- Antras ID puslapyje nurodo konkretaus stulpelio eilutės vietą;
- Pirmas pakeistas bitas yra x'4A' nuo stulpelio eilutės pradžios;
- log įrašas yra UNDO/REDO tipo ir yra dalinis duomenų atvaizdavimas;
- Senos informacijos reikšmė buvo x'565258';

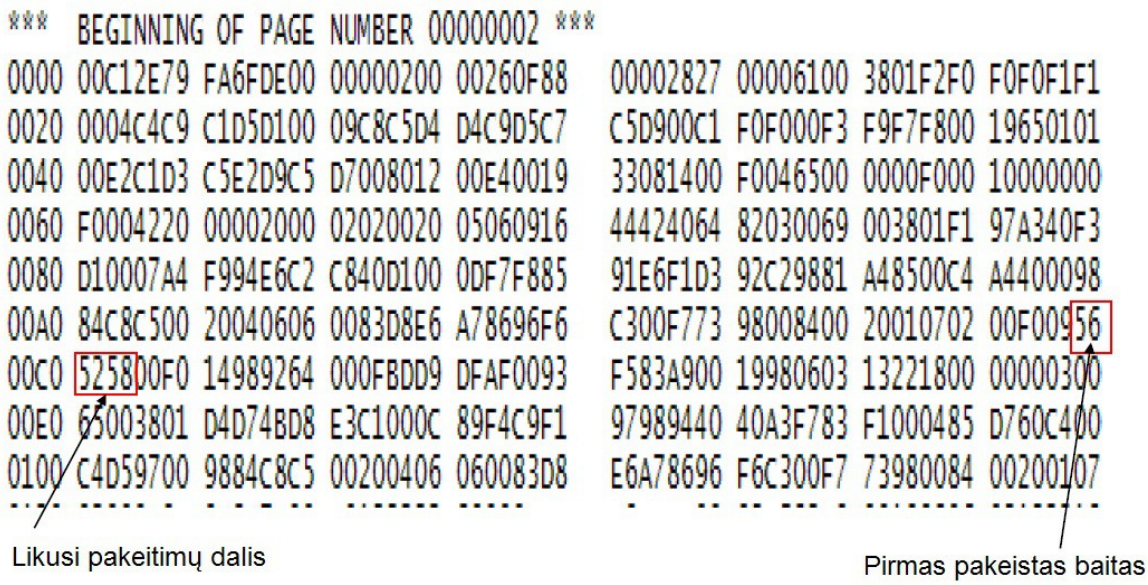
- Naujos informacijos reikšmė yra x'665258'.

Tam, kad būtų galima atsakyti į klausimą, kokios kitos reikšmės yra minėtame stulpelyje, reikia analizei pasitelkti lentelės aprašą šešioliktainės formatu. Tam yra naudojama serverio pagalbinė programa DSN1PRNT. Jos dėka, prieš įvykdant reikšmių pakeitimą, SQL užklausa sugeneruojam failą, kurį pradedame analizuoti nuo antrojo puslapio, nes jame buvo padaryti pakeitimai. Tam, kad būtų galima surasti antrą ID, reikia nuo puslapio pabaigos atimti du baitus, tad kiti du baitai nurodo poslinkį nuo puslapio pradžios (žiūrėti pav. 3.7.).



3.7. pav. DB2 DSN1PRNT programos sugeneruoto failo pavyzdys Nr. 1

Taigi, ID2 turi reikšmę x'0075', tad tam, kad būtų galima atrasti pirmojo pakeisto baito poslinkį, mes turime prie x'004A' pridėti x'0075', tad pirmas pakeistas baitas yra poslinkyje x'BF'. Reikšmių, esančių šiame poslinkyje, ieškome DSN1PRNT sugeneruotame faile (žiūrėti pav. 3.8.).



3.8. Pav. DB2 DSN1PRNT programos sugeneruoto failo pavyzdys Nr. 2

Stulpelis SALARY turi DECIMAL(9,2) duomenų formatą. Šio stulpelio apibrėžimas laikomas penkių baitų dydžio lauke duomenų puslapyje. Pilna šios stulpelio eilutės reikšmė – x'009565258'. Atsižvelgiant į duotą duomenų tipą, mes gauname, kad skaitinė SALARY stulpelio eilutės reikšmė yra 95652,58.

3.3. Išvados

Šioje dalyje buvo aprašyta duomenų bazės serverio log failų struktūra. Iš viso to, kas buvo paminėta šioje dalyje, galima daryti šias išvadas:

- Naudojantis mokslinio tyrimo metodu, kuris buvo skirtas SQL užklausų rekonstravimui MySQL serveryje, naudojant jo log failus , buvo rekonstruotos SQL užklausos DB2 duomenų bazės serveryje. Kad būtų pasiektas šis tikslas, anksčiau minėtas metodas buvo adaptuotas, taip, kad jis būtų pritaikytas kitam duomenų bazės serveriui su visiškai kita failų struktūra.
- Failų struktūra abejose duomenų bazių serveriuose yra skirtinga, bet turinti tam tikrų panašumų, kurie būdingi ir visiems kitiems duomenų bazių serveriams.
- Norit rekonstruoti bet kokią serveryje vykdytą užklausą, reikia turėti informaciją apie failo struktūrą ir šioje struktūroje esančių baitu reikšmes. Žinant šią informaciją,

galima sėkmingai identifikuoti, kokia informacija ir kokiu būdu buvo pakeista.

- Toks užklausų rekonstravimas, naudojant sisteminius failus, gali būti naudojamas ir tais atvejais, kai užklausos vykdomos sistemos administratoriaus arba asmens, turinčio tokias teises.

Kitoje darbo dalyje bus pateikiami eksperimentinio tyrimo rezultatai ir šio tyrimo eiga.

