KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS IR INFORMACINIŲ TECHNOLOGIJŲ SAUGOS STUDIJŲ PROGRAMA

NERIJUS SAIKAUSKAS

VEIKSMŲ KAIP ĮKALČIŲ SKAIČIAVIMŲ DEBESIES SAUGYKLOSE ATKŪRIMO METODIKA

Magistro darbas

Darbo vadovas
doc. dr. J. Toldinas

KAUNAS, 2013

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS IR INFORMACINIŲ TECHNOLOGIJŲ SAUGOS STUDIJŲ PROGRAMA

NERIJUS SAIKAUSKAS

VEIKSMŲ KAIP ĮKALČIŲ SKAIČIAVIMŲ DEBESIES SAUGYKLOSE ATKŪRIMO METODIKA

Magistro darbas

Darbo vadovas
doc. dr. J. Toldinas

Recenzentas
doc. dr. G. Činčikas

KAUNAS, 2013

SANTRAUKA

Skaičiavimų debesies (angl. *cloud computing*) technologijos sukūrimas suteikė galimybę padidinti kompanijų veiklos efektyvumą, tačiau sukėlė ir naujų problemų, viena kurių – skaitmeninės teismo ekspertizės (angl. *digital forensics*) atlikimas nutolusioje aplinkoje. Apskritai teigiama, kad jeigu skaičiavimų debesies paslauga nefiksuoja tinkamų audito įrašų, nustatyti įkalčius tampa sunku arba tiesiog neįmanoma. Deja, paprastai šiam tikslui siūlomas funkcionalumas yra gana ribotas arba iš viso neegzistuoja. Šiame magistriniame darbe yra siūloma nauja metodika-įrankis, Žurnalizavimo Paramos Sistema (ŽPS), programavimo kalba „Python" apjungianti skaitmeninės teismo ekspertizės atlikimui skirtas atvirojo kodo programines priemones „The Sleuth Kit" ir „The Volatility Framework", kuri padeda užfiksuoti ir atkurti vartotojų veiksmus kaip įkalčius skaičiavimų debesies saugyklose. ŽPS įgyvendina kitų autorių pasiūlytą unifikuotą audito įrašų formatą tokio pobūdžio aplinkoms ir sukuria save aprašančių duomenų efektą, kuris, manoma, yra svarbus žingsnis siekiant efektyviai tirti nusikaltimus skaičiavimų debesies saugyklose. Eksperimentinio tyrimo metu Žurnalizavimo Paramos Sistema pademonstravo aukštus efektyvumo rodiklius: jos pagalba pavyko atkurti daugiau kaip 65 % veiksmų priklausomai nuo vartotojų aktyvumo su sąlyga, kad virtualių mašinų (angl. *virtual machine*) kopijos buvo kuriamas ir analizuojamas ne rečiau kaip kas 5 min.

SUMMARY

Even though creation of cloud computing technology has provided opportunities to increase effectiveness of the companies, it has also generated new problems where one of them is digital forensics in the remote environments. It is generally agreed that if the service of a cloud doesn't record appropriate logs, identification of evidence becomes hard if not possible. Unfortunately, the existing functionality for this purpose is limited or absent all together. In this Master‘s thesis a new method-tool, Žurnalizavimo Paramos Sistema (ŽPS), has been proposed which combines open source digital forensic software The Sleuth Kit and The Volatility Framework with the help of Python programming language and helps to record and restore user activities in cloud storage environments. ŽPS implements unified logging format for such types of settings proposed by other authors and creates a data-centric effect which is thought to be an important step towards proper crime investigations in cloud storage environments. During experimental evaluation the method proved to be highly effective managing to reconstruct more than 65 % of user actions depending on their activeness when the copies of virtual machines have been created and analized not rarer than 5 minutes period.

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

ĮVADAS

Spartus informacinių technologijų (IT) vystymasis sudaro naujų galimybių didinti įmonių veiklos efektyvumą, tačiau kartu sukuria ir naujų problemų, iššūkių, kuriuos tenka spręsti, siekiant sėkmingai pasinaudoti naujovių teikiama nauda.

Skaičiavimų debesies technologija, nagrinėjama šiame magistriniame darbe, – ne išimtis. Ji priskiriama vieniems reikšmingiausių paskutinių metų pasikeitimų, nes ėmė keisti būdus, kaip kompanijos realizuoja savo kompiuterijos poreikį – jos dėka, staiga net ir mažiems verslo rinkos dalyviams atsivėrė neriboti ir svarbiausia įperkami skaičiavimų ištekliai: duomenų saugyklos, procesoriai, tinklai ir kt. Skaičiai kalba patys už save: prognozuojama, kad metinis globalus skaičiavimų debesies paslaugos IP srautas išaugs 3,5 karto nuo 1,2 zetabaitų 2012 m. iki 4,3 zetabaitų 2016 m. ir sudarys du trečdalius visų duomenų centrų srauto [22].

Nežiūrint į privalumus, skaičiavimų debesis kelia įvairių abejonių ir rūpesčių. Ypač baiminamasi dėl esamų galimybių paslaugos tiekėjui būti užpultam elektroninės erdvės nusikaltėlių, kurie gali pavogti klientų informaciją. Galima tiek tiesioginė, tiek netiesioginė (pasinaudojant kliento puse) debesies infrastruktūros kompromitacija, egzistuoja įvairios debesies pažeidžiamos vietos, kyla paslaugos tiekėjo šnipinėjimo grėsmė, be to, dėl nutolusių išteklių, sunkiai sprendžiamas audito ir skaitmeninės teismo ekspertizės atlikimo klausimas.

O būtent skaitmeninės teismo ekspertizės problema yra šiandien kaip niekada aktuali. Per pastarąjį dešimtmetį stipriai išaugo nusikaltimų, atliekamų elektroninėje erdvėje, skaičius. To pasekoje, įsikūrė daug kompanijų, kurios siūlo produktų, padedančius teisėsaugos organams nustatyti kas, ką, kur, kada ir kaip padarė nusikaltimus naudojantis kompiuterinėmis priemonėmis. Skaitmeninė teismo ekspertizė, kaip sfera, būtent ir išsivystė tuo pagrindu, kad būtų užtikrintas tinkamas elektroninių įkalčių pristatymas teismui. Tačiau kyla daug klausimų, ar skaičiavimų debesis gali būti tinkamai ištirtas skaitmeninės teismo ekspertizės tikslams.

Viena vertus, centralizuoti duomenys ir milžiniški kompiuterijos ištekliai turėtų palengvinti tyrėjų darbą. Kita vertus, egzistuoja esminiai trūkumai, kurių pagrindinis – nutolę duomenys, kurių fizinė buvimo vieta nėra tiksliai žinoma. Tiriant tokiomis aplinkybėmis, yra didelė tikimybė prarasti svarbius nusikaltimo artefaktus. Pavyzdžiui, gali nebūti galimybių pasiekti skaičiavimų debesies duomenų centruose esančius registro įrašus, laikinuosius failus ir laikinąją atmintį. Jeigu duomenys yra atsisiunčiami iš skaičiavimų debesies, gali būti prarasta ir metainformacija.

Yra ir daugybė kitų problemų, susijusių su teismo ekspertizės atlikimu skaičiavimų debesyje. Tai yra vis dar besivystanti sritis, ir aktyviai kuriami įvairūs metodai jai patobulinti. Šių aplinkybių pagrindu ir buvo rašomas magistrinis darbas.

Pagrindinis darbo tikslas – pasiūlyti naują metodiką skaitmeninės teismo ekspertizės skaičiavimų debesyje atlikimo efektyvumui pagerinti. Kad jį pasiekti, buvo įgyvendinti šie uždaviniai:

1. Išanalizuoti kitų autorių siūlomi skaitmeninės teismo ekspertizės atlikimo skaičiavimų debesyje problemos sprendimo būdai;
2. Pasiūlyta nauja metodika-įrankis, patobulinantis ekspertizės atlikimą;
3. Atliktas metodikos efektyvumą įvertinantis tyrimas.

Bendrai sutinkama, kad jeigu skaičiavimų debesies paslauga nefiksuoja tinkamų audito įrašų, nustatyti įkalčius tampa sunku arba tiesiog neįmanoma [14]. Deja, paprastai šiam tikslui siūlomas funkcionalumas yra gana ribotas arba iš viso neegzistuoja. Šios problemos sprendimui tyrėjai skiria daugiausiai dėmesio.

Siūloma pereiti nuo apsaugos, kuri veikia iš sistemos perspektyvos, prie efektyvesnės, kuri veikia atskaitos tašku pasirinkdama failą. Tai reiškia, kad duomenys, saugomi skaičiavimų debesyje, turi būti patys save apibūdinantys, kad būtų galima atsekti visą veiklą nuo jų sukūrimo iki sunaikinimo, nepriklausomai nuo aplinkos apribojimų [8, 18]. Sunkumų kelia tai, kad nėra oficialaus standarto, kokiu formatu įrašai turėtų būti fiksuojami siekiant vėliau efektyviai ištirti nusikaltimą skaičiavimų debesyje, todėl [11] siūlo priimti bendras gaires, kuriose apibrėžiama konkreti sintaksė. Taip pat skaičiavimų debesies audito įrašai negalės būti panaudoti teismo procese, jeigu nebus užtikrintas jų integralumas, todėl ši problema yra sprendžiama siūlymais įtraukti trečiąją šalį, kuri, apsikeisdama specialiomis žinutėmis su vartotoju ir skaičiavimų debesimi, papildomai išsaugotų ir įrašų privatumą [9, 17, 19].

Šaltinis [5] jau siūlo konkretų praktinį metodą audito įrašų kūrimui virtualioje aplinkoje veikiančioms duomenų saugykloms. Duomenų saugyklos kaip objektas buvo pasirinktos ir šiame magistriniame darbe.

Kad pasiūlyti ir ištirti naują metodą efektyvesniam skaitmeninės teismo ekspertizės atlikimui skaičiavimų debesies saugyklose, „VirtualBox“ programinės įrangos pagalba buvo sukurta virtuali aplinka, kurioje veikė 6-ios virtualios mašinos (VM) su „Linux“ operacinėmis sistemomis. Iš šios aplinkos išgautos „aktyvios“ VM kopijos buvo apdorojamos nauju siūlomu žurnalizavimo paramos sistemos (ŽPS) įrankiu, sukurtu „Python“ programavimo kalba apjungiant skaitmeninei teismo ekspertizei skirtas atvirojo kodo priemones „The Sleuth Kit“ ir „The Volatility Framework“. Papildomai sukurtais scenarijais buvo imituojami įvairūs vartotojų atliekami veiksmai (sukūrimas, redagavimas ir ištrynimas) su virtualios duomenų saugyklos failų sistemos objektais (failais ir katalogais). Galiausiai

buvo tikrinama, kiek šių faktiškų veiksmų pavyko atkurti su ŽPS ir kiek papildomų kompiuterijos išteklių reikalauja ši siūloma audito paramos sistema.

Viso magistrinio darbo struktūra yra tokia:

- 1-ame skyriuje apžvelgiamos skaičiavimų debesies technologijos bei skaitmeninės teismo ekspertizės sąvokos, šių sferų aktualumas ir tarpusavio ryšys; apibrėžiama skaitmeninės teismos ekspertizės atlikimo skaičiavimų debesyje problema, ir išanalizuojami kitų autorių siūlomi būdai jai spręsti;
- 2-ame skyriuje pristatoma naujai siūlomos metodikos-įrankio, žurnalizavimo paramos sistemos, padidinančios skaitmeninės teismo ekspertizės atlikimo skaičiavimų debesies saugyklose efektyvumą apjungiant kelias atvirojo kodo tam tikslui skirtas programines priemones: „The Sleuth Kit“ ir „The Volatility Framework“, koncepcija; pristatomos šių programinių priemonių galimybės ir funkcionalumas, apibrėžiama jų vieta ŽPS;
- 3-iame skyriuje dokumentuojamas žurnalizavimo paramos sistemos efektyvumo tyrimas: aprašoma eksperimentinė realizacija, tyrime naudotos techninės ir programinės priemonės, supažindinama su tyrimo atlikimo metodika, pateikiami rezultatai;
- Pabaigoje pateikiamos pagrindinės darbo išvados, išreiškiami pasiūlymai tolimesniems šią problemą nagrinėjantiems tiriamiesiems darbams.

Magistrinio darbo rezultatai buvo pristatyti 18-toje tarpuniversitetinėje magistrantų ir doktorantų konferencijoje „Informacinė visuomenė ir universitetinės studijos (IVUS 2013)“, kurios iniciatyva taip pat buvo išpublikuotas šio magistrinio darbo pasekoje parašytas autorių straipsnis: „Nusikalstamos veiklos „skaičiavimų debesies“ saugyklose atkūrimo metodas“. Jo kopija pateikta A priede.

1. VEIKSMŲ KAIP ĮKALČIŲ ATKŪRIMO SKAIČIAVIMŲ DEBESIES SAUGYKLOSE PROBLEMOS ANALIZĖ

1.1.Skaičiavimų debesies technologija

Paskutiniais keliais metais IT sferoje įvyko reikšmingų pasikeitimų, vienas kurių - skaičiavimų debesies sukūrimas. Šios technologijos atsiradimas ėmė keisti būdus, kaip organizacijos realizuoja savo kompiuterijos poreikį. Patikimi tiekėjai, tokie kaip „Microsoft“, „Amazon“, „Google“, „Yahoo“ ir kiti, pasiūlė kompanijoms efektyvias galimybes reikiamus IT resursus pirkti iš trečiųjų šalių ir tokiu būdu pagrindinį dėmesį koncentruoti į savo esminį verslą [12].

Skaičiavimų debesis - tai „modelis, suteikiantis visur pasiekiamą, patogią, „pagal pareikalavimą“ tinklo prieigą prie bendrų, konfigūruojamų kompiuterių resursų (tinklų, serverių, saugyklų, vartotojo programinės įrangos, paslaugų ir kt.), kurie gali būti sparčiai keičiami su minimaliomis valdymo pastangomis ar paslaugos tiekėjo įsikišimu“ [24] (1.1 pav.).

1.1 pav. Skaičiavimų debesies modelis [20]

Skaičiavimų debesies modelį apibūdina 5-ios esminės charakteristikos:

1. Savitarnos paslaugos “pagal pareikalavimą“;
2. „Plati“ tinklo prieiga;

3. Resursų dalinimasis su kitais vartotojais iš bendro fondo;
4. Spartus naudojamų paslaugų mąsto keitimas (elastiškumas);
5. Išmatuojamumas.

Kompanijos, užsisakydamos skaičiavimų debesies paslaugas, gali rinktis iš 3-jų paslaugos modelių:

A. „Programinė įranga kaip paslauga“ (angl. *Software as a Service* – SaaS): klientas naudojasi programine įranga, kuri veikia tiekėjo debesies infrastruktūroje;

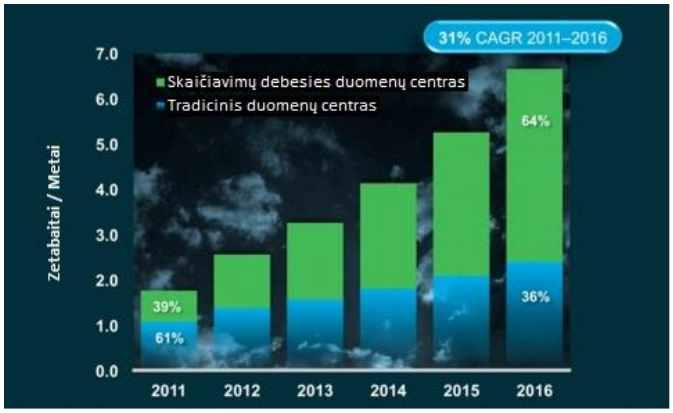
B. „Platforma kaip paslauga“ (angl. *Platform as a Service* – PaaS): klientas gali pats įsidiegti reikiamą programinę įrangą debesyje, kuri sukurta naudojantis tokiomis programavimo kalbomis, bibliotekomis, paslaugomis ir įrankiais, kuriuos palaiko paslaugos tiekėjas;

C. „Infrastruktūra kaip paslauga“ (angl. *Platform as a Service* – PaaS): suteikiamos galimybės patiems klientams 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.

Galiausiai yra galimi 4-ri skaičiavimų debesies diegimo modeliai:

1. Privatus, kai debesies infrastruktūra naudojama išskirtinai tik vienintelės organizacijos;
2. Bendruomeninis, kai paslaugas naudoja specifinė kelių klientų grupė;
3. Viešas, kai debesis yra prieinamas bendrajai publikai;
4. Hibridinis, kai apjungiami keli paminėti diegimo modeliai [24].

Skaičiavimų debesies paslauga sparčiai populiarėja. Prognozuojama, kad metinis globalus jos IP srautas išaugs 3,5 karto nuo 1,2 zetabaitų 2012 m. iki 4,3 zetabaitų (tai sudaro maždaug 4 600 000 000 terabaitų) 2016 m. ir sudarys du trečdalius visų duomenų centrų srauto (1.2 pav.) [22].



1.2 pav. Skaičiavimų debesies globalaus IP srauto prognozė

Tokias optimistines prognozes didžiąja dalimi lemia paslaugos teikiami privalumai. Pirmiausia ir svarbiausia, dramatiškai sumažėja IT resursų kaštai, todėl net ir mažos organizacijos turi galimybes pasinaudoti technologijomis, kurios anksčiau buvo prieinamos tik didelėms kompanijoms. Skaičiavimų debesis suteikia galimybes iškart pasinaudoti plačiais kompiuterijos resursais be išankstinių didelių investicijų. Kompanijos gali lanksčiai ir sparčiai keisti naudojamų paslaugų mąstą priklausomai nuo tą akimirką esamų klientų poreikių. Galiausiai, tai suteikia pagrindą inovacijoms, atsiranda galimybės sukurti tokias paslaugas, kurios anksčiau buvo neįmanomos [10].

Nežiūrint į teigiamas puses, skaičiavimų debesis kelia įvairių abejonių ir susirūpinimų. Ypač išreiškiama baimė dėl esamų galimybių paslaugos tiekėjui būti užpultam elektroninės erdvės nusikaltėlių, kurie gali pavogti klientų informaciją. Galima tiek tiesioginė, tiek netiesioginė (pasinaudojant kliento puse) debesies infrastruktūros kompromitacija, egzistuoja įvairūs debesies pažeidžiamumai, kyla paslaugos tiekėjo šnipinėjimo grėsmė, be to, dėl nutolusių resursų, sunkiai sprendžiamas audito ir skaitmeninės teismo ekspertizės atlikimo klausimas ir kt. [2].