4. Eksperimentinis tyrimas

Eksperimentiniame tyrime buvo vertinama esama kibernetinių nusikaltimų tyrimo metodika ir pasiūlytas naujas log failų analizės metodas. Naujo metodo pagalba buvo analizuojami DB2 duomenų bazių serverio log failai ir jų struktūra.

4.1. Tyrime naudota debesies infrastruktūra

Lentelėje apačioje pateikiama informacija apie eksperimentinio tyrimo metu naudotus sisteminius išteklius.

(9 lentelė: Eksperimento metu naudoti kompiuteriniai ištekliai)

Debesies tipas	IaaS (privatus)
Duomenų bazės serveris	DB2 z/OS v 10.1
Debesies Serveris	Mainframe zEC12
Operacinė sistema	Z/OS 2.1.
Procesorius	2 x zIIP procesoriai po 5,5GHz
Virtuali atmintis	512GB
Duomenų saugyklos dydis	15TB x 6
Duomenų bazė serverio apkrova	50 transakciju/sec
Naudojamos duomenų bazės dydis	3GB

4.2. Tyrimo eiga

Prieš pradedant rekonstruoti užklausas iš log failų, eksperimento metu serveryje buvo sukurta testavimui skirta lentelė pavadinimu EMP. Su EMP lentele tyrimo metu buvo atliekamos duomenų manipuliacijos. Jos struktūra ir duomenų tipai pateikti žemiau.

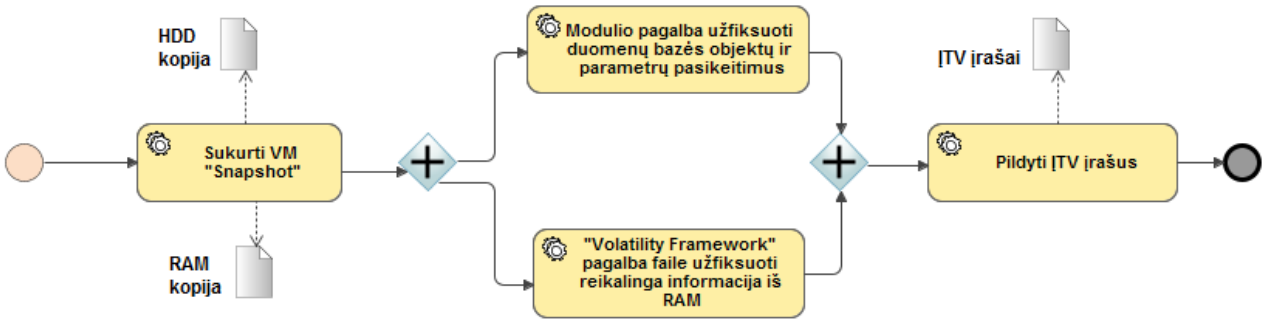
```
CREATE TABLE EMP
    (EMPNO    CHAR(6)                 NOT NULL,
     FIRSTNME  VARCHAR(12)              NOT NULL,
     MIDINIT   CHAR(1)                 NOT NULL,
     LASTNAME  VARCHAR(15)              NOT NULL,
     WORKDEPT  CHAR(3)                   ,
     PHONENO   CHAR(4)       CONSTRAINT NUMBER CHECK
```

```
        (PHONENO >= '0000' AND
         PHONENO <= '9999')              ,
HIREDATE  DATE                           ,
JOB       CHAR(8)                        ,
EDLEVEL   SMALLINT                       ,
SEX       CHAR(1)                        ,
BIRTHDATE DATE                           ,
SALARY    DECIMAL(9,2)                   ,
BONUS     DECIMAL(9,2)                   ,
COMM      DECIMAL(9,2)                   ,
PRIMARY KEY (EMPNO)                      ,
FOREIGN KEY RED (WORKDEPT) REFERENCES DSN8A10.DEPT
        ON DELETE SET NULL )
```

4.1. Pav. Tyrime naudotos EMP lentelės sukūrimas

Vėliau į šią lentelė buvo įterpti duomenis naudojantis specialiomis DB2 serverio programomis. Kokie duomenys buvo įterpti ir koks jų kiekis, pateikta priede B.

Prieš pradedant vykdyti SQL užklausų rekonstravimą iš serverio failų, tyrimo pradžioje buvo vertinama esama tradicinė nusikalstamos veikos tyrimo metodika ir jos galimybės atlikti tokį tyrimą debesies tipo serveriuose. Buvo vertinamas procesas, kurio metu daroma kietojo disko dvejetainė kopija bei operatyvios atminties kopija.

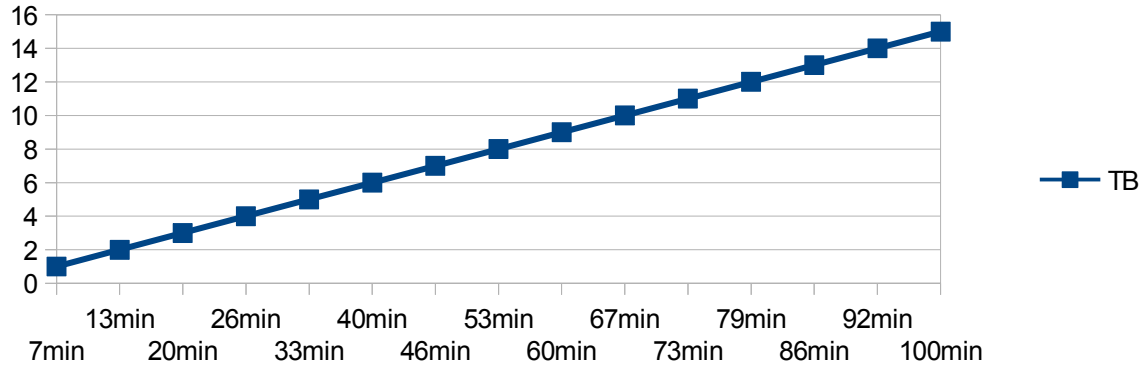


4.2. pav. Tradicinės nusikalstamos veikos kompiuterizuotoje aplinkoje BNMP proceso diagrama

Atliekant minėtus veiksmus IaaS tipo debesies serveryje buvo susidurta su tam tikromis problemomis. IaaS tipo serveris turi 15TB dedikuotą duomenų saugyklą, duomenis šioje saugykloje yra replikuojami šešis kartus. Bandant padaryti dedikuotos saugyklos dvejetainę kopiją, paaiškėjo,

kad šis procesas reikalauja daug laiko (žiūrėti lentelę žemiau).

10 lentelė: Dedikuotos saugyklos kopijų kūrimo laikas



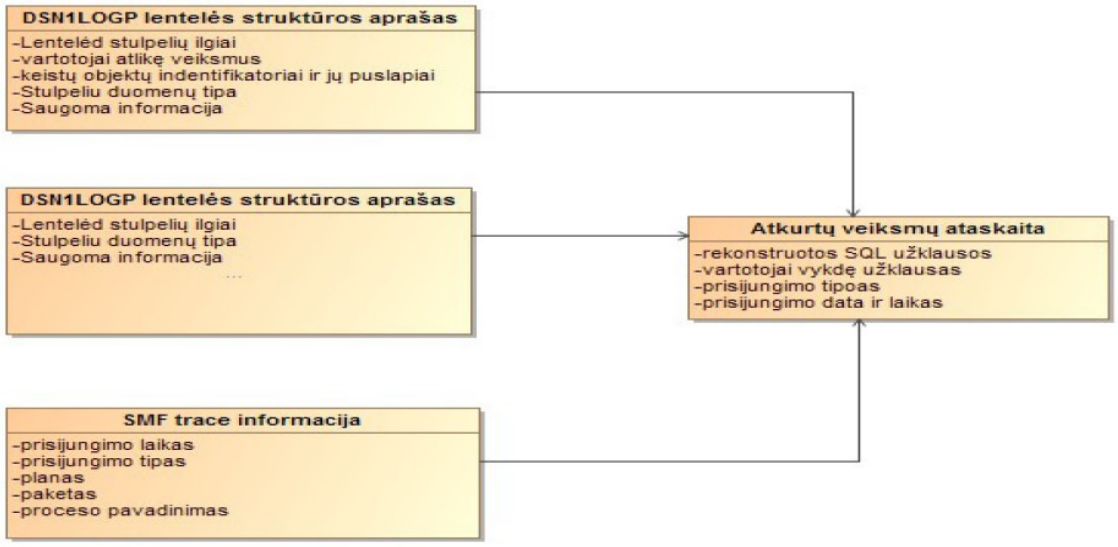
1TB disko kopijos darymas užtruko 7 min., o visos saugyklos kopija buvo daroma 1 val. 40 min. Turint omenyje, kad duomenis yra replikuojami šešis kartu, tai reikštų, kad reikiamas laikas šiam procesui užtrūktų šešis kartus ilgiau. Turint galvoje, jog tokiu atveju dauguma procesų, vykdomų šiame serveryje, būtų tiesiog sustabdyti, galima teigti, kad šis procesas nors teoriškai yra įmanomas, tačiau praktiškai yra neįgyvendinamas, nes tokiu atveju debesies serveris netenka savo esminių bruožų – prieinamumo ir pasiekiamumo.

Kalbant apie IaaS virtualios atminties kopijavimą, galima teigti, kad šis dalykas taip pat nors ir teoriškai yra įmanomas, tačiau praktiškai yra neįgyvendinamas. Nes tokiu atveju butų sustabdyta sistemos veikla. Nėra tiksliai žinoma į kiek loginių dalių yra padalintas debesies serveris ir kiek operatyvios atminties vietos yra dedikuota vienai loginei particijai, kurioje yra suinstaliuotas DB2 serveris. Dėl šių priežasčių buvo pereita prie naujos metodikos taikymo ir tyrimo. Tai sąlygojo keli veiksimai:

- Informacija turi būti surinka neapribojant serverio pasiekiamumo kitiems vartotojams;
- Taikant standartinį tyrimo metodą surinktos informacijos kiekis būtų milžiniškas, todėl toks tyrimas gali užtrukt neribotą laiko tarpą, o tai sulėtintų arba visai sustabdytų teisminį procesą.