Galimos apsaugos nuo pagrindinių pavojų priemonės sąlygiškai skirstomos į pasyvias ir aktyvias. Pasyvios apsaugos priemonės (šifravimas, prieigos valdymas ir kt.) padeda sumažinti iškilusią konfidencialios informacijos atskleidimo riziką, tačiau vien tik to neužtenka. Yra didelis poreikis esamą apsaugą papildyti tokiais būdais, kurie skatina skaidrumą ir atskaitomybę prieš klientus ir valstybines institucijas. Pavyzdžiui, nors audito įrašų darymas yra kritinis pasitikėjimo komponentas, dabartinės populiarios skaičiavimų debesies paslaugos („Amazon EC2/S3“, „Microsoft Azure“) neturi savyje pilnavertės galimybės sekti ir tikrinti priegų prie failų istoriją. Geriausiu atveju dabar klientai gali stebėti tik esminius pagrindinės sistemos darbo „log įrašus“ ir paslaugos darbo efektyvumo metrikas.

Aktyvios apsaugos priemonės naudojamos identifikuoti privatumo ar saugumo rizikas, pavyzdžiui, įsilaužimo stebėsenos sistemos (angl. *intrusion detection systems*), audito įrašai ir jų analizės priemonės. Be visa ko, jos padeda aptikti ir anomalijas, kylančias iš debesies tiekėjo vidaus. Paprastai reikalingos tiek aktyvios, tiek pasyvios priemonės, kad būtų užtikrinta racionali apsauga [8].

1.2.Skaitmeninės teismo ekspertizės sfera

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

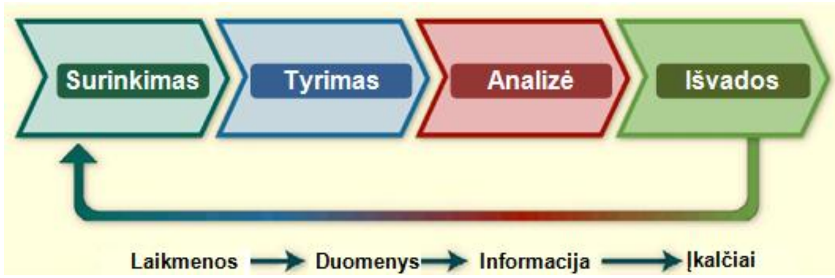
Skaitmeninės teismo ekspertizė turi daug apibrėžimų, bet plačiąja prasme tai yra „mokslo panaudojimas siekiant identifikuoti, surinkti, ištirti ir išanalizuoti duomenis išlaikant jų vientisumą ir išsaugant griežtą „arešto grandinę“ [25] (angl. *chain of custody*).

Per pastarąjį dešimtmetį stipriai išaugo nusikaltimų, atliekamų elektroninėje erdvėje, skaičius. To pasekoje, įsikūrė daug kompanijų, kurios siūlo produktų, padedančius teisėsaugos organams nustatyti kas, ką, kur, kada ir kaip padarė nusikaltimus naudojantis kompiuterinėmis priemonėmis. Skaitmeninė teismo ekspertizė, kaip sfera, būtent ir išsivystė tuo pagrindu, kad būtų užtikrintas tinkamas elektroninių įkalčių pristatymas teismui.

Pats skaitmeninės teismo ekspertizės procesas yra sudarytas iš 4-ių pagrindinių fazių:

1) Surinkimas (angl. *collection*). Pirmiausia iš potencialių šaltinių turi būti nustatyti, užvadinti, surinkti ir išsaugoti visi nusikaltimo įkalčiai. Tai daroma vadovaujamasis 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;
2) 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;
3) 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;
4) 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 pačio teismo ekspertizės proceso patobulinimui: gairių, įrankių, politikos, procedūrų. Paprastai esama situacija apsprendžia, kokio formalumo ataskaitos yra reikalingos kiekvienu atveju atskirai.

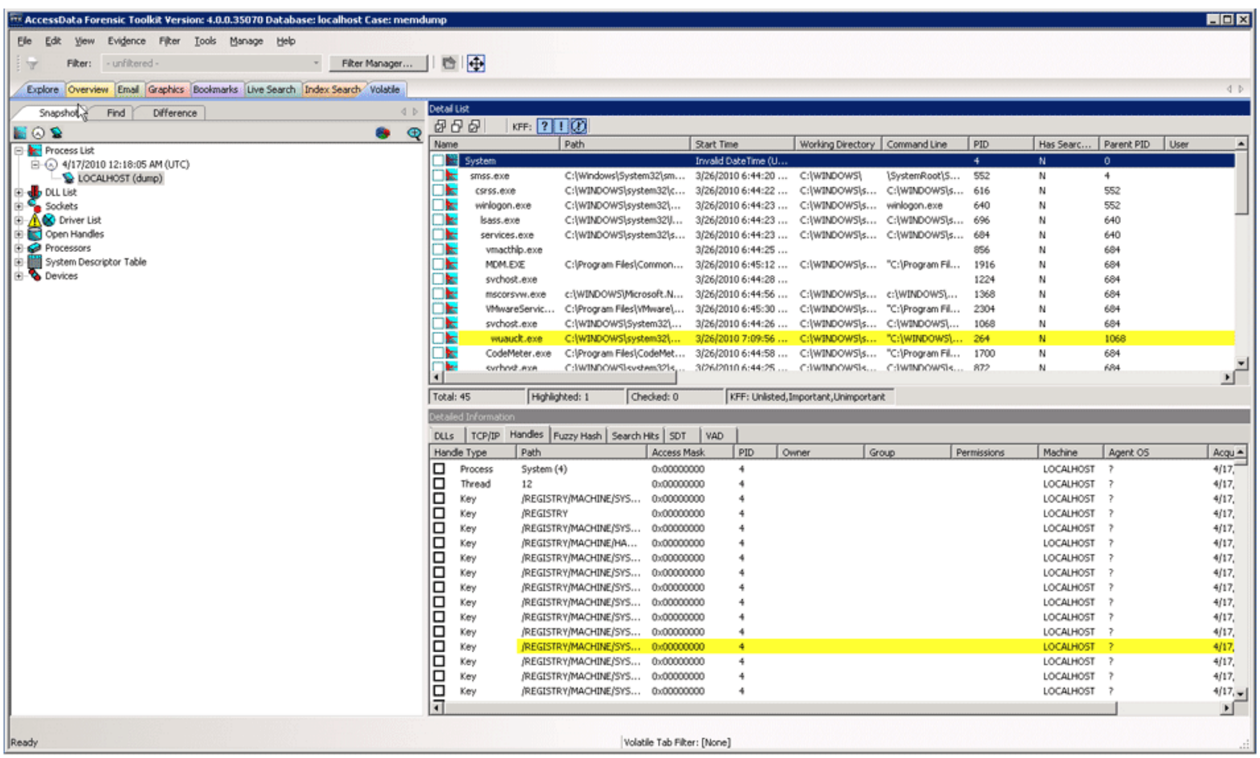
1.3 paveiksle šios fazės yra pavaizduotos grafiškai.



1.3 pav. Skaitmeninės teismo ekspertizės procesas

Kaip matome, teismo ekspertizės procesas informacijos saugojimo priemonę (angl. *media*) transformuoja į įkalčius (angl. *evidence*), pereinant duomenų (angl. *data*) ir informacijos (angl. *information*) tarpines grandis [25].

Vienos populiariausių skaitmeninės teismo ekspertizės priemonių yra komercinė programinė įranga „EnCase“ [23], FTK (1.4 pav.) [21], taip pat atviro kodo “The Sleuth Kit” [26].



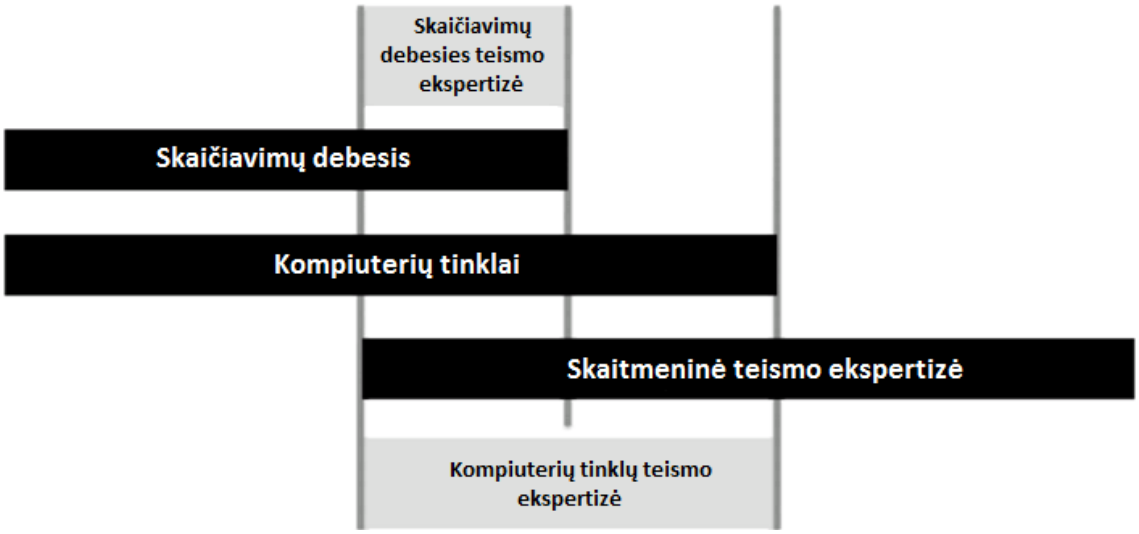
1.4 pav. Skaitmeninės teismo ekspertizės įrankis FTK

1.3.Skaitmeninė teismo ekspertizė skaičiavimų debesies kontekste

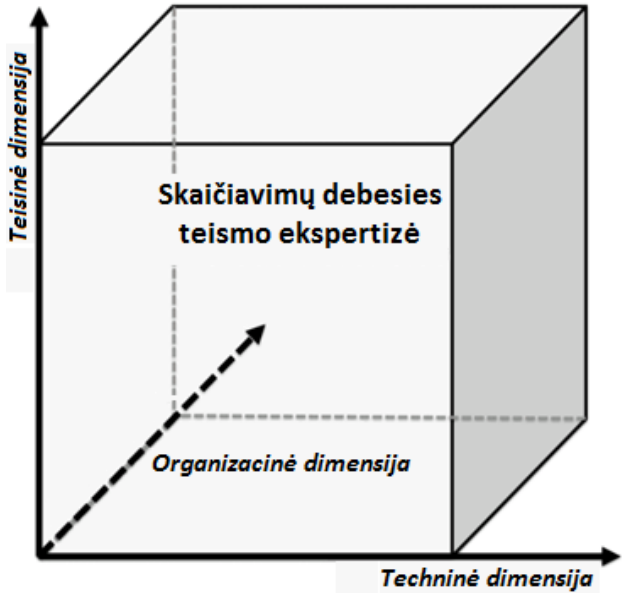
Kyla daug klausimų, ar skaičiavimų debesis gali būti tinkamai ištirtas skaitmeninės teismo ekspertizės tikslams. Viena vertus, šios technologijos tinkamumas tokios užduotims nėra pakankamai

išanalizuotas. Kita vertus, toks gebėjimas nėra esminis jos reikalavimas – dėl tarptautinės teisinės bazės šiuo klausimu nebuvimo, tai skaitoma daugiau kaip prabanga, kurią gali arba negali būti įsidiegę paslaugos tiekėjai savo nuožiūra [14].

Skaitmeninė teismo ekspertizė skaičiavimų debesyje (angl. *cloud forensics*) apjungia 3-is siauresnes sritis: skaitmeninę teismo ekspertizę, skaičiavimų debesį ir kompiuterių tinklus (1.5 pav.). Ją vertinant, reikia nepamiršti tarpusavyje susijusių teisinių, organizacinių ir techninių aspektų (1.6 pav.) [13].



1.5 pav. Skaitmeninės teismo ekspertizės skaičiavimų debesyje modelis



1.6 pav. Skaitmeninės teismo ekspertizės skaičiavimų debesyje aspektai

Kalbant apie techninius skaitmeninės teismo ekspertizės atlikimo skaičiavimų debesyje aspektus, yra galimi 3-s įkalčių šaltiniai:

1. Debesies dalis (taip vadinama virtuali instancija), kurią naudoja klientas. Joje yra saugomi duomenys ir atliekami procesai. Paprastai tai yra ta vieta, kur atsitinka incidentai, todėl jų tyrimai natūraliai pradedami būtent nuo čia. IaaS paslaugos modelio atveju, yra galimybės „gyvai“ tirti vis dar veikiančias virtualias instancijas (pavyzdžiui, „Amazon EC2” siūloma technologija „XenAccess“ [7]) arba padaryti jų kopijas tam tikru laiko momentu (angl. *snapshot*) ir išanalizuoti jas atskirai [3, 4]. SaaS ir PaaS paslaugų modelių atvejais tai galima padaryti tik susiduriant su dideliais apribojimais (arba išvis neįmanoma).

2. Kompiuterių tinklas. Skirtingi tinklo OSI „sluoksniai“ galėtų suteikti įvairios naudingos informacijos apie juose veikiančius protokolus, kurie užtikrina duomenų srautų tarp „skaičiavimų debesies“ ir kliento apsikeitimą. Deja, tipiškas debesies paslaugų tiekėjas nesuteikia jokių tinklo komponentų audito įrašų, todėl, pavyzdžiui, kenksmingos programinės įrangos užsikrėtimo atveju gauti informacijos iš susijusių maršrutizatorių nebūtų įmanoma. Iš esmės tai reiškia, kad įkalčių kiekis priklauso tik nuo debesies dalies ir kliento sistemos.

3. Kliento sistema. Paprastai tai yra tik naršyklė, veikianti kliento kompiuteryje, kurios pagalba yra pasiekiamos visos reikiamos skaičiavimų debesies paslaugos, ypač SaaS paslaugos modelio atvejais. Tai reiškia, kad potencialus įkalčių šaltinis gali būti naudojamos naršyklės aplinka – to nederėtų atmesti.

Akivaizdu, kad potencialiai galimų įkalčių skaičius stipriai priklauso nuo to, koks skaičiavimų debesies paslaugos modelis yra naudojami kiekvienu nagrinėjamu atveju. Tai plačiau paaiškinta 1.1 lentelėje [1].

1.1 lentelė. Skaičiavimų debesies paslaugos modelių įtaka galimų įkalčių kiekiui

SaaS	➔ Klientas nekontroliuoja jokios pagrindinės infrastruktūros (tinklų, serverių, operacinių sistemų ir t.t.), netgi tos programinės įrangos, kurią naudoja, todėl nėra galimybių gauti detalesnį sistemos vaizdą. ➔ Galimi tik paprastų vartotojų naudojimosi parametrų pakeitimai. To pasekoje, nusikaltimo tyrėjas gali pasikliauti tik debesies tiekėjo suteikiamais bendriniais audito įrašais. ➔ Jeigu audito įrašų kūrimo sistema nėra naudojama, klientas neturi jokių galimybių pats susikurti įkalčių surinkimui tinkamą aplinką – papildomų audito priemonių diegimas yra apribotas.

PaaS	➔ Klientas kontroliuoja pagrindinę programinę įrangą. ➔ Jis gali susikurti ir įsidiegti sau pritaikytas audito sistemas, kurios lanksčiai komunikuotų su aktualia infrastruktūra (duomenų bazėmis, saugyklomis ir pan.). ➔ Sukurti audito įrašai gali būti šifruojami ir siunčiami trečiosioms šalims.
IaaS	➔ Tai pati „draugiškiausia“ aplinka įkalčių surinkimui, nes kadangi klientas kontroliuoja visą infrastruktūrą, jis gali sau prisitaikyti bet kokią audito sistemą, netgi veikiančią atskirose virtualiose instancijose, tokiu būdu garantuojant didesnį jų patikimumą.

Vertinant iš skaitmeninės teismo ekspertizės pusės, pagrindinis skaičiavimų debesies privalumas yra centralizuoti duomenys. Tai palengvina darbą nusikaltimų tyrėjams, nes vienoje vietoje esančią informaciją lengviau sukontroliuoti. Pasinaudodami centralizuotais duomenimis, IaaS tiekėjai netgi gali debesyje sukurti dedikuotą teismo ekspertizės serverį, kurį būtų galima panaudoti atsiradus poreikiui.

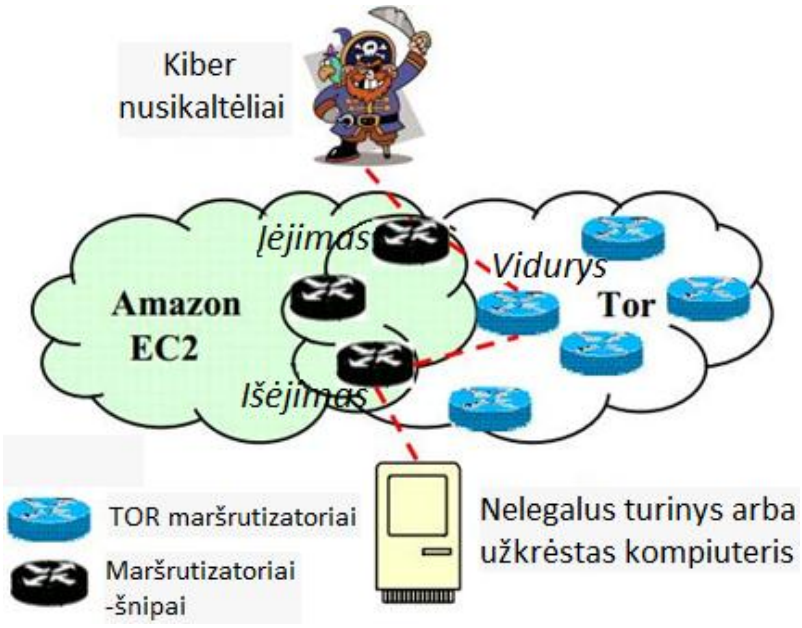
Kitas teigiamas dalykas – milžiniški kompiuterijos resursai, kuriuos galima panaudoti teismo ekspertizės tikslams. Saugyklos, kurių dydis gali siekti petabaitus, ir neišsenkami procesorių skaičiavimo resursai nusikaltimo tyrėjams suteikia anksčiau neturėtas galimybes. Jie gali saugoti neribotą tiriamos informacijos kiekį (tyrimo metu daromos kietųjų diskų kopijos užima daug vietos) ir daug sparčiau atlikti sudėtingų skaičiavimų reikalaujančius veiksmus, pavyzdžiui, slaptažodžių nulaužimą.

Skaičiavimų debesies tiekėjai papildomai suteikia integruotas maišos (angl. *hash*) funkcijas ir tokiu būdu padeda nusikaltimo tyrėjams, kurie šiuos laikui imlius veiksmus privalo atlikti visada, kai tik yra daromos laikmenų (saugyklų) kopijos. Pavyzdžiui, „Amazon S3” skaičiavimų debesies paslauga sugeneruoja MD5 maišos reikšmę iškart, kai tik yra išsaugomas objektas.

Teismo ekspertizės tyrimams daug naudingos informacijos gali suteikti įvairūs audito įrašai. Nežiūrint į tai, paprastose kompiuterinėse sistemose dėl vietos diske trūkumo tokių įrašų kūrimas paprastai yra apribotas arba išjungtas. Skaičiavimų debesies mąstas suteikia neribotas galimybes audito įrašų kūrimo mechanizmus sukurti ir suderinti taip, kaip to pageidauja esama kliento situacija, kad ir kokia kompleksiška ir imli resursams ji bebūtų [12].

Yra netgi tyrimų, kurie pademonstruoja, kaip skaičiavimų debesies paslauga pati savaime gali tapti nusikaltimo tyrimo įrankiu. Pavyzdžiui, dabartiniais laikais, kai elektroninės erdvės nusikaltėliai

neretai naudojasi anonimiškumą užtikrinančio „Tor“ tinklo paslaugomis, teisėsaugos organai gali įsigyti dešimtis virtualių „Amazon EC2” skaičiavimų debesies instancijų ir paversti jas „Tor“ tinklo tarpiniais mazgais, šnipinėjančiais pro juos keliaujančius informacijos srautus ir tokiu būdu, esant poreikiui, identifikuojančiais nusikaltėlius (1.7 pav.) [5].



1.7 pav. Skaičiavimų debesies kaip nusikaltimo tyrimo įrankio pavyzdys

Visgi skaičiavimų debesis, vertinant ją iš skaitmeninės teismo ekspertizės pozicijų, turi ir esminių trūkumų, kurių pagrindinis – nutolę duomenys, kurių fizinė buvimo vieta nėra tiksliai žinoma. Standartiniame skaitmeninės teismo ekspertizės procese kompiuterinė technika pirmiausia yra areštuojama ir tada tiriama. Tas nėra praktiška skaičiavimų debesies atveju: jį palaikantys duomenų centrai yra išdėstyti visame pasaulyje, ir vieno kliento duomenys gali būti paskirstyti per keletą jų (be to, jie neretai yra užšifruoti [3]). Esant tokioms aplinkybėms, išlaikyti „arešto grandinę“ yra labai sunku, dažnai net neįmanoma. Tai apsunkina nusikaltimo scenos atkūrimą, kadangi nutolusius duomenis yra sudėtinga susieti tarpusavyje į logišką, susijusių įvykių grandinę [6].

Tiriant tokiomis aplinkybėmis, yra didelė tikimybė prarasti svarbius nusikaltimo artefaktus. Pavyzdžiui, gali nebūti galimybių pasiekti skaičiavimų debesies duomenų centruose esančius registro įrašus (angl. *registry entries*), laikinus failus (angl. *temporary files*) ir laikinąją atmintį (angl. *memory*). Jeigu duomenys yra parsisiunčiami iš skaičiavimų debesies, gali būti prarasta ir meta informacija (angl. *metadata*). Meta informacija, tokia kaip failo sukūrimo, modifikavimo ir nuskaitymo datos, teismo ekspertui gali būti naudingas įkalčių šaltinis.

Dar vienas trūkumas yra įrankių, skirtų skaitmeninės teismo ekspertizės atlikimui skaičiavimų debesyje, nebuvimas. Nors kompiuterių teismo ekspertizė yra pakankamai nauja sfera, ji išsivystė iki tokio lygio, kad yra sukurta pakankamai priemonių standartinių nusikaltimų tyrimui. Įrankiai, tokie kaip programinė įranga „EnCase“, „Helix“ ir FTK, gali būti naudojami įvairioms užduotims, pradedant pirminiu duomenų surinkimu ir baigiant specialių ataskaitų, tinkamų pristatyti teisme, paruošimu.

Egzistuoja žmogiška problema, susijusi su skaitmeninės teismo ekspertizės atlikimu skaičiavimų debesyje. Visi įkalčiai turi būti pristatyti ir išaiškinti teisėjui naudojant techninį žargoną, kuris gali būti per sudėtingas apibūdinant vien lokalias kompiuterines sistemas, jau nekalbant apie skaičiavimų debesies duomenų centrus, nutolusius per tūkstančius kilometrų, savyje saugančius, pavyzdžiui, 512 serverių su 40 000 virtualių instancijų, kurias naudoja 1000 klientų, ir vienas iš jų yra įtariamasis. Vidutinis teisėjas paprastai turi tik tiek IT žinių, kiek reikia pasinaudoti namuose esančiu kompiuteriu [12].

Yra ir daugybė kitų problemų, susijusių su teismo ekspertizės atlikimu skaičiavimų debesyje, tokių kaip teisiniai aspektai (duomenų centrų išsidėstymas per kelias šalis kelia jurisdikcijos klausimą), įkalčių stabilumas (ar dinaminė aplinka bėgant laikui jų nepakeičia), virtualią instanciją palaikančios kompiuterių architektūros nustatymas (kad būtų galima ją areštuoti) ir kitos, kurias aktyviai siūloma spręsti mokslininkų bendruomenei [16].

Galima teigti, kad išvardinti trūkumai iš esmės susiformavo iš trijų pagrindinių aplinkybių:

- Ribotų skaičiavimų debesies architektūros žinių;
- Trūkstamų žinių apie konkrečių duomenų buvimo vietą;
- Įkalčių surinkimo tokioje aplinkoje problemų [7].

Akivaizdu, kad skaičiavimų debesies paslaugų tiekėjai dar aiškiai neišsprendė klausimo, kaip jų teikiamos paslaugos bus pritaikytos teismo ekspertizės atlikimui. Teismo ekspertai taip pat dar neturi aiškių procedūrų, kaip elgtis tyrimo debesyje atvejais. Skaitmeninė teismo ekspertizė skaičiavimų debesyje yra vis dar besivystanti sfera, ir yra aktyviai siūloma įvairių metodų jai patobulinti [12].

1.4.Audito įrašai kaip priemonė atkurti veiksmus

Jeigu skaičiavimų debesies paslauga nefiksuoja tinkamų audito įrašų, nustatyti įkalčius tampa sunku arba tiesiog neįmanoma [24]. Šiam klausimui yra skiriamas didžiausias dėmesys.

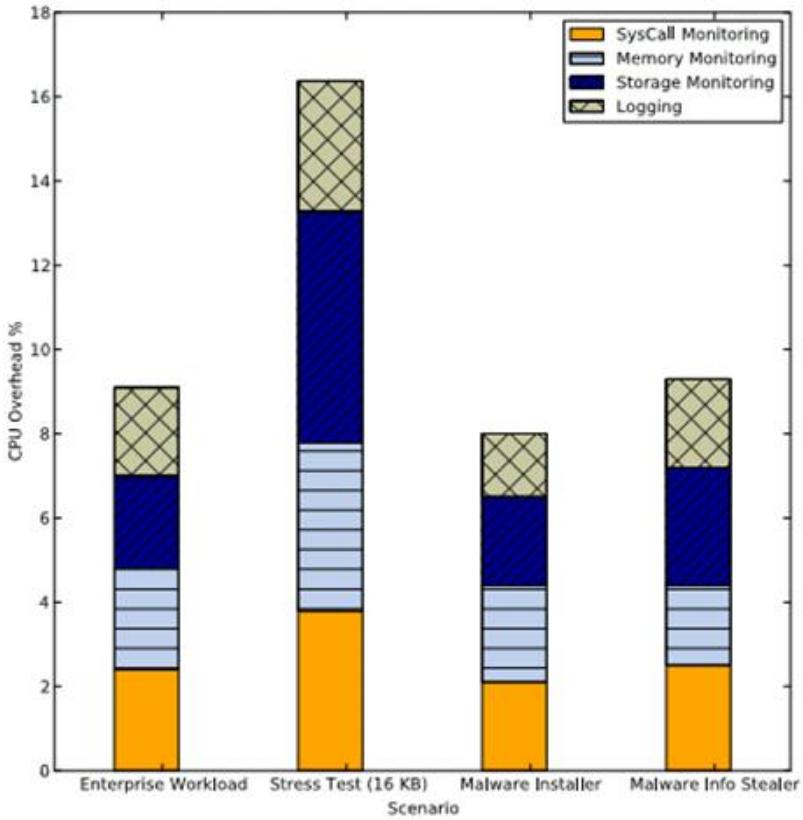
Yra siūloma pereiti nuo apsaugos, kuri veikia iš sistemos perspektyvos, prie efektyvesnės, kuri veikia atskaitos tašku pasirinkdama failą. Tai reiškia, kad duomenys, saugomi skaičiavimų debesyje, turi būti patys save apibūdinantys, kad būtų galima atsekti visą veiklą nuo jų sukūrimo iki sunaikinimo

nepriklausomai nuo aplinkos apribojimų [18]. Šaltinis [8] apibūdina pagrindinius iššūkius, susijusius su šios idėjos, dar vadinamos duomenų centriškumu (angl. *data-centric view*), įgyvendinimu ir siūlo galimą platformą (angl. *framework*) tokios informacijos apsikeitimui skaičiavimų debesyje tarp skirtingų sistemų, kuri padėtų išsaugoti reikiamus meta duomenis. Pavadinta „TrustCloud“, ji priskiriama aktyvių apsaugos priemonių kategorijai ir siūlo technikų rinkinį, padedantį spręsti skaičiavimų debesies saugumo, patikimumo ir atskaitingumo problemas. Šios platformos pasiūlymai aprėpia ir bendrąją politiką bei atskiras taisykles, susijusias su IT sistemomis.

Autoriai pabrėžia, kad dabartiniai skaičiavimų debesies apsaugos metodai daugiausia remiasi pasyviomis priemonėmis, pavyzdžiui, šifravimu, tačiau paskutiniu metu įvykę aukšto lygio incidentai parodė, kad to nepakanka. Jos nesuteikia vartotojams paslaugos skaidrumo ir atskaitomumo, be to, nesudaro galimybių atsekti su pačiais save apibūdinančiais duomenimis susijusius veiksmus, todėl aktyvus audito įrašų kūrimas yra būtinas, be to, turėtų tenkinti tokius reikalavimus:

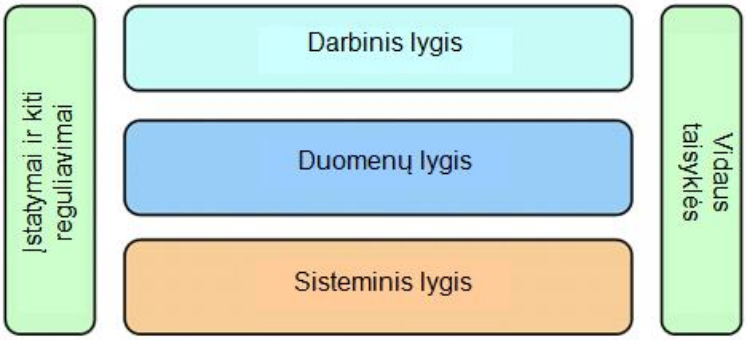
- ✓ Sekti failus. Aplinkoje, kurioje įgyvendintas duomenų centriškumas, failų audito įrašų kūrimas suteiktų gebėjimą fiksuoti visą failo gyvavimo ciklą. Stebimi sisteminiai failų sistemos skaitymo/rašymo kreipiniai (angl. *call*) padėtų atsekti ir susieti virtualias ir fizines failų atminties vietas, kas suteiktų papildomos informacijos tolimesniam tyrimui (įrodyta, kad dėl šios papildomos funkcijos virtualizacijos platformos procesoriaus apkrovimas padidėtų iki 20% (1.8 pav.) [5]). Failų sekimas turėtų būti organizuojamas ne tik vietiniu vieno mazgo, bet ir platesniu, tinklo, mąstu, kadangi skaičiavimų debesį sudaro daugybėje lokacijų išsidėstę fiziniai ir virtualūs serveriai. Tinklo audito įrašų sekimas padėtų išsaugoti failo judėjimą iš vieno vietos į kitą (nepriklausomai nuo to, fizinę ar virtualią) ir lokacijas, kuriomis jis išsiunčiamas už skaičiavimų debesies ribų;
- ✓ Sekti duomenis. Informacijos požiūriu, sekti vien tik failus neužtenka. Tam, kad būtų galima argumentuotai pateikti įrodymus apie duomenų sukurimą, modifikavimą arba ištrynimą, būtina fiksuoti, kas jais naudojasi. Kilmės informacija paprastai laikoma prideramų privatumo ir patikimumo modelių pagrindu, nes ji padeda patikrinti, kurie procesai dalyvavo sukuriant/naudojantis duomenimis, ir leidžia aptikti veiksmų anomalijas;
- ✓ Sekti informaciją. Kadangi būtent informacija, o ne duomenys ar failai yra svarbiausia organizacijos vertybė, padedanti priimti teisingus ir verslo sėkmę įtakojančius sprendimus, užfiksuoti audito įrašai, parodantys, kaip failai ir duomenys transformavosi į informaciją, yra neįkainojamos vertės tiriant saugumo incidentus;

- ✓ Sekti informaciją ir duomenų srautus. Aukšto lygio rizikos, tokios kaip kompiuterių programose cirkuliuojanti sprendimų priėmimo informacija, turėtų būti stebima bei kontroliuojama.



1.8 pav. Procesoriaus apkrovimo padidėjimas papildomai apdorojant sisteminius failų sistemos skaitymo/rašymo kreipinius skaitmeninės teismo ekspertizės tikslams

„TrustCloud“ platforma siūlo apsibrėžti naujus skaičiavimų debesies sluoksnius (angl. *layers*), vietoje paplitusių skirstymų, tokių kaip, pavyzdžiui, „Programinė įranga kaip paslauga“ (SaaS), „Platforma kaip paslauga“ (PaaS) ir „Infrastruktūra kaip paslauga“ (IaaS), kurie remiasi sistemos centrališkumu (angl. *system-centric*), o ne duomenų. „TrustCloud“ skirstymas tiksliai apjungia ir apibrėžia audito įrašų, padedančių suformuoti save apsirašančius duomenis, mąstą: įrašai varijuoja nuo sisteminių failinių iki duomenų srautų (1.9 pav.):



1.9 pav. „TrustCloud" koncepcija

1. Sistemis sluoksnis – atsakingas už failų sekimą skaičiavimų debesyje;
2. Duomenų sluoksnis – seka duomenų ir informacijos pasikeitimus;
3. Duomenų srautų sluoksnis – seka duomenų ir informacijos cirkuliavimą;
4. Įstatymų ir kt. reguliavimų sluoksnis – užtikrina, kad audito įrašai būtų kuriami tenkinant teisinius aktus;
5. Vidaus taisyklių lygis – užtikrina, kad audito įrašai būtų kuriami tenkinant vidines kompanijos taisykles ir audito reikalavimus.

Autoriai pažymi, kad platformoje pagrindinis dėmesys yra skiriamas bendriniams audito įrašų sluoksniams, nesusiejant jų su sistemos architektūra, todėl „TrustCloud" nėra priklausomais nuo konkrečių virtualių ar fizinių aplinkų [8].

Reikalus apsunkina tai, kad nėra oficialaus standarto, kokiu formatu įrašai turėtų būti fiksuojami siekiant vėliau efektyviai ištirti nusikaltimą skaičiavimų debesyje. Šaltinis [11] siūlo priimti bendras gaires, kuriose apibrėžiama konkreti sintaksė. Siūloma metodika, nusakanti, kada, kur ir kaip turi būti kuriami audito įrašai – tam, kad būtų sudarytos galimybės juos panaudoti trims pagrindiniams tikslams: skaitmeninei teismo ekspertizei, apskaitai ir koreliacijai. Teigiama, kad nesivadovaujant šiomis gairėmis tiksliai atkurti vartotojų veiksmus pareikalavus teismui yra neįmanoma, audito įrašų kūrimo taisyklės yra būtina kiekvieno teismo ekspertizės proceso sudedamoji dalis.

Siūlomos audito įrašų kūrimo gairės yra pritaikytos šiandieninei infrastruktūrai, dažnai veikiančiai skaičiavimų debesies pagrindu, kur vieno vartotojo veiksmus apdoroja įvairūs skirtingi komponentai ir asinchroninės operacijos. Jos yra sukurtos su tikslu optimizuoti tyrėjų, programinių priemonių kūrėjų ir vykdančiųjų komandų verslo procesus.

Išskiriami tokie iššūkiai, susiję su audito įrašų analize ir tyrimu:

- Audito įrašų decentralizacija;
- Jų pažeidžiamumas (angl. *volatility*);

- Daugybė programinių architektūrinių lygių ir sluoksnių;
- Archyvavimas ir archyvų saugojimas nustatytą laiko tarpą;
- Audito įrašų pasiekiamumas;
- Jų nebuvimas;
- Kritinės informacijos įrašuose stoka;
- Nesuderinami, atsitiktinai parinkti audito įrašų formatai.