4.3. DB2 log failų analizavimas

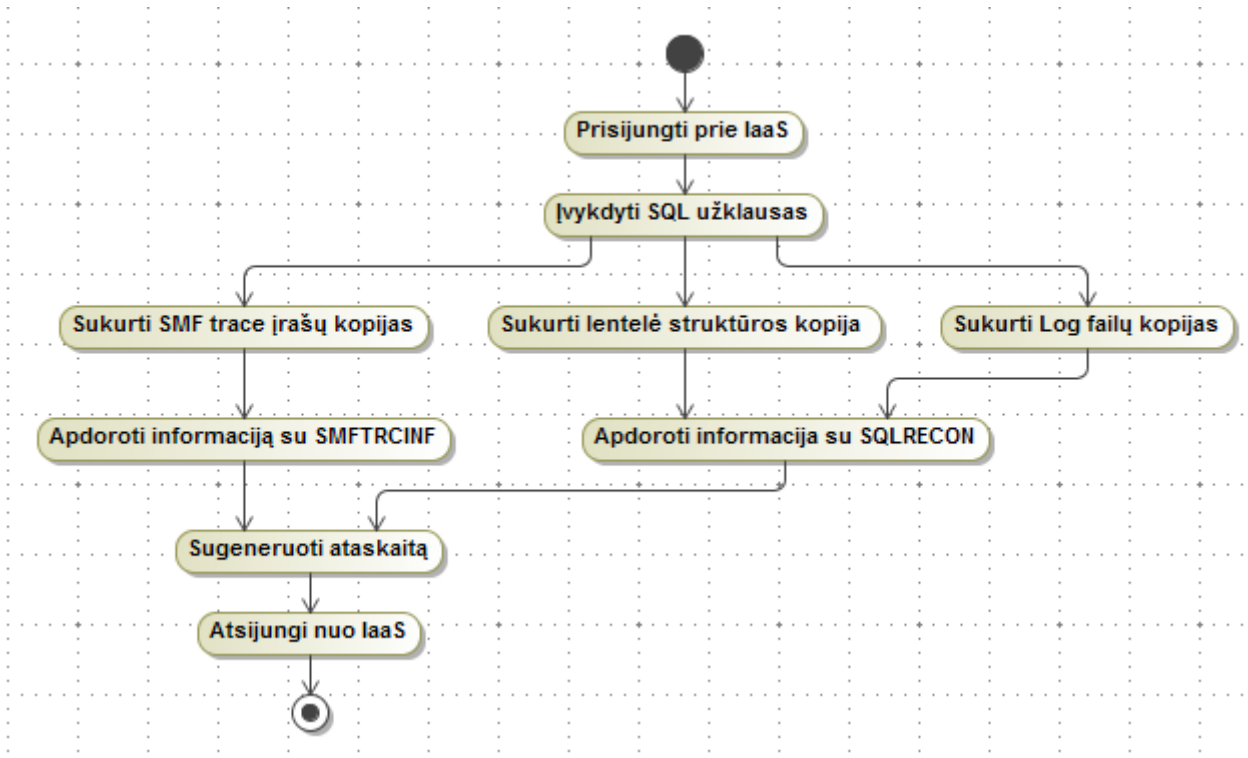
Vykdant DB2 serverio failų analizę buvo sukurtos kelios programos REXX ir COBOL programavimo kalba. Šios kalbos buvo pasirinks atsižvelgus į IaaS serveryje suinstaliuotą operacinės sistemos versiją. Žemiau pateikiamas koncepcinis šio prototipo duomenų modelis.



4.3. pav. Koncepcinis duomenų modelis

Tam, kad būtų galima rekonstruoti serveryje vykdytas užklausas ir ir pateikti pateikti ataskaitą, buvo vykdomi šie veiksmai (žiūrėti pav. 4.4.):

- Prisijungta prie IaaS serverio
- Įvykdytos SQL INSERT/UPDATE/DELETE užklausos;
- Sukurtos DB2 serverio log failų kopijos;
- Sukurtos SMF trace failų kopijos;
- Apdorotas log failas SQLRECON programa;
- Apdoroti SMF trace įrašai SMFTRCINF programa;
- Sugeneruota atskaita;
- Atsijungta.



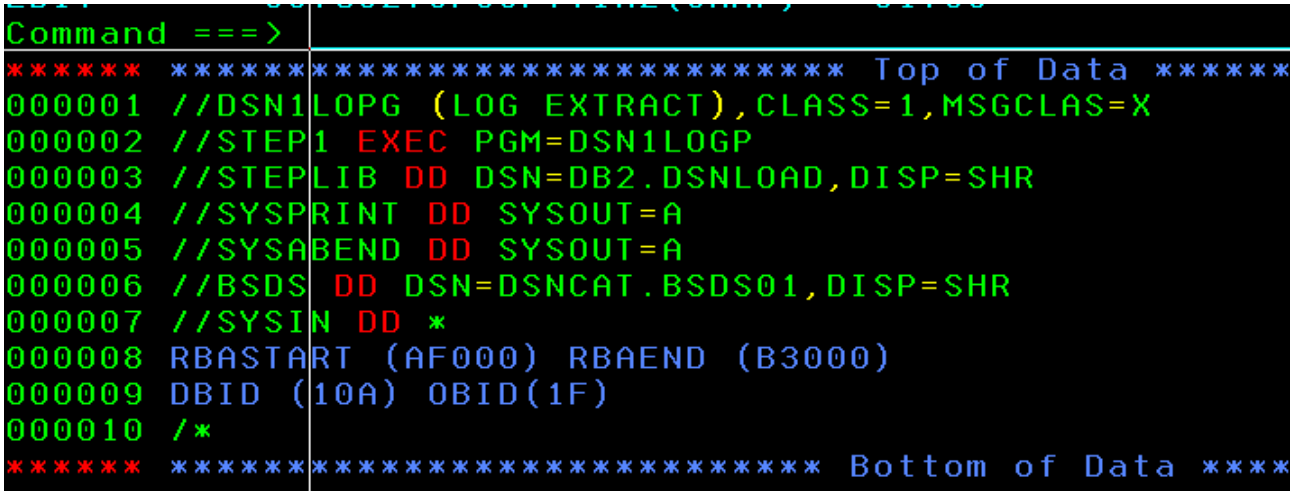
4.4. pav. Koncepcinis log failų analizės modelis

Eksperimento pradžioje buvo į vykdytos kelios SQL užklausos (žiūrėti pav 4.5.).

```
000006
000007 UPDATE EMP
000008 SET SALARY = SALARY + 1000
000009 WHERE SALARY = 95652.58 ;
000010 ---------+---------+---------+---------+---------+---------+
000011 DSNE615I NUMBER OF ROWS AFFECTED IS 1
000012 DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
000013 ---------+---------+---------+---------+---------+---------+
000014
000015  INSERT INTO EMP
000016      VALUES ('000333','HACK','T','MASTER','D11','2866',
000017              '1981-08-10','ANALYST',16,'F','1956-05-22',
000018             16345,500,2300);
000019 ---------+---------+---------+---------+---------+---------+
000020 DSNE615I NUMBER OF ROWS AFFECTED IS 1
000021 DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
000022 ---------+---------+---------+---------+---------+---------+
000023
000024 DELETE FROM EMP
000025 WHERE FIRSTNME = 'EVA';
000026 ---------+---------+---------+---------+---------+---------+
000027 DSNE615I NUMBER OF ROWS AFFECTED IS 1
000028 DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
000029 ---------+---------+---------+---------+---------+---------+
```