Tam, kad išspręsti šias problemas, yra reikalingas audito įrašų valdymo sprendimas, siūlantis tokį funkcionalumą:

- Visų audito įrašų centralizavimas;
- Plečiamos įrašų saugyklos;
- Spartus duomenų pasiekimas ir nuskaitymas;
- Bet kokių įrašų formatų palaikymas;
- Galimybė paleisti duomenų analizės užduotis;
- Audito įrašų saugojimas nustatytą laiko tarpą;
- Senų įrašų archyvavimas ir atstatymas atsiradus poreikiui;
- Atskirtas duomenų pasiekimas naudojantis prieigos kontrolę;
- Įrašų integralumo užtikrinimas;
- Galimybė atsekti prieigos prie audito įrašų istoriją.

Išvardintas funkcionalumas sprendžia beveik visus anksčiau paminėtus iššūkius, išskyrus įrašų nebuvimo, kritinės informacijos stokos ir nesuderinamų formatų problemas. Kad įdiegti audito įrašų platformą, reikalingi 3-s žingsniai:

1. Aktyvuoti įrašų kūrimą kiekvienoje infrastruktūroje ir programinių priemonių komponentuose;
2. Nustatyti jų perdavimo būdus;
3. Optimizuoti įrašų kūrimo konfigūracijas.

Kada turi būti kuriami įrašai, priklauso nuo vartotojų poreikių, kurie skaičiavimų debesies aplinkoje skirstomi į:

- Aktualius verslui;
- Aktualius operaciniam lygmeniui;
- Aktualius saugumo tikslams (tyrimui);
- Reikalaujamus teisinių aktų.

Kaip taisyklė, turėtų būti fiksuojamas kiekvienas programinės įrangos atsakas (angl. *return call*) nepriklausomai nuo to, sėkmingas jis ar ne. Tokiu būdu yra išsaugomos klaidos ir gali būti atsekta su aplikacija susijusi veikla.

Mažų mažiausiai kiekviename audito įraše turi būti laiko, aplikacijos, vartotojo, sesijos numerio, svarbos (angl. *severity*), priežasties ir kategorijos laukai, nes jie padeda atsakyti į klausimus kada, kas, ką ir kodėl, be to, sudaro sąlygas gauti visą skirtingų poreikių vartotojus tenkinančią informaciją.

Kitas svarbus klausimas yra sintaksė, kurios pagrindas kilęs iš normalizacijos proceso – nustatymo, ką reiškia kiekvienas audito įrašo laukas. Autoriai, remdamiesi standartais ir tyrimais audito įrašų formalizavimo srityje siūlo juos kurti forma, pavaizduota 1.10 pav.

```
time=2010-05-13 13:03:47.123231PDT,
session_id=08BaswoAAQgAADVDG3IAAAAD,
severity=ERROR,user=pixlcloud_zrlram,
object=customer,action=delete,status=failure,
reason=does not exist
```

1.10 pav. Siūloma standartizuota audito įrašų sintaksė

Kiekvienas laukas yra atvaizduojamas kaip pavadinimo-reikšmės pora (angl. *key-value pair*) – tai pati svarbiausia įrašų savybė, sudaranti galimybes lengvai juos apdoroti ir nesunkiai interpretuoti. Viskas rašoma mažosiomis raidėmis, nenaudojant specialių ženklų (galima išimtis – pabraukimo simbolis „_“).

Kita ypatybė – trijų laukų, skirtų kategorizavimui, naudojimas: objekto, veiksmo ir būsenos (angl. *status*). Kiekvienam jų yra priskiriama po vienintelę reikšmę. Yra įvairių standartų, siūlančių skirtingus kategorizavimo būdas, bet galutiniame rezultate kompanija turėtų naudoti tokias reikšmes, kurios tiktų konkrečiai specifinei situacijai, atsižvelgiant į vartotojų poreikius. Toks skirstymas padeda sugrupuoti audito įrašus į naudingas logines grupes, pavyzdžiui, objekto reikšmė „klientas“ galėtų žymėti su klientais susijusius įrašus, o būsenos reikšmė „klaida“ atskirtų visas nesėkmingai pasibaigusias operacijas infrastruktūros mastu.

Autoriai palyginimui pateikia keleto programinių priemonių audito įrašų kūrimo funkcionalumą (1.2 lent.).

1.2 lentelė. Programinių priemonių siūlomo auditavimo funkcionalumo palyginimas

Programinė priemonė	Kas yra audituojama
„Django“	Objektų pasikeitimai, autentifikavimas ir autorizacija, išimtys,

	klaidos, ypatingų funkcijų panaudojimas (angl. *feature usage*)
„JavaScript“	AJAX kreipiniai, ypatingų funkcijų panaudojimas
„Apache“	Vartotojų užklausos (tiek sėkmingos, tiek ne)
„MySQL“	Veiksmai duomenų bazėje
Operacinė sistema	Sistemos būsena, operatyvinio lygmens klaidos

Kaip galimas siūlomų audito įrašų gairių įgyvendinimo praktikoje pavyzdys yra pateikiamas papildytas „Apache“ sistemos įrašas, kurį sudaro laiko žymė, užklausos šaltinis, URL adresas, aplikacijos sugeneruotas sesijos numeris ir HTTP kodas, padedantis nustatyti, kaip į užklausą sureagavo interneto svetainės serveris (1.11 pav.).

```
76.191.189.15 - - [29/Jan/2010:11:15:54 -0800]
  "GET / HTTP/1.1" 200 3874 "http://pixlcloud.service.ch/"
  "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_2; en-us)
  AppleWebKit/531.21.8 (KHTML, like Gecko) Version/4.0.4
  Safari/531.21.10" duvpqQqgOxAAABruAPYAAAAE
```

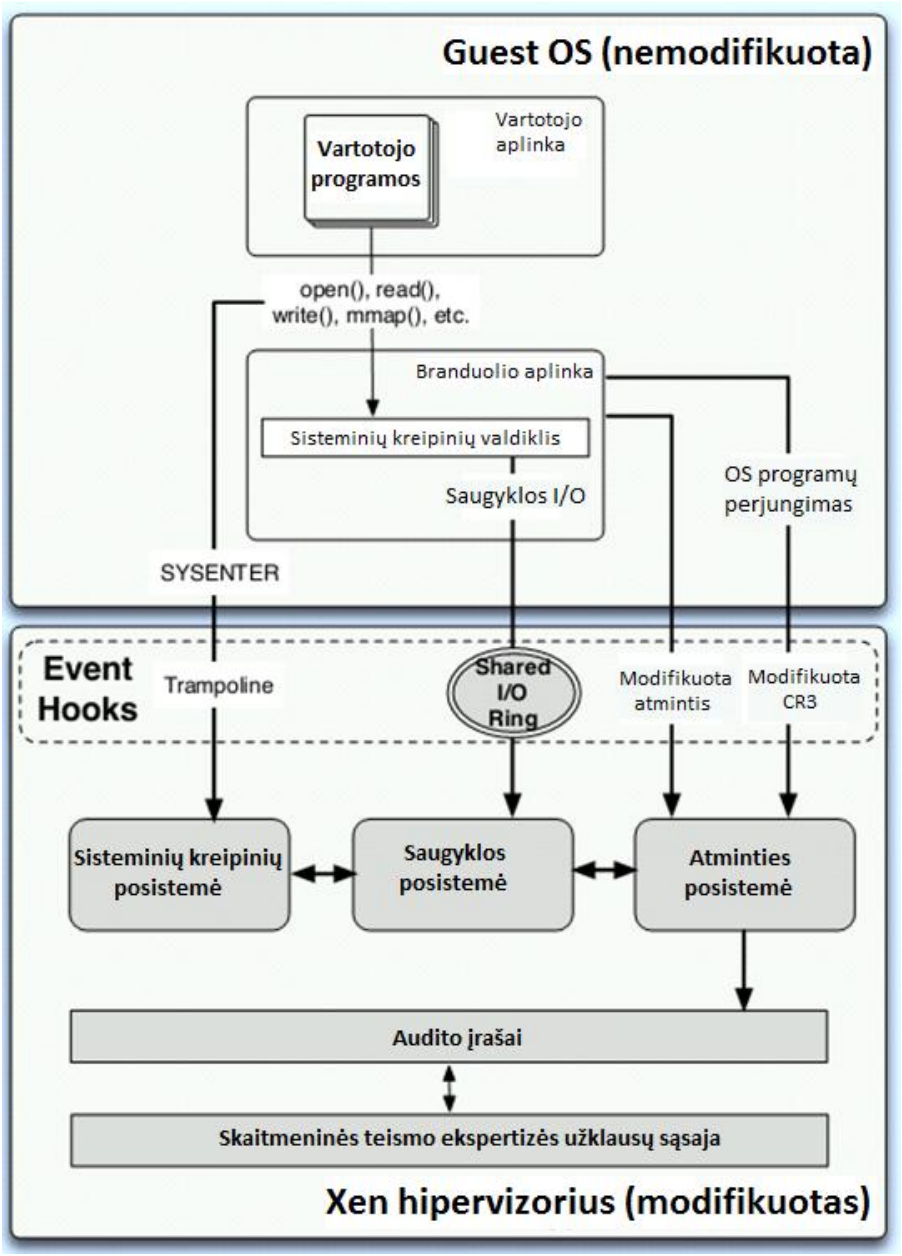
1.11 pav. Patobulinto „Apache“ audito įrašo pavyzdys

Šaltinis [5] siūlo konkretų praktinį metodą auditų įrašų kūrimui virtualioje aplinkoje veikiančioms duomenų saugykloms. Platforma skaidriai stebi ir fiksuoja prieigas prie duomenų pasinaudodama tik hipervizoriaus suteikiamomis abstrakcijomis. Šias prieigas ji susieja su veikiančiais procesais, tokiu būdu atsekdama veiksmų šaltinius, o atskirai veikiantis skaitmeninės teismo ekspertizės „sluoksnis“ išsaugo informaciją keleto versijų įrašuose. Tai sudaro galimybes vėliau efektyviai atkurti visus veiksmus ir jų sąlygotus pokyčius.

Viskas įgyvendinta „Xen“ virtualizacijos platformoje, kurios hipervizorių sudaro dvi pagrindinės dalys: privilegijuotas domenas ir virtualių mašinų monitorius (angl. *virtual machine monitor* - VMM). Privilegijuotas domenas „emuliacijos“ (angl. *emulation*) pagalba virtualioms mašinoms suteikia galimybę naudotis fiziniais įrenginiais, o VMM tvarko fizinius centrinio procesoriaus ir atminties resursus bei skirsto prieigas prie jų. Tokios architektūros dėka, siūloma platforma gali įgyvendinti savo funkcionalumą – pasinaudojant prieiga prie hipervizoriaus fiksuoti įvykius, kurie atsitinka virtualiose mašinose.

Ją sudaro trys pagrindiniai moduliai, kurie seka laikmenas, atmintį ir sisteminius kreipinius (1.12 pav.). Jie tampa pačio hipervizoriaus dalimi, ir virtualiose mašinose nereikia jokių papildomų diegimų. Sistema aktyvuojama jai nurodant sekti konkrečius virtualios mašinos laikmenų blokus (angl. *disk blocks*). Laikmenos modulis seka visus tiesioginius kreipinius į šiuos blokus ir su jais susijusius

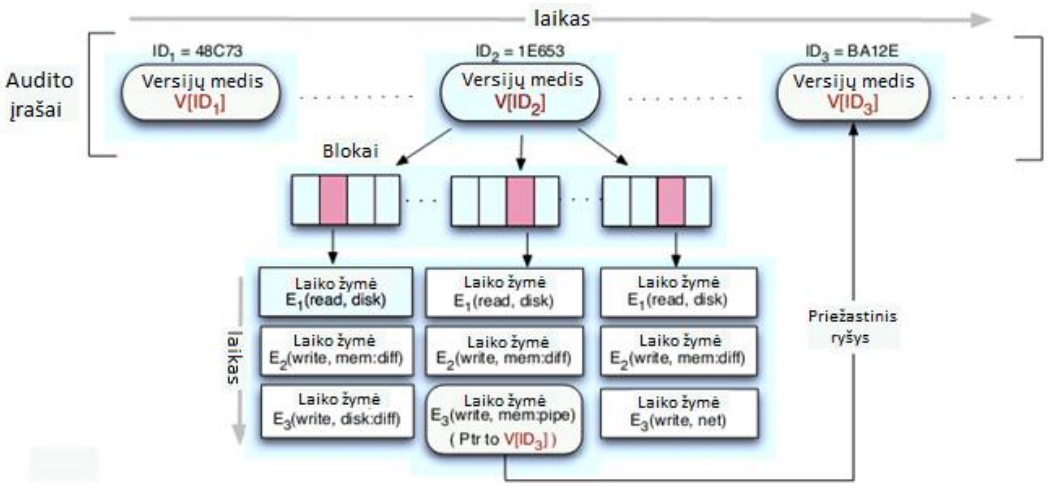
objektus, o pakartotiniai bandymai juos pasiekti yra užfiksuojami atminties ir sisteminių kreipinių modulių pagalba: sąveikaudami tarpusavyje jie sudaro galimybes stebėti kreipinius į objektus tada, kai šie jau būna patalpinti operatyvioje atmintyje. Atminties modulis taip pat atlieka veiksmų ir procesų susiejimo tarpusavyje funkciją.



1.12 pav. Koncepcinė siūlomos audito įrašų kūrimo platformos schema

Dėl tokios platformos architektūros atsiranda taip vadinama „semantinė spraga“ (angl. *semantic gap*), būdinga dirbant su panašaus lygio abstrakcijomis ir kurią reikia išspręsti: susieti blokus su failais, fizinius kompiuterių adresus su virtualiais ir instrukcijas su sisteminiais kreipiniais. Kadangi sistema

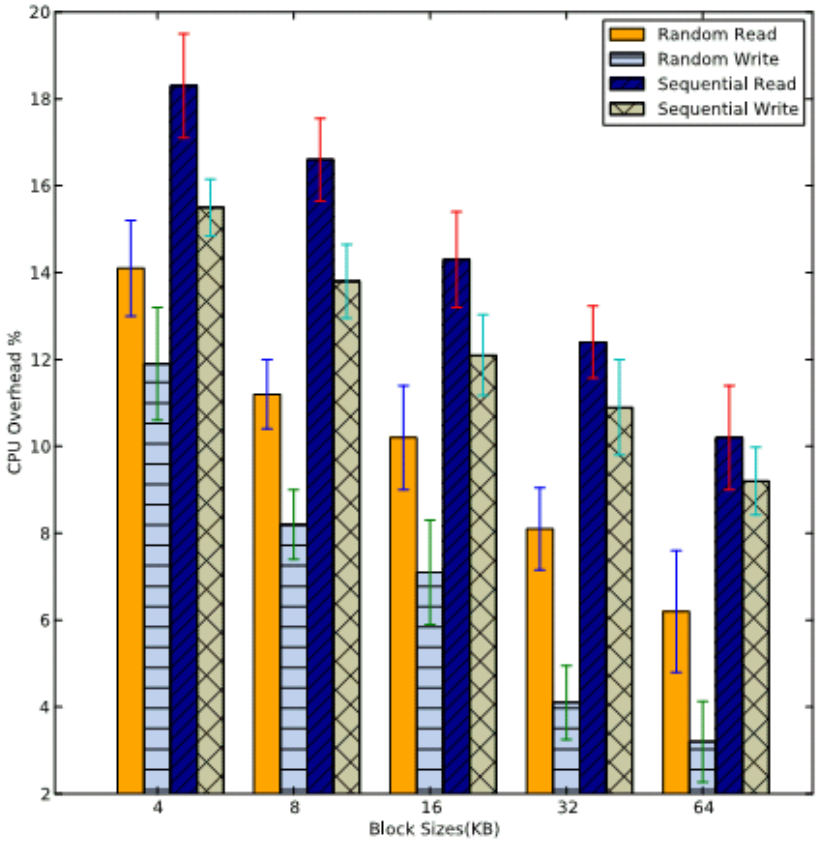
sukurta fiksuoti virtualioje mašinoje atliekamus veiksmus, kiekvienas jų gali sukelti daugybę hipervizoriaus įvykių, persidengiančių per įvairius abstrakcijos lygmenis, pavyzdžiui, pakartotinis tekstų redaktoriaus įrašinėjimas į failą reikalauja, kad disko objektai būtų vėl susieti su tuo pačiu failu, pakartotiniai rašymai į operatyvios atminties vietą taip pat turi būti vėl iš naujo susiejami su analogiška vieta fizinėje atmintyje ir t.t. Siekiant išsaugoti tokias sąsajas, moduliai sąveikauja tarpusavyje panaudodami šiuolaikiškas euristines priemones. Viskas saugoma versijomis pagrįstuose audito įrašuose, kuriuose laiko žymėmis yra aprašomos visos skaitymo ir rašymo sekos (1.13 pav.).



1.13 pav. Versijomis pagrįstų audito įrašų koncepcija

Akivaizdu, kad blokai patys savaime nesuteikia daug vertės, kol nėra sugrupuoti remiantis semantiniu požiūriu. Pagrindinis iššūkis, su kuriuo susiduria siūloma platforma, yra tas, kad kadangi įvykiai yra sekami blokų lygmenyje, failų sistemos objektai yra „nematomi“, todėl vėliau reikia tokius sąryšius atstatyti. Failų sistemos informacijos diske išdėstymui naudoja skirtingus būdus. Į tai įeina ir kaip failai, direktorijos ir kiti sisteminiai objektai yra susiejami su konkrečiais disko blokais, be to, tokios struktūros turi konkretų binarinį formatą ir saugomos konkrečiose nustatytose disko vietose. Autoriai daro prielaidą, kad virtualiose mašinose naudojamos failų sistemos yra žinomos (pavyzdžiui, ext3, NTFS ir pan.), ir laikmenos modulis periodiškai skanuoja diskus ieškodamas reikiamos meta informacijos, kuri vėliau būtų panaudojama skaitmeninės teismo ekspertizės tikslams.

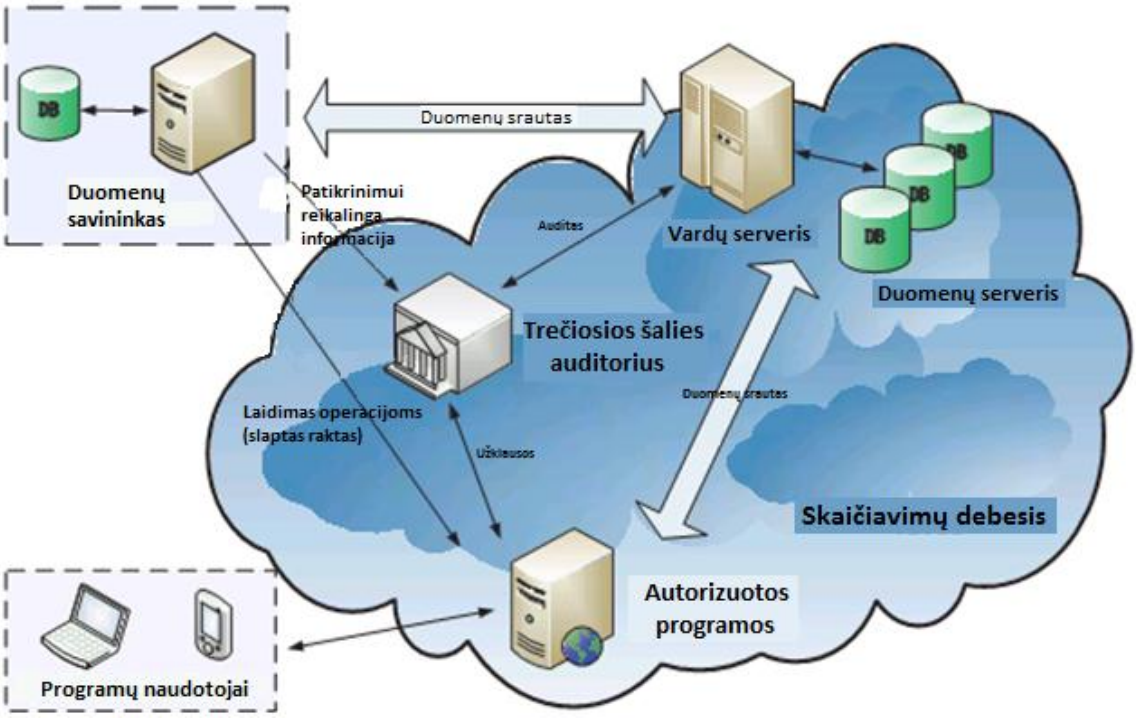
Kad pademonstruoti siūlomos platformos naudingumą, buvo atlikti išsamūs empiriniai bandymai, įskaitant ir realų gyvenimišką scenarijų, įrodantį, kaip buvo atkurta naudinga informacija, kas, kada ir kaip įsilaužė. Buvo pateikti ir virtualizacijos platformos, kurioje įdiegta platforma, papildomo procesoriaus apkrovimo vertinimai (1.14 pav.). Pabrėžiama, kad šios audito platformos patikimumas visiškai priklauso nuo to, kaip yra apsaugotas hipervizorius .



1.14 pav. Procesoriaus apkrovimo padidėjimas audito platformai sekant skirtingų dydžių disko blokus

Skaičiavimų debesies audito įrašai negalės būti panaudoti teismo procese, jeigu nebus užtikrintas jų integralumas. Ši problema, iškilusi skaičiavimų debesies saugyklų kontekste, yra sprendžiama siūlymais įtraukti trečiąją šalį, kuri, apsikeisdama specialiomis žinutėmis su vartotoju ir skaičiavimų debesimi, papildomai išsaugotų ir įrašų privatumą.

Šaltinio [19] siūlomos auditavimo sistemos su trečiąja šalimi architektūros schema pateikta 1.15 pav. Sistemą sudaro 4 dalys: duomenų savininkas, kuriam reikalinga daugybę duomenų patalpinti skaičiavimų debesyje, šios paslaugos tiekėjas, turintis pakankamus saugyklų ir kitų reikiamų kompiuterijos resursų, trečiosios šalies auditorius, kuris turi pakankamai gebėjimų stebėti skaičiavimų debesyje talpinamus duomenis ir yra įgaliotas duomenų savininko tai daryti, bei autorizuotos aplikacijos, kurios turi reikiamas teises nuskaityti ir manipuliuoti duomenimis.

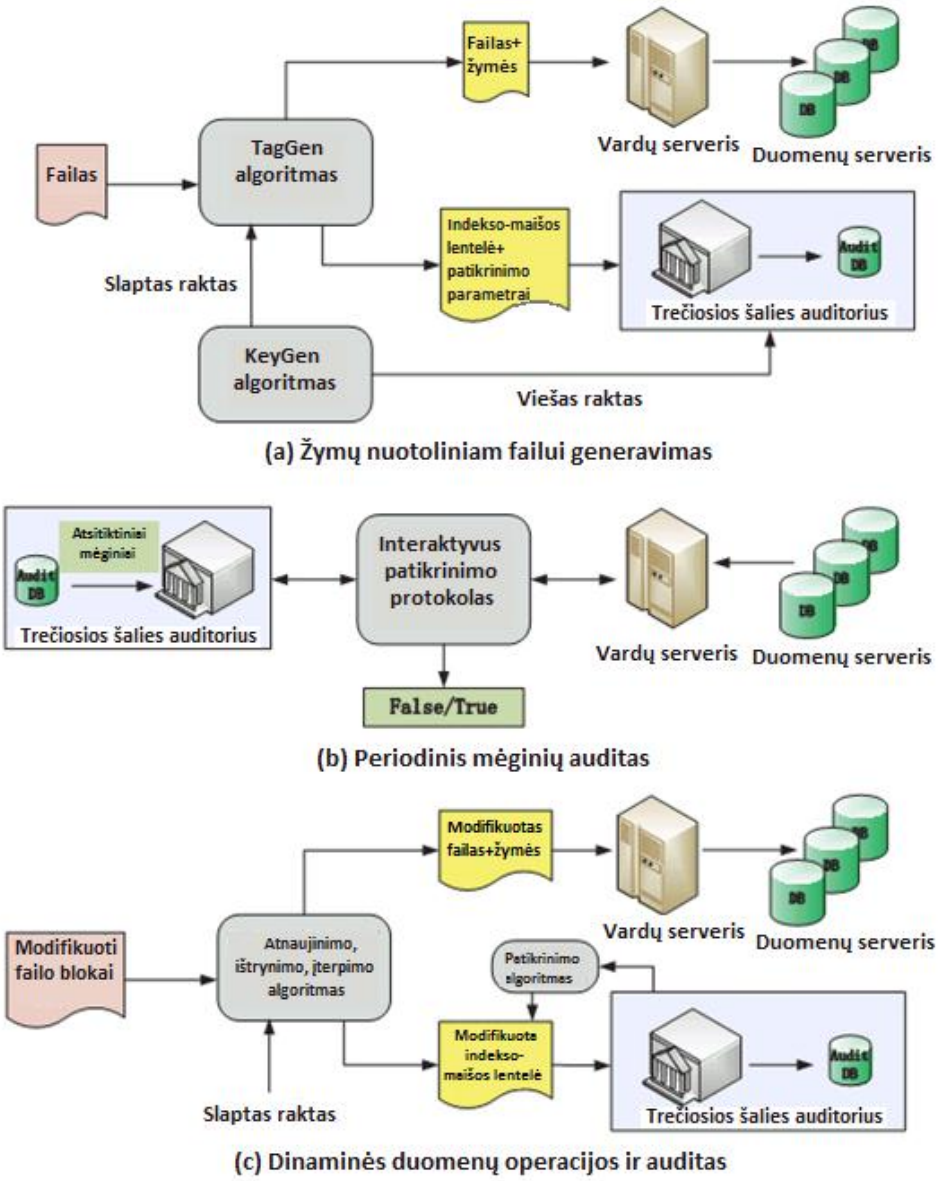


1.15 pav. Auditavimo sistemos su trečiąja šalimi architektūra

Kad įgyvendintų savo funkcijas, ši auditavimo sistema naudoja 3-is procesus:

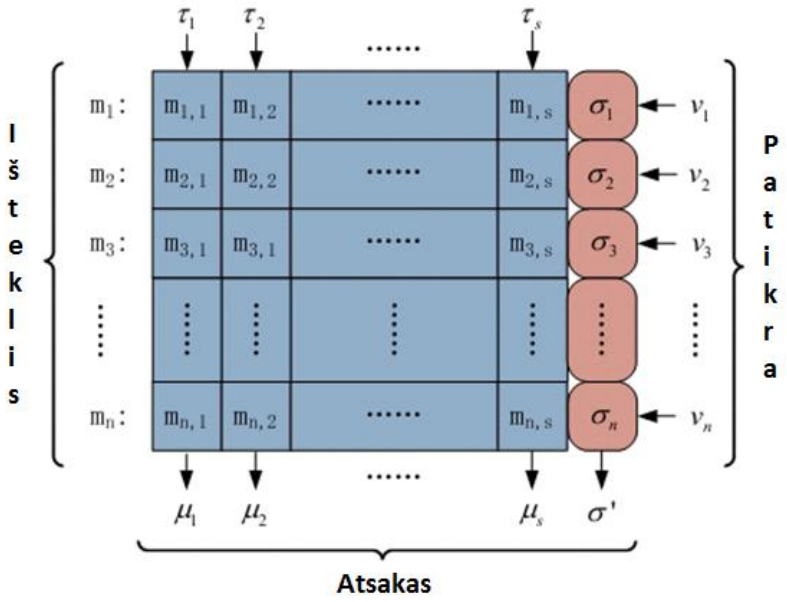
1) Žymų (angl. *tag*) generavimas: klientas (duomenų savininkas), pasinaudodamas slaptu raktu *sk*, apdoroja failą, sudarytą iš *n* blokų, ir sugeneruoja viešam patikrinimui skirtų parametrų rinkinį bei indekso-maišos (angl. *hash*) lentelę, kurie išsaugomi pas trečiosios šalies auditorių. Failas ir dalis patikrinimo žymų nusiunčiami skaičiavimų debesies teikėjui – po to visa lokaliai sukurta informacija gali būti ištrinta;
2) Periodinis mėginių auditas: trečiosios šalies auditorius specialaus protokolo pagalba tikrina atsitiktinę duomenų skaičiavimų debesyje dalį. Šiam tikslui pasinaudojama viešam patikrinimui skirtų parametrų rinkiniu bei indekso-maišos lentele;
3) Dinaminių operacijų auditavimas: autorizuotos aplikacijos, žinančios slaptą raktą *sk*, gali manipuliuoti skaičiavimų debesyje esančiais duomenimis, tuo pačiu atnaujinant pas trečiosios šalies auditorių saugomą indekso-maišos lentelę. Rakto *sk* slaptumas ir integralumo patikros algoritmas užtikrina, kad suklastoti duomenų nebūtų įmanoma.

Visi šie procesai yra pavaizduoti 1.16 paveiksle.



1.16 pav. Auditavimo sistemos procesai

Kad maksimizuoti saugyklos efektyvumą ir auditavimo našumą, siūloma audito sistema duomenų skaičiavimų debesyje apdorojimui naudoja fragmentais pagrįstą metodą. Failas yra padalintas į n blokų $\{m_1, m_2, ..., m_n\}$, o kiekvienas blokas m_i – į s sektorių $\{m_{i,1}, m_{i,2}, ..., m_{i,s}\}$. Sistema naudoja n blokų-žymų porų (m_i, σ_i), kur σ_i yra bloko m_i žyma, sugeneruota pasinaudojant paslapčių seka $\tau = (\tau_1, \tau_2, ..., \tau_s)$. Šios blokų-žymų poros yra saugomos skaičiavimų debesies tiekėjo, o paslapčių sekos – pas trečios šalies auditorių. Nors tokia sistema yra nesudėtinga ir tiesmuka, ji padeda sumažinti saugomų blokų-žymų skaičių (padidinant sektorių s skaičių) (1.17 pav.).



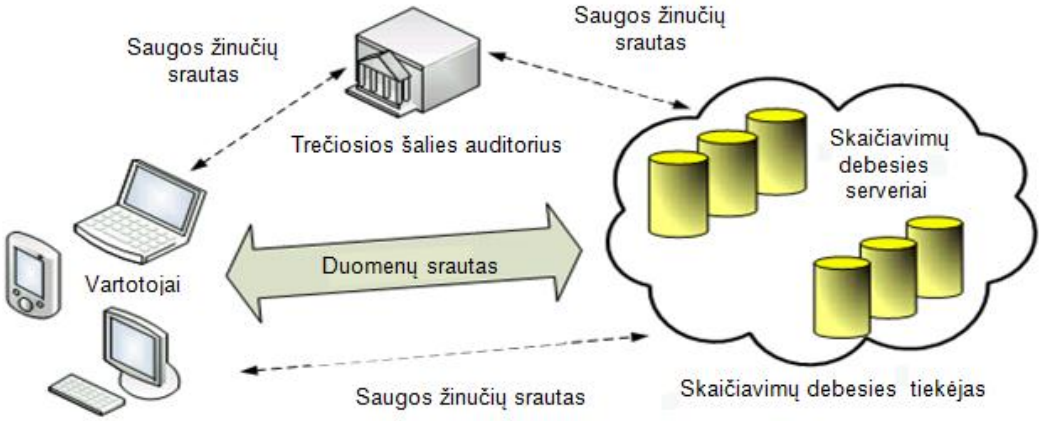
1.17 pav. Fragmentais pagrįstas duomenų apdorojimas

Eksperimentais įrodyta, kad tokia siūloma auditavimo sistema tik neženkliai ir pastoviai apkrauna kompiuterijos resursus, tokiu būdu užtikrindama sąlyginai nedideles papildomas išlaidas.

Šaltinis [15] siūlo labai panašią auditavimo sistemą su trečiąja šalimi, ypatingą dėmesį skiriant privatumo užtikrinimui. Autorių sukurtas metodas irgi išlaisvina skaičiavimų debesies vartotojus nuo sudėtingo ir brangaus auditavimo proceso bei apsaugo jų duomenis nuo galimo netyčinio paviešinimo iš trečiosios šalies auditoriaus pusės. Be to, siūloma sistema gali veikti ir daugelio vartotojų režime, t.y. vienu metu atlikti keletą skirtingų auditavimo užduočių.

1.18 pav. pateikta koncepcinė auditavimo sistemos schema, kurią sudaro 3-ys pagrindinės dalys: vartotojas, skaičiavimų debesies saugyklose saugantis didelį duomenų kiekį, šios paslaugos serveriai, kuriuos valdo ir organizuoja tiekėjas, galintis pasiūlyti reikiamą saugyklos ir kompiuterijos resursų dydį, ir trečiosios šalies auditorius, turintis žinių ir gebėjimų atlikti auditą skaičiavimų debesyje ir kuriuo pasitiki vartotojas nusamdydamas atlikti tokio pobūdžiu paslaugas.

Vartotojai visas debesies priežiūros ir aptarnavimo paslaugas patiki jų tiekėjui. Jie taip pat bet kada gali manipuliuoti savo patalpintais duomenimis, o, esant poreikiui, paprašyti trečiosios šalies auditoriaus, kad šis patikrintų nutolusioje aplinkoje talpinamų jų duomenų saugumą su garantija, kad duomenys išlaikys privatumą. Trečiosios šalies auditorius turi sugebėti efektyviai atlikti auditą skaičiavimų debesyje neturėdamas duomenų kopijų ir papildomai neapkraudamas vartotojo.



1.18 pav. Dar vienos siūlomos auditavimo sistemos su trečiąja šalimi schema

Siūlomą auditavimo sistemą sudaro keturi algoritmai: *KeyGen*, *SigGen*, *GenProof* ir *VerifyProof*). *KeyGen*, kurį naudoja vartotojas, sugeneruoja raktus, tokiu būdu inicializuodamas visą sistemą. Tas pats vartotojas algoritmo *SigGen* pagalba sukuria saugumo patikrinimui reikalingą informaciją, tokią kaip MAC reikšmes, maišos funkcijų ir kitus rezultatus, kurie vėliau yra naudojami auditavimo metu. Algoritmas *GenProof* vykdomas skaičiavimų debesies serverio pusėje, siekiant sukurti saugumo įrodymui skirtus duomenis, kuriuos vėliau algoritmo *VerifyProof* pagalba patikrina trečiosios šalies auditorius.

Visa auditavimo sistema veikia dviejų fazių principu:

1. Diegimas: vartotojas, pasinaudodamas algoritmu *KeyGen*, sugeneruoja reikiamus viešus bei slaptus raktus ir *SigGen* pagalba sukuria nutolusioje aplinkoje būsimiems patalpintiems failams reikalingą saugumo patikrinimo informaciją. Tada failai perkeliami į skaičiavimų debesį, ištrinant jų vietines kopijas, o patikrinimo informacija pateikiama trečiosios šalies auditoriui.
2. Auditavimas: trečiosios šalies auditorius specialaus protokolo pagalba užklausia skaičiavimų debesies serverio pateikti saugumo informaciją, kuri sugeneruojama *GenProof* algoritmo pagalba, ir ją patikrina su *VerifyProof* algoritmu.

Kad užtikrinti aukšto lygio privatumą, sistema naudoja taip vadinamą homomorfinių autentifikatorių metodą. Homomorfiniai autentifikatoriai yra sunkiai suklastojami patikrinimui skirti metaduomenys, sugeneruojami iš atskirtų duomenų blokų ir agreguojami į bendrą autentifikatorių, kurį ir naudoja auditoriai galutiniam patikrinimui. Iš eilės einančių duomenų blokų seka gali netyčia atskleisti privačią vartotojų informaciją, todėl autoriai papildė metodą atsitiktinėmis maskavimo technikomis. Homomorfinių autentifikatorių metodo efektyvumas sudaro ir galimybes audito sistemai veikti daugelio vartotojų režime. Tyrimais įrodyta, kad ji veikia užtikrintai saugiai ir greitai.

Siekiant užtikrintį reikiamą duomenų, talpinamų skaičiavimų debesyje, saugumą, būtina stebėti ne tik juos, bet ir pačius juos aprašančius audito įrašus, kurie gali būti suklastojami nusikalstamą veiklą vykdančių asmenų. Šią problemą sprendžia [17] straipsnio autoriai, pasiūlydami teorinį sprendimą, kuriame trečiųjų šalių pagalba nėra reikalinga (kad galima apseiti be pašalinių įsikišimo pademonstruoja ir [9]).

Kaip ir kiti autoriai, jie pabrėžia didelę audito įrašų svarbą, siekiant efektyviai tirti kompiuterinėje erdvėje įvykusius nusikaltimus. To pasekoje, įrašai neretai patys tampa nusikaltėlių taikiniu, kurie, siekdami paslėpti savo atliktus kenkėjiškus veiksmus, stengiasi audito įrašus ištrinti arba modifikuoti. Patyrę nusikaltėliai pirmiausia būtent to ir siekia.