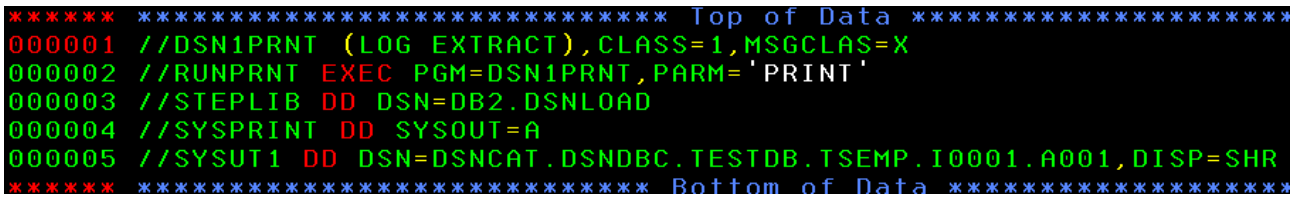
4.5. pav. DB2 serveryje įvykdytos SQL užklausos

Sėkmingai įvykdžiusi šias užklausas, naudojantis JCL skriptais (žiūrėti pav 4.6.), serveryje buvo išgautas log failo įrašas, kuriame buvo laikoma informacija apie duomenų manipuliacijas EMP lentelėje.



4.6. pav. JCL skriptas DSN1LOGP programai vykdyti

Taigi, šio skripto dėka gauname informaciją, kuri reprezentuoja veiksmus, atliktus su EMP lentele. log įrašo pavyzdys jau buvo pateiktas anksčiau šiame darbe. Tam, kad būtų galima rekonstruoti užklausas, reikia žinoti lentelės struktūrą, ši informaciją eksperimentinio tyrimo metu buvo surinka naudojantis DSN1PRNT serverio programa. Tai taip pat buvo padaryta naudojantis JCL skriptu (žiūrėti pav. 4.7.).



4.7. pav. JCL skriptas DSN1PRNT programai vykdyti

Sugeneruotas failas kartu su prieš tai DSN1LOGP programa sugeneruotu failu apdorojamas sukurtais skriptais, kurių dėka yra nuskaitoma ir į skaitomą formatą transformuojama informacija. Šių skriptų dėka sugeneruotas ataskaitos failas nusako laiką, kada buvo vykdomos

duomenų manipuliacijos, vartotoją, rekonstruoja vykdytas užklausas. Iš DB2 Trace papogramių išgauname informaciją apie tai, kokiu būdu vartotojas buvo prisijungęs prie serverio, laiką. Šios informacijos kiekis priklauso nuo DB2 serverio sisteminių parametrų. Verta paminėti, jog reikalui esant, ir naudojant DB2 serverio pagalbinę programą RECOVERY UTILITY, galima grąžinti lentelės duomenis į tą būvį, kuris buvo iki pakeitimų. Tam iš LRSN/RBA reikšmių, nurodytų ataskaitos faile, reikia atimti vienetą, o gautą reikšmę naudoti kaip įvesties reikšmę RECOVERY programoje (žiūrėti pav. 4.8.).

```
****** ********************************* Top of Data ********************************
000001 ------ LOG EXTRACT PRADZIA -------- ------ LOG EXTRACT PABAIGA --------
000002        Data Laikas Log RBA/LRSN             Data Laikas Log RBA/LRS
000003 ------------------------------ ------------------------------------
000004 2015/05/01 12:50:00 10A0E2584758 --> 2015/05/01 12:59:59 10A0E2B13C3
000005 CEEDD43050FC CEEDD66C6
000006 ------------------------------------------------------------------------
000007 -- TABLE: DEYGS2.EMP
000008 -- DBID : 1410 PSID: 27 OBID: 28
000009 -- URID : 10A0E2863034 RBA: 10A0E28630F3
000010 -- LRSN : CEEDD4FB75A6
000011 ------------------------------------------------------------------------
000012 UPDATE DEYGS2.EMP
000013 SET SALARY = 95652.58
000014 WHERE
000015 SALARY = 95652.58
000016 ;
000017 INSERT INTO DEYGS2.EMP
000018 ( EMPNO , FIRSTNME , MIDINIT , LASTNAME , WORKDEPT , PHONENO ,
000019 HIREDATE , JOB , EDLEVEL , SEX , BIRTHDATE , SALARY , BONUS ,
000020 COMM )
000021 VALUES
000022 ( '000333' , 'HACK' , 'T' , 'MASTER' , 'D11' , '2866' ,
000023 '1981-08-10' , 'ANALYST ' , 16 , 'F' , '1956-05-22' ,
000024 16345.00 , 500.00 , 2300.00 )
000025 ;
000026 DELETE FROM DEYGS2.EMP
000027 WHERE EMPNO = '000070'
000028 AND FIRSTNME = 'EVA'
000029 AND MIDINIT = 'D'
000030 AND LASTNAME = 'PULASKI'
000031 AND WORKDEPT = 'D21'
000032 AND PHONENO = '7831'
000033 AND HIREDATE = '1980-09-30'
000034 AND JOB = 'MANAGER '
000035 AND EDLEVEL = 16
000036 AND SEX = 'F'
000037 AND BIRTHDATE = '1953-05-26'
000038 AND SALARY = 36170.00
000039 AND BONUS = 700.00
000040 AND COMM = 2893.00
000041 ;
000042
000043 ------------------TRACE REPOR ----------------------------------
000044
000045 " DB2 AUDIT REPORT FROM SMF 102 RECORDS
000046                              " ACCESS TO TABLE EMP "
000047  SMF
000048  REC IFC    SMF         SMF      CONNECT    PLAN
000049  TYP  ID  LOG DATE  LOG TIME      NAME      NAME   OPERATOR/ID DBID NAME
000050  ____ ____ ________ ____________ ________ ________ ___________ ____ ________
000051  102  144 01/05/15 12:50:00.13 TSO      ADB       REMORTUSER 1410 EMP
000052
000053
****** ********************************* Bottom of Data *****************************
```

4.8. pav. SQLRECON ir SMFTRCINF programos išeities failas

4.4. Išvados

Atlikus eksperimentinį tyrimą galima padaryti šias išvadas:

- Tradicinis nusikalstamos veikos tyrimo būdas, kurio metu yra padaroma operatyvios atminties kopija kartu su kietojo disko arba duomenų talpyklos kopija, yra sunkiai įgyvendinamas debesies tipo serveriuose dėl ilgos tokio būdo trukmės;
- Debesies tipo serveriai dažnais atvejais disponuoja milžiniškais informaciniais resursais. Tyrime naudoto IaaS tipo debesies serveris turėjo 15TB dedikuotą duomenų talpyklą. Tokio dydžio talpyklos kopijavimas trunka beveik dvi valandas, o tai reškia, kad tuo metu serveris praktiškai nepasiekiamas;
- Operatyvios atminties kopijos padarymas debesies tipo serveryje praktiškai įmanomas, tačiau teoriškai beveik neįgyvendinamas procesas;
- Nors ir žinoma, kokio dydžio yra visa serverio operatyvioji atmintis, tačiau nėra žinoma į kiek virtualių mašinų ji yra paskirstyta, taip pat nėra žinoma, kurioje iš tų mašinų yra suinstaliuotas duomenų bazės serveris.
- Tam, kad nebūtų sutrikdyta debesies serverio veikla, naudojant pagalbines programas, galima padaryti momentines sisteminių failų kopijas, kurias išanalizavus galima rekonstruoti serveryje vykdytas užklausas.
- SQL užklausų rekonstravimas, naudojantis tik DBVS generuojamais failais, reikalauja labai gero supratimo apie serverio veikimo principus, sisteminius failus bei jose laikomą informaciją.

IŠVADOS

Išanalizavus elektroninių nusikaltimų pėdsakų fiksavimą debesų saugyklų kompiuterijos aplinkoje bei pasiūlius būdą, leidžiantį atkurti veiksmus tokiame serveryje, kai jokio tam skirto papildomo funkcionalumo neegzistuoja, galima daryt šias išvadas:

- Šiai dienai egzistuoja akivaizdi nusikalstamos veikos tyrimo problematika debesies tipo technologijose;
- Daugelis autorių siūlo diegti papildomas stebėsenos sistemas, kurios rinktų informaciją apie sistemoje vykdytus veiksmus, tačiau jei tokių sistemų nėra įdiegta serveryje, nusikalstamos veikos tyrimas pasidaro gan komplikuotas, o neretai praktiškai neįmanomas;
- Debesies tipo serveriai turi savo reikalavimus iš vartotojų pusės. Įmonė teikianti tokias paslaugas turi užtikrinti, kad serveris visada bus pasiekiamas paslaugos gavėjams, kitu atveju gresia reputacijos ir pelno praradimas. Be to, vykdant nusikalstamos veikos tyrimą tokio tipo serveryje, neįmanoma užtikrinti nepertraukiamo jo pasiekiamumo;
- Nusikalstamos veikos tyrimą taip pat apsunkina ir tas faktas, kad fiziškai neįmanoma pasiekti šio serverio, nes ji dažniausiai yra kitoje šalyje;
- Naudojantis kitų autorių studijomis ir surinkta patirtimi, buvo pritaikytas MySQL duomenų bazės serverių log failų tyrimo metodas analizuojant tokio paties tipo DB2 duomenų bazės serverio failus;
- Pasiūlyto DB2 duomenų bazės serverio log failų analizės metodo dėka eksperimentinio tyrimo metu pavyko rekonstruoti užklausas serveryje. Šis būdas leido surinkti informaciją iš debesies duomenų bazės serverio nenutraukiant jo veikimo;
- Pasiūlyta metodika taip pat tinkama tais atvejais, kai minėtosios užklausos buvo vykdomos sistemos administratoriaus arba asmens, turėjusio tokias teises.

Dabartiniai duomenų bazės serveriai savo lentelių struktūroje gali laikyti ne tik organizuotus duomenis, bet ir neorganizuotus, pavyzdžiui XML tipo duomenis. Todėl ateities tyrimai šia tematika turėtų kreipti dėmesį į tai, kaip galima būtų rekonstruoti pakeitimus, vykdytus tokio tipo lentelėse. Kartu taip pat ateityje reikėtų ištirti tokios informacijos rekonstravimo duomenims, kurie buvo suspausti.

LITERATŪROS SĄRAŠAS

[1]“Security Functions of IBM DB2 10 for z/OS“ [žiūrėta 2015-05-12]. Prieiga per Internetą: http://www.ibm.com/redbooks;

[2] NIST, The NIST definition of cloud computing, September 2011, [žiūrėta 2014-06-12]. Prieiga per Internetą: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf;

[3] [žiūrėta 2014-06-12]. Prieiga per Internetą: http://cioresearchcenter.com/wordpress/wp-content/uploads/2010/12/UnderstandingCloudComputing-e1291188677368.jpg;

[4] Fabian Gremper, *Relational Cloud: A Database-as-a-Service for the Cloud*, IGI Global, April 2011;

[5] Cloud Security Alliance, Security guidance for critical areas of focus in cloud computing V3.0, 2011,žiūrėta 2014-06-12]. Prieiga per Internetą: https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf ;

[6] U. Oktay, O.K. „*Sahingoz Attack Types and Intrusion Detection Systems in Cloud Computing“* įtraukta 6th *International Information Security & Cryptology Conference* ;