Siūloma teorinė žurnalizavimo sistema, pavadinta „Blind-Aggregate-Forward“ (BAF), skirta didelėmss paskirstytosioms sistemos - skaičiavimų debesims. Ji pasiekia 6-is iš pažiūros tarpusavyje konfliktuojančius tikslus, susijusius su saugiu audito įrašų kūrimu: labai nedidelis dėl auditavimo sistemos padidėjęs kompiuterijos resursų išnaudojimas, beveik nulinė reikalinga vieta saugyklose, minimalus komunikacijos srautų padidėjimas, galimybė atlikti patikrinimą viešai be trečiosios šalies įsikišimo, galimybė patikrinimą atlikti labai greitai duotuoju momentu ir aukštas tikrinimo efektyvumas. Apibendrintos sistemos sąvybės pateiktos 1.3 lentelėje.

1.3 lentelė. Auditavimo sistemos BAF sąvybės

Sąvybė	Paaiškinimas
Efektyvus audito įrašų kūrimas	Naudojant BAF metodą, kompiuterijos resursų sukurti audito įrašą vienam duomenų objektui reikia tik tiek, kiek jų reikia atlikti tris kriptografines maišos funkcijų operacijas. Tai taip pat efektyvu, kaip ir simetrines kriptografijos schemas naudojantys metodai.
Efektyvus saugyklų ir komunikacijos kanalų išnaudojimas	Auditavimo sistema naudoja itin mažai saugyklų vietos ir labai minimaliai padidina komunikacijos kanalų užimtumą. Nepriklausomai nuo to, kiek laiko periodų ar duomenų vienetų yra pasirenkama, saugyklose sistemos sukurtų duomenų užimama vieta yra pastovi. Tai yra žymiai efektyviau nei simetrinius metodus naudojančiose sistemose.
Efektyvus audito įrašų patikrinimas	Kad patikrinti audito įrašus, yra panaudojama mažiau kompiuterijos resursų už kitus metodus.
Galimybė patikrinti viešai	BAF sukuria įrašus, kuriuos galima patikrinti viešai, kas labiau tinka paskirstytosioms sistemoms nei simetrines schemas naudojantys

	metodai.
Galimybė patikrinti be trečiųjų šalių auditorių ir duotuoju laiko momentu	Priešingai nei naudojant kitus metodus, su BAF trečiosios šalies auditorius yra nebūtinas. Tokiu būdu išvengiama komunikacijos kanalų apkrovimo padidėjimo, susidarančio tarpusavyje keičiantis žinutėmis auditoriui ir audito įrašų tikrintojui. Tai sumažina kritinių taškų (angl. *point of failure*) skaičių ir padidina sistemos efektyvumą. Be to, sistema sudaro galimybes patikrinimą atlikti operatyviai duotuoju laiko momentu, taip išvengiant galimų atakų tikrinant per ilgesnį laiko tarpą.

Dėl išvardintų sąvybių BAF sistema idealiai tinka atlikti saugaus auditavimo funkciją didelėse paskirstytose sistemose. Ji sukurta vadovaujantis sekančiais principais:

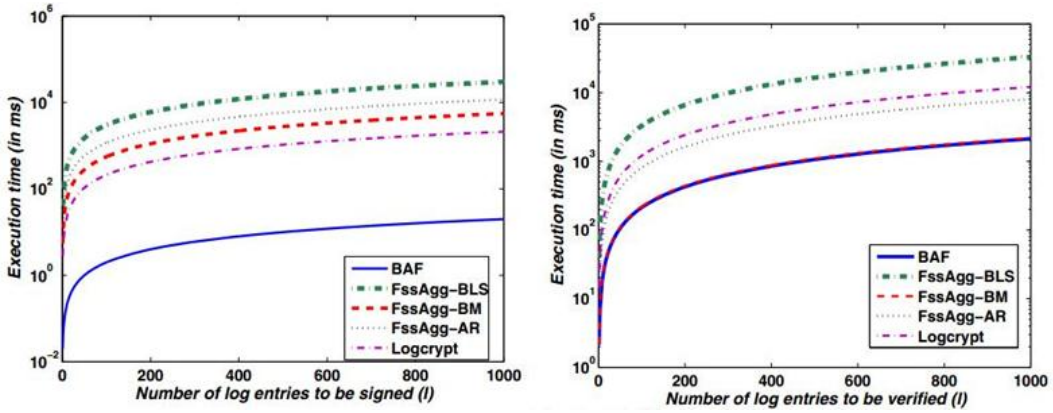
- Kuriant audito įrašus, vengti viešojo rakto infrastruktūros operacijų. Tai reikalauja per didelių kompiuterijos resursų, todėl BAF sistemoje šiam tikslui yra naudojamos tik pagrindinės aritmetinės operacijos ir kriptografinės funkcijos.
- Išvengti trečiosios šalies dalyvavimo. Priešingu atveju, operatyviai patikrinti audito įrašus būtų neįmanoma, be to, atsirastų grėsmė, jog ilgesnio patikrinimo metu duomenys gali būti sukompromituoti kenkėjiškų asmenų.

Sistema BAF patikrinimo duomenis generuoja 3-mis fazėmis:

1. Individualus skaitmeninio parašo generavimas: paprastos operacijos pagalba kiekvienam duomenų vienetui sukuriamas skaitmeninis parašas, panaudojant slaptus raktus. Šios operacijos rezultatas yra unikalus ir iš pažiūros atrodantis atsitiktinis, todėl jo suklastoti nežinant raktų yra neįmanoma.
2. Raktų atnaujinimas: raktų pora yra atnaujinama dviejų maišos funkcijos operacijų pagalba ir seni raktai iš atminties ištrinami.
3. Skaitmeninio parašo agregavimas: iš visų sugeneruotų skaitmeninių parašų specialios funkcijos pagalba yra suformuojamas vientisas skaitmeninis parašas, kurį gali patikrinti tik pirmuosius (ir jau ištrintus) slaptus raktus sugeneravęs asmuo.

Straipsnyje autoriai detaliai aprašo BAF sistemos veikime dalyvaujančias funkcijas ir, atlikę palyginimus su kitais metodais, įrodo, kad tai yra geriausias siekiant saugiai kurti audito įrašus didelėse paskirstytose net ir intensyviai apkrautose aplinkose (1.19 pav.).

	PKC-based					Symmetric
	BAF	FssAgg-BLS [8]	FssAgg-BM [1], [9]	FssAgg-AR [1], [9]	Logcrypt [11]	[2], [4], [5], [8]
Sig	0.06	30.0	5.55	11.66	2.11	0.06
Ver	2.11	33.0	2.2	8.1	12.09	0.06



1.19 pav. Auditavimo sistemos BAF palyginimas su kitais metodais

1.5.Išvados

Išanalizavę veiksmų kaip įkalčių atkūrimo skaičiavimų debesies saugyklose problemą, galime padaryti tokias išvadas:

- ✓ Skaičiavimų debesies technologija pasiūlė naujus būdus žymiai padidinti įmonių veiklos efektyvumą, o ateityje planuojamas ypač spartus ja besinaudojančių kompanijų skaičiaus augimas – tą patvirtina ekspertų prognozės;
- ✓ Nežiūrint į teigiamas puses, skaičiavimų debesis kelia įvairių susirūpinimų, vienas kurių - dėl nutolusių resursų sunkiai sprendžiamas skaitmeninės teismo ekspertizės atlikimo klausimas;
- ✓ Per pastarąjį dešimtmetį stipriai išaugo nusikaltimų, atliekamų elektroninėje erdvėje, skaičius, ir tuo pagrindu išsivystė skaitmeninės teismo ekspertizės sfera, užtikrinanti tinkamą elektroninių įkalčių pristatymą teismui;
- ✓ Skaičiavimų debesies paslaugų tiekėjai dar aiškiai neišsprendė klausimo, kaip jų teikiamos paslaugos bus pritaikytos teismo ekspertizės atlikimui, teismo ekspertai taip pat dar neturi aiškių procedūrų, kaip elgtis tyrimo debesyje atvejais - skaitmeninė teismo ekspertizė skaičiavimų debesyje yra vis dar besivystanti sfera, ir yra aktyviai siūloma įvairių metodų jai patobulinti;
- ✓ Bendrai sutinkama, kad jeigu skaičiavimų debesies paslauga nefiksuoja tinkamų audito įrašų, nustatyti įkalčius tampa sunku arba tiesiog neįmanoma, todėl šiam klausimui yra skiriamas didžiausias dėmesys;

- ✓ Vienas pagrindinių siūlymų skaitmeninės teismo ekspertizės skaičiavimų debesyje atlikimo efektyvumui padidinti yra perėjimas nuo apsaugos, kuri veikia iš sistemos perspektyvos, prie efektyvesnės, kuri veikia atskaitos tašku pasirinkdama failą: saugomi duomenys turi būti patys save apibūdinantys, kad būtų galima atsekti visą veiklą nuo jų sukūrimo iki sunaikinimo nepriklausomai nuo aplinkos apribojimų.

Toliau pateikiamas naujas metodas-įrankis, Žurnalizavimo Paramos Sistema (ŽPS), kuri padeda užfiksuoti ir atkurti vartotojų veiksmus kaip įkalčius skaičiavimų debesies saugyklose. ŽPS įgyvendina kitų autorių pasiūlytą unifikuotą audito įrašų formatą tokio pobūdžio aplinkoms ir sukuria save apsirašančių duomenų efektą, kuris, kaip buvo minėta anksčiau, yra svarbus žingsnis siekiant efektyviai tirti nusikaltimus skaičiavimų debesies saugyklose. Metodo efektyvumas įvertinamas atliktais tyrimais.

2. VEIKSMŲ KAIP ĮKALČIŲ ATKŪRIMO SKAIČIAVIMŲ DEBESIES SAUGYKLOSE METODIKA

2.1.Žurnalizavimo Paramos Sistemos koncepcija

Šiame skyriuje yra pateikiamas naujas metodas-įrankis, Žurnalizavimo Paramos Sistema (ŽPS), kuri, pasinaudodama atviro kodo priemonėmis, padeda, pirmiausia, užfiksuoti, o vėliau - atkurti vartotojų veiksmus kaip įkalčius skaičiavimų debesies saugyklose.

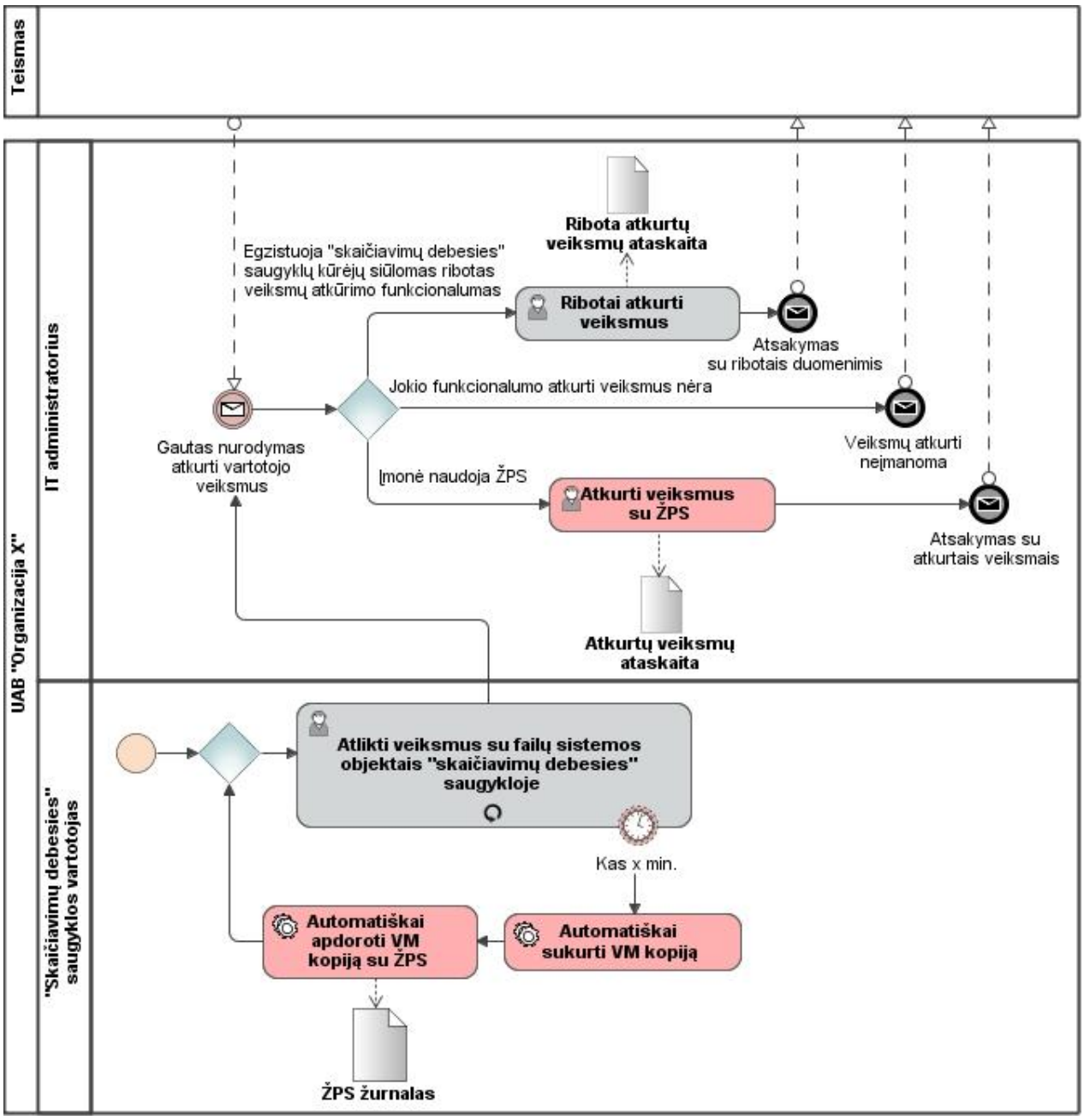
Prieš tai buvusiame skyriuje buvo išanalizuotas ir aiškiai išreikštas tokios paslaugos poreikis šių dienų teismo nagrinėjimo procesuose, kuriuose susiduriama su skaičiavimų debesies saugyklose galimai esamais įkalčiais. Problema iškyla todėl, kad nėra vieningų reikalavimų ar standartų, apibrėžiančių asmenų, besinaudojančių skaičiavimų debesies saugyklomis, veiksmų fiksavimo būtinumą ir konkrečias šio funkcionalumo įgyvendinimo gaires. Vadinasi, įvykus nusikaltimui ir prireikus tokius veiksmus atkurti, teismas turi pasikliauti skaičiavimų debesies saugyklos paslaugos kūrėjų suteiktu vidiniu tam tikslui skirtu funkcionalumu, kuris ne tik kad gali būti nepakankamas – kaip paaiškėjo atlikus šaltinių analizę, jis dažniausiausiai neegzistuoja išvis.

Siūloma Žurnalizavimo Paramos Sistema skirta fiksuoti visus nustatytus duomenis iš skaičiavimų debesies saugyklų ir yra alternatyva prieš tai dviem paminėtiems atvejams. ŽPS vieta nagrinėjamos problemos kontekste pavaizduota 2.1 pav.

Ji suprojektuota taip, kad veiktų kaip atskiras komponentas, savarankiška programinė įranga, kurią savo pasirinkimu galėtų įdiegti bet kokia privati kompanija, kuri naudojasi lokaliomis skaičiavimų debesies saugyklomis. 2.2 pav. pateiktas bendras diegimo modelis, kuriame pavaizduotas galimas variantas, kai ŽPS diegiama virtualizacijos platformos hipervizoriuje (angl. *hypervisor*).

Bazines skaičiavimų debesies paslaugas organizuoja specializuota programinė įranga - hypervizorius, kuri yra valdoma iš taip vadinamos kontroliuojančios operacinės sistemos (paprastai veikiančios Unix pagrindu). Jos įrankių pagalba yra organizuojamas virtualių mašinų (VM) darbas: kūrimas, paleidimas, stabdymas, sunaikinimas ir kt.

VM pagalba yra teikiamos skaičiavimų debesies saugyklų paslaugos. Asmenys jungiasi internetu per tam nustatytas, specialiai suprogramuotas prieigas (paprastai per naršyklę) ir naudojasi suteikta virtualia failų sistema taip, lyg ji būtų jų kompiuteryje: gali kurti, redaguoti, kopijuoti, naikinti failus, atlikti analogiškus veiksmus su katalogais.

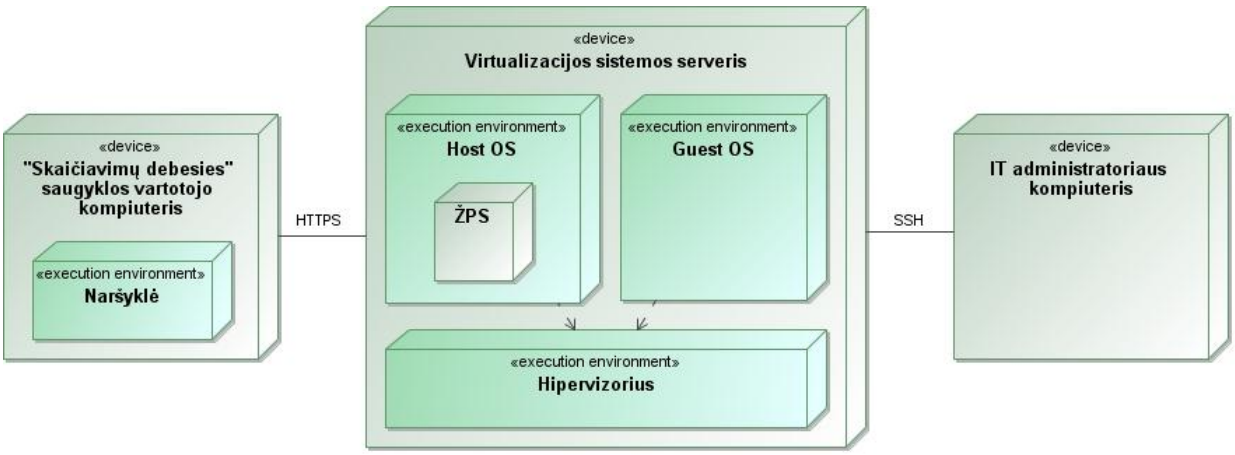


2.1 pav. Žurnalizavimo Paramos Sistemos vieta problemos kontekste (BPMN modelis)

Žurnalizavimo Paramos Sistemą funkcionalumo prasme sudaro 2-i dalys:

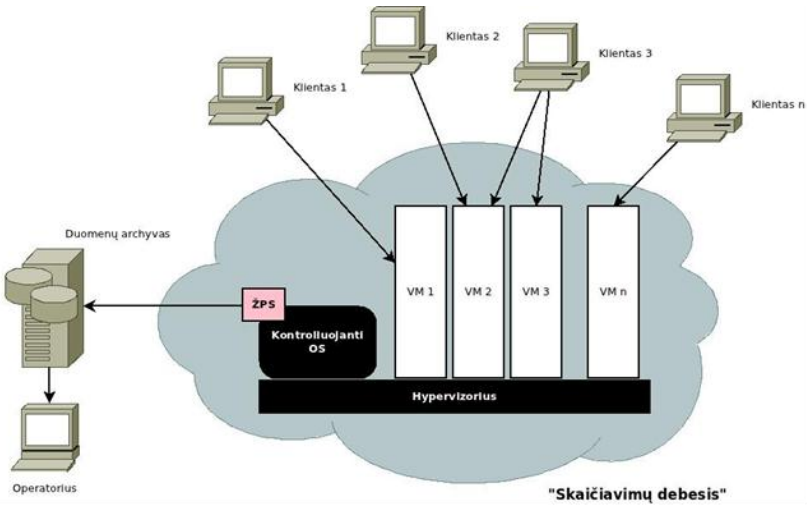
I. Veiksmus fiksuojanti: kas nustatytą laiko tarpą apdoroja naujai atsiradusias virtualių mašinų kopijas ir papildo ŽPS audito žurnalą;

II. Veiksmus atkurianti: atkuria veiksmus pasirinktame laikotarpyje. Tai yra programa, kuri perskaito veiksmus fiksuojančios programos sukurtus įrašus ir atrenka reikiamus tam momentui.



2.2 pav. ŽPS diegimo modelis

Tiek veiksmus fiksuojančios, tiek veiksmus atkuriančios dalys gali būti įdiegtos ne tik į hipervizorių kontroliuojančią operacinę sistemą, bet ir į bet kokią pasirinktą Windows arba Linux kompiuterinę sistemą pagal patogumą – svarbiausia, kad iš jų būtų galima pasiekti paruoštas VM kopijas ir vėliau ŽPS suformuotą audito žurnalą (norint atkurti veiksmus). Pavyzdžiui, įrašai nustatytais laiko tarpais gali būti išsiunčiami iš kontroliuojančios operacinės sistemos į specialiai jiems saugoti skirtas vietas, tarkime, dedikuotas duomenų saugyklas, o, atsiradus poreikiui, juos iš ten vėliau gali pasiimti ir su veiksmus atkuriančia programa apdoroti už tai atsakingi asmenys (2.3 pav.).



2.3 pav. ŽPS sukurtų audito įrašų persiuntimo į dedikuotus duomenų serverius schema

2.2.Virtualių mašinų kopijų kūrimas

Siūlomos Žurnalizavimo Paramos Sistemos veikimui yra reikalingos „aktyvios“ virtualių mašinų kopijos (angl. *snapshot*), pasižyminčios tokiomis sąvybėmis:

- Turi būti 2-u failai: virtualaus HDD kopija ir virtualaus RAM kopija;
- HDD kopija turi būti išsaugota (esant reikalui, konvertuota) vadinamuoju „grynu“ (angl. *raw*) formatu;
- RAM kopija taip pat turi būti pateikta „grynu“ formatu.

Ypatingo dėmesio reikalauja antrasis reikalavimas. Realioje verslo aplinkoje, kur yra naudojamos virtualios mašinos, virtualus diskas yra saugomas keleto failų pavidalu: dažniausiai naudojamas pagrindinis ir neribotas kiekis diferencinių virtualių diskų (pastarieji paprastai sukuriami po kiekvienos aktyvios kopijos sukūrimo operacijos). Pagrindinis virtualus diskas dėl didesnio efektyvumo taip pat dažnai saugojamas keleto failų pavidalu. Tam, kad virtualų HDD būtų galima analizuoti ŽPS priemonės pagalba, būtinas vienas failas, todėl virtualios mašinos „aktyvios“ kopijos atlikimo metu visi virtualaus disko failai turi būti sujungti į vieną.

Kaip, lengvai ar sudėtingai, galima patenkinti šias visas sąvybes, priklauso nuo specifinės virtualizacijos platformos. Vienos iškart pateikia visus reikiamus įrankius (paprastai veikiančius iš komandinės eilutės), kitos tokį funkcionalumą siūlo tik naudojantis trečiųjų šalių sukurtais, dažniausiai mokamais įrankiais.

Dėl šių priežasčių ŽPS priemonė atlieka tik jau tinkamai paruoštų ir į nustatytą direktoriją patalpintų VM kopijų apdorojimo funkcionalumą, atsiribodama nuo kopijų kūrimo klausimo ir su juo susijusios problematikos.

2.3.Veiksmų fiksavimas

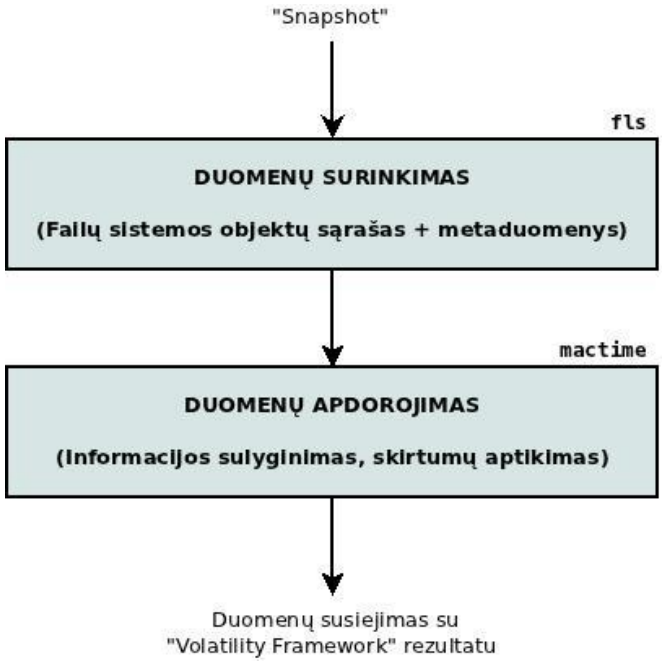
Žurnalizavimo Paramos Sistema yra pagrįsta „Python“ programavimo kalba apjungiant dvi nemokamas atviro kodo programines priemones: „The Sleuth Kit“ (TSK) [26] ir „The Volatility Framework“ (TVF) [27]. TSK pagalba iš virtualių mašinų kopijų yra išgauna visa reikalinga su fizine failų sistema susijusi informacija, o TVF suteikia galimybes surasti reikiamus artefaktus operatyviojoje atmintyje.

2.3.1. Failų sistemos analizė „The Sleuth Kit“ priemonėmis

„The Sleuth Kit“ yra nemokama atvirodo kodo C biblioteka ir komandinės eilutės įrankių rinkinys, kurie skirti failų sistemos analizei skaitmeninės teisminės ekspertizės tikslams. Ši sistema turi įvairių galimybių, bet mūsų siūlomo metodo atveju svarbiausios yra šios:

- Geba analizuoti „grynas“ diskų kopijas;
- Palaiko pagrindines, populiariausias failų sistemas: NTFS, FAT, UFS 1, UFS 2, EXT2FS, EXT3FS, HFS ir ISO 9660;
- Suteikia galimybę nuskaityti failų sistemos objektų metaduomenis;
- Aptinka ištrintus dokumentus;
- Turi paruoštas priemones veiksmų su failų sistemos objektais fiksavimui.

Veiksmų fiksavimas ir yra pagrindinis TSK priemonių panaudojimo siūlomoje Žurnalizavimo Paramos Sistemoje tikslas. Rezultatas pasiekimas dvejais žingsniais: duomenų surinkimu ir duomenų apdorojimu (2.4 pav.).



2.4 pav. „The Sleuth Kit“ priemonių panaudojimo veiksmų fiksavimui schema

Duomenų surinkimo žingsnyje iš virtualios mašinos kopijos yra nuskaitomas visų esamų failų ir katalogų sąrašas su meta duomenimis. Tai atliekama įrankio *fls* pagalba, kuris aptinka ir ištrintus failus (2.5 pav.).

```
AttrDef|4-128-4|r/rr-xr-xr-x|48|0|2560|1328252048|1328252048|1328252048|1328252048
BadClus|8-128-2|r/rr-xr-xr-x|0|0|0|1328252048|1328252048|1328252048|1328252048
BadClus:$Bad|8-128-1|r/rr-xr-xr-x|0|0|37578862592|1328252048|1328252048|1328252048|1328252048
Bitmap|6-128-4|r/rr-xr-xr-x|0|0|1146816|1328252048|1328252048|1328252048|1328252048
Boot|7-128-1|r/rr-xr-xr-x|48|0|8192|1328252048|1328252048|1328252048|1328252048
Extend|11-144-4|d/dr-xr-xr-x|0|0|552|1328252048|1328252048|1328252048|1328252048
Extend/$ObjId:$O|25-144-5|r/rr-xr-xr-x|0|0|344|1328252051|1328252051|1328252051|1328252051
Extend/$Quota:$O|24-144-3|r/rr-xr-xr-x|0|0|88|1328252051|1328252051|1328252051|1328252051
Extend/$Quota:$Q|24-144-2|r/rr-xr-xr-x|0|0|208|1328252051|1328252051|1328252051|1328252051
Extend/$Reparse:$R|26-144-5|r/rr-xr-xr-x|0|0|56|1328252051|1328252051|1328252051|1328252051
Extend/$RmMetadata|27-144-2|d/dr-xr-xr-x|0|0|336|1328252051|1328252051|1328252051|1328252051
Extend/$RmMetadata/$Repair|28-128-4|r/rr-xr-xr-x|0|0|0|1328252051|1328252051|1328252051|13282
Extend/$RmMetadata/$Repair:$Config|28-128-2|r/rr-xr-xr-x|0|0|8|1328252051|1328252051|132825205
Extend/$RmMetadata/$Txf|30-144-17|d/dr-xr-xr-x|0|0|48|1331643756|1331643756|1331643756|132825
Extend/$RmMetadata/$Txf/0000000000000B21 (deleted)|95603-128-1|-/rrwxrwxrwx|0|0|7156|13283666
Extend/$RmMetadata/$Txf/0000000000000B24 (deleted)|95617-128-1|-/rrwxrwxrwx|0|0|3528|13283666
Extend/$RmMetadata/$Txf/0000000000000B34 (deleted)|96874-128-1|-/rrwxrwxrwx|0|0|584|132836661
Extend/$RmMetadata/$Txf/0000000000000B35 (deleted)|96876-128-1|-/rrwxrwxrwx|0|0|3976|13283666
Extend/$RmMetadata/$Txf/0000000000000B3F (deleted)|96886-128-1|-/rrwxrwxrwx|0|0|25852|1328366
```

2.5 pav. Įrankio *fls* išvestis

Iš surinktų duomenų ŽPS svarbiausia yra ši informacija:

- Duomenų surinkimo laikas;
- Failų sistemos objektų pavadinimai;
- Objektų sukūrimo laikai;
- Objektų modifikavimo laikai;
- Objektų ištrynimo laikai.

Duomenų apdorojimo žingsnyje paskutiniai duomenys yra sulyginami su surinktais ankstesnio seanso metu ir, tokiu būdu aptikus skirtumus, yra užfiksuojami su failų sistemos obektais atlikti veiksmai: sukūrimas, nuskaitymas, redagavimas ir ištrynimas. Šį veiksmą atlieka TSK įrankis *mactime* (2.6 pav.).

```
Columns:
      Date/Time              Size    Activity        Unix      User     Group     inode     File Name
                           (Bytes)    Type        Permissions   Id        Id
Example:
[...]
Thu Aug 21 2003 01:20:38      512       m.c.       -/-rwxrwxrwx     0        0        4        /file1.dat
                              900       m.c.       -/-rwxrwxrwx     0        0        8        /file3.dat
Thu Aug 21 2003 01:21:36      512       m.c.       -/-rwxrwxrwx     0        0        12       /_ILE5.DAT (deleted)
Thu Aug 21 2003 01:22:56      512       .a..       -/-rwxrwxrwx     0        0        4        /file1.dat
[...]
```

2.6 pav. Įrankio *mactime* išvestis

Gautas rezultatas perduodamas tolimesniam ŽPS apdorojimui.

2.3.2. Operatyviosios atminties analizė „The Volatility Framework“ priemonėmis

„The Volatility Framework“ yra nemokamų atviro kodo įrankių rinkinys, sukurtas „Python“ programavimo kalba ir skirtas skaitmeninių artefaktų operatyviosios atminties (RAM) kopijose suradimui. TVF geba nuskaityti „aktyvias“ virtualių mašinų kopijas ir supranta populiariausių šiandienos operacinių sistemų šeimų (Windows, Unix) RAM valdymo ypatumus.

Žurnalizavimo Paramos Sistemos atveju, „The Volatility Framework“ parametro *netstat* pagalba iš virtualios RAM kopijos yra nuskaitomi prie VM duotuoju momentu prisijungę IP adresai, kurie vėliau susiejami su „The Sleuth Kit“ priemonėmis užfiksuotais veiksmais, atliktais su failų sistemos objektais. Įrankio išvestis pateikta 2.7 pav.

```
0x17de8980 TCPv6    :::49153                :::0                   LISTENING        444      lsass.exe
0x17f35240 TCPv4    0.0.0.0:49155           0.0.0.0:0              LISTENING        880      svchost.exe
0x17f362b0 TCPv4    0.0.0.0:49155           0.0.0.0:0              LISTENING        880      svchost.exe
0x17f362b0 TCPv6    :::49155                :::0                   LISTENING        880      svchost.exe
0xfd96570  TCPv4    -:0                     232.9.125.0:0          CLOSED           1        ?C?
0x17236010 TCPv4    -:49227                 184.26.31.55:80        CLOSED           2820     iexplore.exe
0x1725d010 TCPv4    -:49359                 93.184.220.20:80       CLOSED           2820     iexplore.exe
0x17270530 TCPv4    10.0.2.15:49363         173.194.35.38:80       ESTABLISHED      2820     iexplore.exe
0x17285010 TCPv4    -:49341                 82.165.218.111:80      CLOSED           2820     iexplore.exe
0x17288a90 TCPv4    10.0.2.15:49254         74.125.31.157:80       CLOSE_WAIT       2820     iexplore.exe
0x1728f6b0 TCPv4    10.0.2.15:49171         204.245.34.130:80      ESTABLISHED      2820     iexplore.exe
0x17291ba0 TCPv4    10.0.2.15:49347         173.194.35.36:80       CLOSE_WAIT       2820     iexplore.exe
```

2.7 pav. Parametro *netstat* išvestis

2.3.3. Galutinis duomenų apdorojimas

Virtualių mašinų kopijos yra kuriamos kas nustatytą, laisvai pasirenkamą laiko tarpą (žr. 3-iame skyriuje pateiktą nuo to priklausantį Žurnalizavimo Paramos Sistemos efektyvumo įvertinimą) ir tiek „The Sleuth Kit“, tiek „The Volatility Framework“ įrankių sugeneruoti duomenys išsaugomi ŽPS audito žurnaluose, kurių formatas atitinka [11] reikalavimus ir kuriuose dėl patogumo naudojamas Unix laiko formatas (2.8, 2.9 pav.). Bylos nagrinėtojams pareikalavus atkurti veiksmus, jie yra atkuriami būtent iš šių failų pasinaudojus atskiru ŽPS sistemos įrankiu (žr. toliau skyrių „Veiksmų atkūrimas“).

```
time=1358303434,object=./data/folder1/,action=created
time=1358303438,object=./data/folder1/file3,action=created
time=1358303441,object=./data/folder1/file3,action=modified
time=1358303444,object=./data/folder1/folder4/,action=created
time=1358303447,object=./data/folder1/file3,action=deleted
```

2.8 pav. ŽPS audito žurnalo įrašas su užfiksuotais failų sistemos objektų pokyčiais

```
time=1358289391,object=[10.0.2.2,65.7.11.15]
time=1358293119,object=[192.168.14.22]
time=1358299579,object=[10.0.2.15,192.168.0.14,213.15.26.5]
```

2.9 pav. ŽPS audito žurnalo įrašas su užfiksuotais IP adresais

Kadangi „The Sleuth Kit“ nuskaito failų sistemos objektų meta duomenis, visais atvejais yra pateikiama galutinė ir užtikrinta informacija apie su objektais atliktus veiksmus. Turint omeny, kad vartotojams paprastai yra suteikiama prieiga ne prie absoliučiai visų, o tik specialiai tam paskirtų failų sistemos resursų (atskirų particijų arba katalogų), Žurnalizavimo Paramos Sistemos darbas yra optimizuojamas konfigūracijoje nurodant atitinkamas failų sistemos dalis, kurių veiksmus reikia fiksuoti (2.10 pav.).