[7] Qussai Yaseen, Brajendra Panda „*Tackling Insider Threat in Cloud Relational Databases“* įtraukta IEEE/ACM Fifth International Conference on Utility and Cloud Computing 2012 ;

[8] „McAfee, *Database Security in Virtualization and Cloud Computing Environments“* [žiūrėta 2014-06-12]. Prieiga per Internetą:https://portal.mcafee.com/downloads/General%20Documents/database_security_in_virtualization_and_cloud_computing_environments.pdf ;

[9]W. Xin, H. Ting-lei, and L. Xiao-yu, "*Research on theIntrusion detection mechanism based on cloud computing*," įtraukta *Intelligent Computing and Integrated Systems (ICISS), International Conference* 2010;

[10] C. Mazzariello, R. Bifulco, and R. Canonic, “*Integrating a Network IDS into an Open Source Cloud Computing Environment*,” įtraukta *Sixth Internationl Conference on Information Assurance and Security (IAS)*, 2010;

[11] S. Roschke, F. Cheng, and C. Meinel, *"Intrusion Detection in the Cloud*", In Proceedings of Workshop Security in Cloud Computing (SCC'09), IEEE Press, Chengdu, China, pp. 729-734 (December 2009);

[12]Anthony Reyes; Richard Brittson “*Cyber Crime Investigations*” , Syngress, 2007

[13] Chang-Tsun Li “*Emerging Digital Forensics Applications for Crime Detection*” , Prevention, and Security”, IGI Global, 2013;

[14] Information Resources Management Association, USA, “*Cyber Crime*”, IGI Global, 2011

[15] Keyun Ruan "*Cybercrime and Cloud Forensics*", IGI Global, 2012
[16] Kevvie Fowler *SQL Server Forensic Analysis*. Addison-Wesley Professional, 2008, ISBN 0-321-54436-6
[17] Diane Barrett, Greg Kipper *Virtualization and Forensics*, Syngress 2010, ISBN 978-1-59749-557-8
[18] Ming Xu *A Forensic Analysis Method for Redis Database Based on RDB and AOF File* Journal Of Computers, Vol. 9, No. 11, November 2014
[19] Pavlou, R. Snodgrass. *Forensic analysis of database tampering*. ACM Transactions on Database Systems, 33(4), November 2008.
[20] Jitendra R Chavan *Database Forensic Analysis Using log Files* International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
[21] Harmeet Kaur Khanujal *A Framework For Database Forensic Analysis* Computer Science & Engineering: An International Journal (CSEIJ), Vol.2, No.3, June 2012
[22] Karen B. Alexander *Database Forensic Analysis* International Journal of Advance Research in Computer Science and Management Studies, Volume 2, Issue 3, March 2014
[23] MySQL administrative and Utility programs [žiūrėta 2015-05-12]. Prieiga per Internetą https://dev.mysql.com/doc/refman/4.1/en/programs-admin-utils.html
[24] Administration Guide IBM DB2 10 for z/OS [žiūrėta 2015-03-12]. Prieiga per Internetą: http://www.ibm.com/redbooks
[25] BM DB2 z/OS logging [žiūrėta 2015-01-12]. Prieiga per Internetą ftp://public.dhe.ibm.com/software/zseries/pdf/DB2-logging-Webcast-Bartak.pdf
[26] IBM DB2 Trace commands [žiūrėta 2015-01-12]. Prieiga per Internetą http://www01.ibm.com/support/knowledgecenter/SSEPEK_10.0.0/com.ibm.db2z10.doc.comref/src/tpc/db2z_cmd_starttrace.dita
[27] Peter Fr¨uhwirt, Peter Kieseberg *„InnoDB Database Forensics: Reconstructing Data Manipulation Queries from Redo logs"* įtraukta *SBA ResearchVienna, Austria*
[28] Marcel Niefindt „*Forensic Analysis of MySQL DB Systems" įtraukta* S*ANS DFIR Conference paper,* Prague 2014

PRIEDAI

Priedas A. Flag baitų reikšmės

B'10000000'	YES IF UPDATE TO HASH ANCHOR NO IF RECORD CHANGE.
B'zXXzzzzz'	TYPE OF SUB-OPERATION: (LGBSUBOP) XX = 00 IF IN-PLACE RECORD UPDATE OR HASH-ANCHOR UPDATE. 10 IF INSERT 01 IF DELETE 11 IF NON-IN PLACE REC UPDATE
B'01000000'	YES IF RECORD INSERTION
B'00100000'	YES IF RECORD DELETION
B'00010000'	YES IF ID-MAP ENTRY ADDED OR DELETED FOR RECORD ISRT/DLET
B'00001100'	TYPE OF DATA CAPTURE OPERATIONS
B'00001000'	BIT =1, TABLE IS DEFINED FOR DATA CAPTURE.
B'00000100'	BIT =1, RECURSIVE DATA CAPTURE OPERATION NOT DONE.
B'00000010'	BIT =1, RI CAUSED REPLACE OR DELETE.
B'00000001'	YES IF LOG RECORD IS WRITTEN ON BEHALF OF AN SQL UPDATE OPERATION. YES AND LGBSUBOP = '10' - LOG RECORD REPRESENTS THE INSERTION OF AN OVERFLOW RECORD ON BEHALF OF AN SQL UPDATE OPERATION.
B'10000000'	1 IF PARTIAL IMAGE LOGGED DOES NOT INCLUDE PREFIX DIFFERENCES IN PGSLTH (DATA RECORD LENGTH).
B'01000000'	1 IF RECORD HAS COMPRESSED DATA.
B'00100000'	1 IF RECORD HAS EXPANDED FORMAT LIKE CDC LOG RECORDS.
B'00010000'	1 IF GROSS LOCK IS ON
B'00001111'	VARIATION NUMBER: 0000 INSERT OR DELETE LOG RECORD 0001 UPDATE VARIATION 1 LOG RECORD 0010 UPDATE VARIATION 2 LOG RECORD 0011 UPDATE VARIATION 3 LOG RECORD 0100 UPDATE VARIATION 4 LOG RECORD (EXCEPT 1ST LOG REC IN SERIES) 0101 UPDATE VARIATION 5 LOG RECORD (EXCEPT 1ST LOG REC IN SERIES) 0110 UPDATE VARIATION 6 LOG RECORD (EXCEPT 1ST LOG REC IN SERIES) 0111 UPDATE VARIATION 7 LOG RECORD (EXCEPT 1ST LOG REC IN SERIES) 1000 UPDATE VARIATION 8 LOG RECORD (EXCEPT 1ST LOG REC IN SERIES) 1001 UPDATE VARIATION 9 LOG REOCRD 1110 1ST LOG RECORD IN A VARIATION 4 OR 5 SERIES 1111 1ST LOG RECORD IN A VARIATION 6, 7 OR 8 SERIES

Priedas B. EMP lentelėje saugota informacija

EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE
000010	CHRISTINE	I	HAAS	A00	3978	1965-01-01
000020	MICHAEL	L	THOMPSON	B01	3476	1973-10-10
000030	SALLY	A	KWAN	C01	4738	1975-04-05
000050	JOHN	B	GEYER	E01	6789	1949-08-17
000060	IRVING	F	STERN	D11	6423	1973-09-14
000070	EVA	D	PULASKI	D21	7831	1980-09-30
000090	EILEEN	W	HENDERSON	E11	5498	1970-08-15
000100	THEODORE	Q	SPENSER	E21	0972	1980-06-19
000110	VINCENZO	G	LUCCHESSI	A00	3490	1958-05-16
000120	SEAN		O'CONNELL	A00	2167	1963-12-05
000130	DOLORES	M	QUINTANA	C01	4578	1971-07-28
000140	HEATHER	A	NICHOLLS	C01	1793	1976-12-15
000150	BRUCE		ADAMSON	D11	4510	1972-02-12
000160	ELIZABETH	R	PIANKA	D11	3782	1977-10-11
000170	MASATOSHI	J	YOSHIMURA	D11	2890	1978-09-15
000180	MARILYN	S	SCOUTTEN	D11	1682	1973-07-07
000190	JAMES	H	WALKER	D11	2986	1974-07-26
000200	DAVID		BROWN	D11	4501	1966-03-03
000210	WILLIAM	T	JONES	D11	0942	1979-04-11
000220	JENNIFER	K	LUTZ	D11	0672	1968-08-29
000230	JAMES	J	JEFFERSON	D21	2094	1966-11-21
000240	SALVATORE	M	MARINO	D21	3780	1979-12-05
000250	DANIEL	S	SMITH	D21	0961	1969-10-30
000260	SYBIL	P	JOHNSON	D21	8953	1975-09-11
000270	MARIA	L	PEREZ	D21	9001	1980-09-30
000280	ETHEL	R	SCHNEIDER	E11	8997	1967-03-24
000290	JOHN	R	PARKER	E11	4502	1980-05-30
000300	PHILIP	X	SMITH	E11	2095	1972-06-19

(EMPNO)	JOB	EDLEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM
(000010)	PRES	18	F	1933-08-14	52750.00	1000.00	4220.00
(000020)	MANAGER	18	M	1948-02-02	41250.00	800.00	3300.00
(000030)	MANAGER	20	F	1941-05-11	38250.00	800.00	3060.00
(000050)	MANAGER	16	M	1925-09-15	40175.00	800.00	3214.00
(000060)	MANAGER	16	M	1945-07-07	32250.00	600.00	2580.00
(000070)	MANAGER	16	F	1953-05-26	36170.00	700.00	2893.00
(000090)	MANAGER	16	F	1941-05-15	29750.00	600.00	2380.00
(000100)	MANAGER	14	M	1956-12-18	26150.00	500.00	2092.00
(000110)	SALESREP	19	M	1929-11-05	46500.00	900.00	3720.00
(000120)	CLERK	14	M	1942-10-18	29250.00	600.00	2340.00
(000130)	ANALYST	16	F	1925-09-15	23800.00	500.00	1904.00
(000140)	ANALYST	18	F	1946-01-19	28420.00	600.00	2274.00
(000150)	DESIGNER	16	M	1947-05-17	25280.00	500.00	2022.00
(000160)	DESIGNER	17	F	1955-04-12	22250.00	400.00	1780.00
(000170)	DESIGNER	16	M	1951-01-05	24680.00	500.00	1974.00
(000180)	DESIGNER	17	F	1949-02-21	21340.00	500.00	1707.00
(000190)	DESIGNER	16	M	1952-06-25	20450.00	400.00	1636.00
(000200)	DESIGNER	16	M	1941-05-29	27740.00	600.00	2217.00
(000210)	DESIGNER	17	M	1953-02-23	18270.00	400.00	1462.00
(000220)	DESIGNER	18	F	1948-03-19	29840.00	600.00	2387.00
(000230)	CLERK	14	M	1935-05-30	22180.00	400.00	1774.00
(000240)	CLERK	17	M	1954-03-31	28760.00	600.00	2301.00
(000250)	CLERK	15	M	1939-11-12	19180.00	400.00	1534.00
(000260)	CLERK	16	F	1936-10-05	17250.00	300.00	1380.00
(000270)	CLERK	15	F	1953-05-26	27380.00	500.00	2190.00
(000280)	OPERATOR	17	F	1936-03-28	26250.00	500.00	2100.00
(000290)	OPERATOR	12	M	1946-07-09	15340.00	300.00	1227.00
(000300)	OPERATOR	14	M	1936-10-27	17750.00	400.00	1420.00