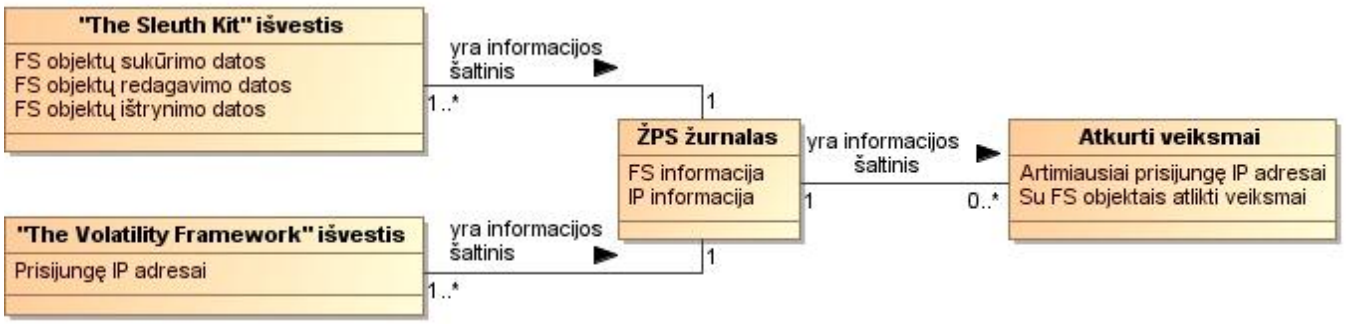
```
[DATA]
VM=10012511
/Letters/
/Shared Folder/Tomas/Approved Data/

VM=88551124
/Business Data/Forms/HR/

VM=21546546
/
/home/tomas/logs/
/resources/local/invoices/2013/02/15
[DATA-END]
```

2.10 pav. ŽPS konfigūracinio failo turinys

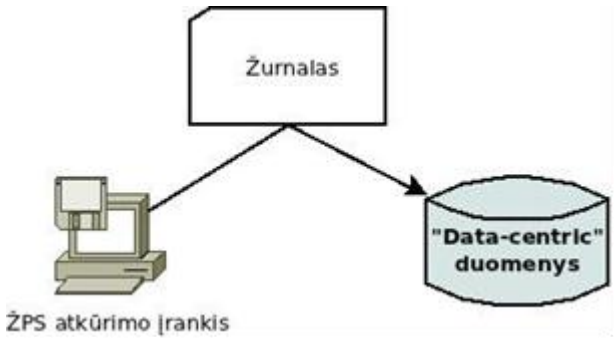
Tuo tarpu „The Volatility Framework“ analizuoja laikiną informaciją, - operatyviosios atminties būklė yra nuolat kintanti, - todėl užtikrinti, kad bus laiku užfiksuoti visi reikiami tinklo sujungimai prieš jiems baigus darbą, yra neįmanoma. Žurnalizavimo Paramos Sistema TVF pagalba išsaugo visus prie VM prisijungusius tinklo IP adresus nepriklausomai nuo to, ar jie galimai susiję su „The Sleuth Kit“ dalies užfiksuotais failų sistemos objektų pasikeitimais, ir pateikia tai kaip papildomą informaciją bylos nagrinėtojams. 2.11 pav. pateikiamas ŽPS koncepcinis duomenų modelis.



2.11 pav. ŽPS koncepcinis duomenų modelis

2.4.Veiksmų atkūrimas

Veiksmams atkurti yra naudojamas atskiras, specialiai siūlomai Žurnalizavimo Paramos Sistemai „Python“ programavimo kalba sukurtas įrankis. Jo veikimo principas yra paprastas: nuskaityti audito žurnalus, kuriuose yra išsaugota ŽPS surinkta informacija, ir suformuoti kitą failą su pagal nurodytus filtrus atrinktais duomenimis (2.12 pav.).



2.12 pav. ŽPS atkūrimo įrankio veikimo schema

Įrašus galima surasti pagal:

- Datą;
- IP adresą;
- Failų sistemos objektą (-us);
- Veiksmą (sukūrimas, redagavimas, ištrynimas).

Šie filtrai gali būti kombinuojami tarpusavyje - parametrai nurodomi komandinės eilutės pagalba. 2.13 pav. pateiktas atkurtų veiksmų, susijusių su failu „file1“, audito įrašų pavyzdys. Išvestyje įrašomi prieš failų sistemos objektų pasikeitimus arčiausiai užfiksuoti IP adresai. Gauname atkuriamąją informaciją apie objektą, nepriklausomai nuo to, kokiose sistemose ar priemonėse jis buvo naudojamas.

Tokiu būdu įgyvendiname [8] ir [18] pasiūlytą save apsirašančių duomenų koncepciją, kuri, kaip buvo aptarta 1-ame skyriuje, yra svarbus žingsnis siekiant efektyviai tirti nusikaltimus skaičiavimų debesies saugyklose.

```
time=1358225932,object=[10.0.2.15]
time=1358228301,object=./data/file1,action=created
time=1358229629,object=./data/file1,action=modified
time=1358233118,object=[10.0.2.15,192.168.0.147,213.15.26.5]
time=1358234629,object=./data/file1,action=modified
time=1358234660,object=./data/file1,action=deleted
```

2.13 pav. ŽPS atkurti veiksmai, susiję su failų sistemos objektu „file1“

Veiksmų atkūrimo priemonė buvo specialiai sukurta kaip nepriklausoma ŽPS dalis, kad ja būtų galima naudotis skirtingoje aplinkoje, negu yra atliekamas įrašų fiksavimas. Tai didžiąja dalimi susiję ir su tuo, kad paprastai žurnalizavimo įrašai yra saugomi atskirose duomenų saugyklose, ir jų analizę atlieka kiti, specialiai už tai atsakingi asmenys. Be to, egzistuojant tokiai ŽPS architektūrai, galima sukurti papildomas žurnalizavimo įrašų atkūrimo priemones priklausomai nuo konkrečių vartotojų poreikių: su grafine vartotojo aplinka ir kitomis galimybėmis.

2.5.Išvados

Apibendrinant siūlomą metodiką, galima padaryti tokias išvadas:

- ✓ Žurnalizavimo Paramos Sistema padeda užfiksuoti ir vėliau atkurti vartotojų veiksmus kaip įkalčius skaičiavimų debesies saugyklose, nepriklausomai nuo to, ar šiuo klausimu debesies paslaugos tiekėjas siūlo kokį nors funkcionalumą (paprastai jis neegzistuoja išvis);
- ✓ ŽPS yra pagrįsta „Python“ programavimo kalba apjungiant dvi nemokamas atvirojo kodo programines priemones, skirtas skaitmeninės teismo eskpertizės atlikimui – „The Sleuth Kit“ ir „The Volatility Framework“: TSK pagalba iš virtualių mašinų kopijų yra išgauna visa reikalinga su fizine failų sistema susijusi informacija, o TVF suteikia galimybes surasti reikiamus artefaktus operatyviojoje atmintyje;
- ✓ Kas nustatytą laiko tarpą analizuodama „aktyvias“ virtualių mašinų kopijas, Žurnalizavimo Paramos Sistema audito žurnaluose išsaugo failų sistemos objektų sukūrimo, modifikavimo ir ištrynimo laikus bei duotuoju momentu prie VM prisijungusius IP adresus, kas vėliau padeda atkurti vartotojų veiksmus kaip įkalčius skaičiavimų debesies saugyklose;

- ✓ ŽPS įgyvendina kitų autorių pasiūlytą unifikuotą audito įrašų kūrimo formatą tokio pobūdžio aplinkoms ir sukuria taip vadinamą save apsirašančių duomenų efektą, kuris, manoma, yra būtinas žingsnis siekiant efektyviai tirti nusikaltimus skaičiavimų debesyje.

Kitame skyriuje pateikiamas Žurnalizavimo Paramos Sistemos efektyvumo įvertinimas.

3. METODIKOS EFEKTYVUMO ĮVERTINIMAS

3.1.Metodikos realizacija

„Python“ programavimo kalbos pagalba sukurtą Žurnalizavimo Paramos Sistemos priemonę sudaro 2-u pagrindiniai ir 3-s papildomi scenarijai (angl. *script*) (žr. 3.1 lent.).

3.1 lentelė. ŽPS priemonę sudarantys scenarijai

Pagrindiniai scenarijai (pagrindinis ŽPS funkcionalumas)	
- **`zps-snap.py`** - **`zps-restore.py`**	Apdoroja VM kopiją Atkuria veiksmus
Papildomi scenarijai (tyrimo tikslams)	
- **`user-imitate.py`** - **`zps-log.py`** - **`zps-compare.py`** - **`make-ram-copy.sh`**	Imituoja vieno vartotojo veiksmus Paruošia ŽPS informaciją tyrimui Palygina realius faktus su užfiksuotais ŽPS Sukuria RAM kopiją

Scenarijus **`zps-snap.py`** suteikia pagrindinį ŽPS funkcionalumą. Jis nuolat tikrina nustatytą direktoriją ir apdoroja ten talpinamas virtualių mašinų kopijas. Užfiksuota informacija išsaugoma dviejuose failuose:

- **`zps-fs-temp.log`**. Saugoma anksčiau aprašyta „The Sleuth Kit“ priemonės įrankio **`fls`** išvestis (2.5 pav.). Tai tarpinė informacija apie failų sistemos objektų (failų ir katalogų) meta duomenimis užfiksuotus sukūrimo, modifikavimo ir ištrynimo laikus. Ji išgaunama iš virtualaus HDD kopijos ir panaudojama vėliau atkuriant veiksmus su **`zps-restore.py`** scenarijumi, be to, ruošiant informaciją tyrimui scenarijaus **`zps-log.py`** pagalba.
- **`zps-ip.log`**. Rašoma konkrečiu laiko momentu (VM kopijos atlikimo laiku) užfiksuoti prisijungę IP adresai, gauti panaudojus „The Volatility Framework“ priemonės parametrą **`netstat`** (2.9 pav.). Šaltinis – virtualios RAM kopija. Laikas dėl patogumo saugomas UNIX formatu. Šis žurnalas vėliau panaudojamas atkuriant veiksmus su **`zps-`**

`restore.py`, taip pat tyrimo tikslais vykdant scenarijų **`zps-compare.py`**.

Kitas pagrindinis scenarijus, **`zps-restore.py`**, skirtas veiksmų atkūrimui. Jis naudojasi **`zps-snap.py`** užfiksuotais tarpiniais duomenimis ir atkurtą informaciją išveda į ekraną arba nurodytą failą pagal pasirinktus anksčiau aprašytus filtrus, kokius veiksmus yra norima atkurti (2.13 pav.). Išvestyje įrašomi prieš failų sistemos objektų pasikeitimus arčiausiai užfiksuoti IP adresai. Laikas vėlgi pateikiamas UNIX formatu, kurį lengva paversti į bet kokį norimą formatą (pagal poreikį).

Papildomi scenarijai, sukurti tik šio tyrimo tikslams, plačiau detalizuojami 3.2.2 skyriuje.

3.2.Eksperimentinis tyrimas

3.2.1. Naudota įranga

Buvo naudojamas kompiuteris „HP nx6125“ (3.2 lent.) su „Lubuntu 12.10“ 64 bitų operacine sistema („Linux“ branduolio versija – „3.5.0-21-generic“), kurioje įdiegta „Oracle VirtualBox 4.1.18“ virtualizacijos platforma. Sukurtos 6-ios virtualios mašinos su „Debian 4.3.5-4“ 32 bitų operacine sistema („Linux“ branduolio versija – „2.6.32-5-686“). Viena buvo naudojama kaip tyrimo objektas – būtent jos kopijos ir buvo kuriamos bei analizuojamos, o likusios buvo panaudotos kaip skirtingų 5-ių vartotojų kompiuterių imitavimas, kuriais buvo jungiamasi prie pagrindinės VM ir atliekami veiksmai su failų sistemos objektais. Žurnalizavimo Paramos Sistema sistema buvo leidžiama iš pagrindinės kompiuterio OS.

3.2 lentelė. Tyrime panaudoto kompiuterio techniniai parametrai

Procesorius	AMD Turion 64 Mobile Technology ML-30 (1.6-GHz, 1-MB L2 cache)
Pagrindinė plokštė	ATI RADEON XPRESS 200M Chipset
RAM	2048-MB 333-MHz DDR SDRAM
HDD	50-GB 5400 rpm
Vaizdo korta	Integrated ATI MOBILITY RADEON X300 (128-MB allocated system memory)

Papildoma tyrime naudota specifinė programinė įranga pateikta 3.3 lentelėje.

3.3 lentelė. Tyrime panaudota specifinė programinė įranga

Programinė priemonė	**Kam panaudota tyrimo metu**
„The Sleuth Kit 4.0.1“	Nuskaityti virtualaus HDD kopijos failų sistemos objektų

	meta duomenis
„Volatility Framework 2.2“	Nuskaityti prisijungusius IP iš virtualios RAM kopijos
„LiME 1.1-r14“	Sukurti virtualios mašinos RAM kopiją
„qemu-img 1.2.0“	Konvertuoti HDD kopiją iš VMDK į „gryną“ formatą

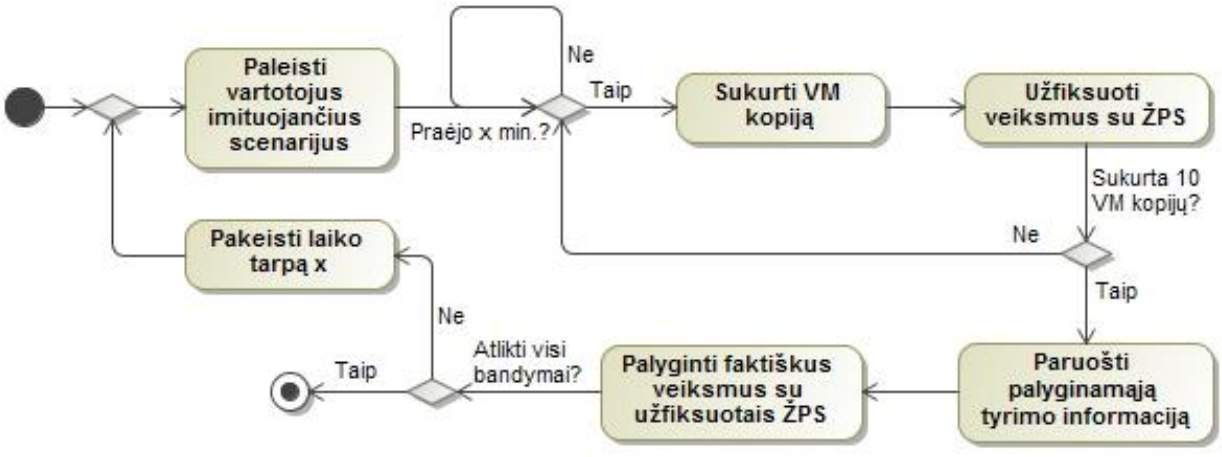
3.2.2. Eksperimentinio tyrimo metodika

Siūlomas Žurnalizavimo Paramos Sistemos metodas-įrankis buvo tiriamas šiais pjūviais:

- ✓ Atkurtos informacijos kiekio priklausomybė nuo virtualių mašinų kopijų kūrimo (ir atitinkamai apdorojimo su ŽPS) dažnumo;
- ✓ Atkurtos informacijos kiekio priklausomybė nuo vartotojų veiksmų dažnumo;
- ✓ ŽPS veikimo trukmės priklausomybė nuo virtualaus HDD dydžio;
- ✓ ŽPS veikimo trukmės priklausomybė nuo virtualios RAM dydžio.

Pats svarbiausias šio darbo uždavinys buvo nustatyti, kaip priklauso atkurtos informacijos kiekis nuo VM kopijų kūrimo dažnumo. Kad ištirti šią priklausomybę, buvo atliekami tokie veiksmai:

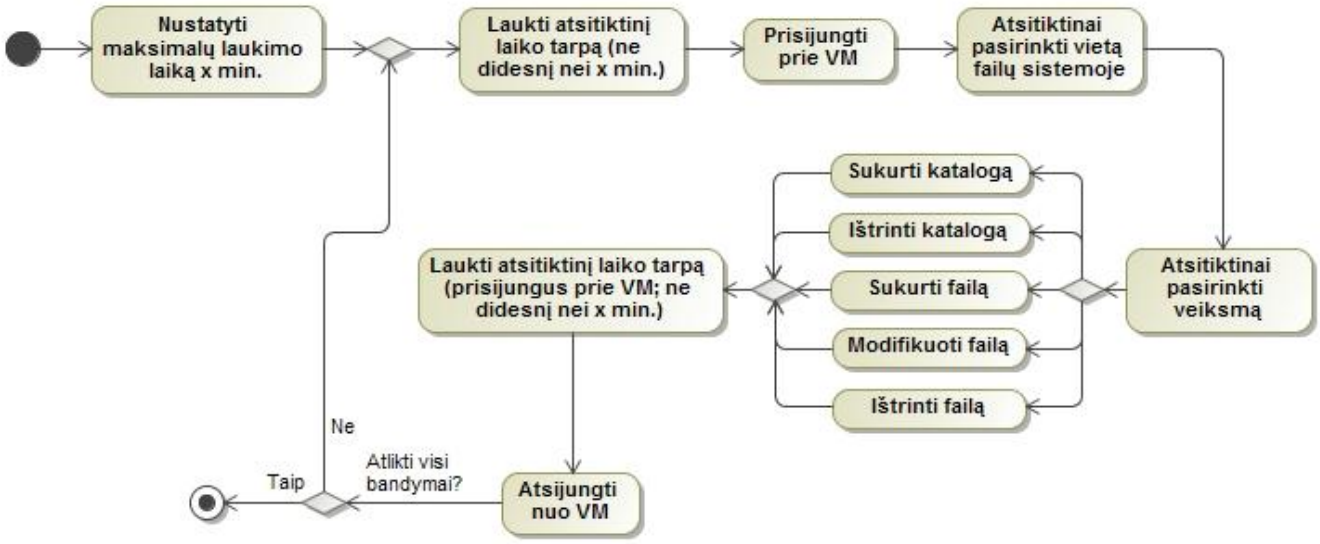
1. Paleisti vartotojų veiksmus realiu laiku imituojančius scenarijus;
2. Sukurti virtualios mašinos kopiją kas pasirinktą laiko tarpą. Viso – 10 kartų;
3. Užfiksuoti veiksmus su ŽPS priemone (po kiekvienos kopijos sukūrimo);
4. Viską pabaigus, iš sukurtų ŽPS žurnalų paruošti palyginamąją tyrimo informaciją;
5. Palyginti faktiškai atliktus vartotojų veiksmus su ŽPS sugeneruota informacija;
6. Pakeisti kopijų kūrimo laiko tarpą ir atlikti visus veiksmus nuo pradžių (3.1 pav.).



3.1 pav. Atkurtos informacijos kiekio priklausomybės nuo VM kopijų kūrimo dažnumo tyrimo schema

Kad imituoti vartotojų veiksmus realiu laiku, „Python“ programavimo kalba buvo specialiai sukurtas scenarijus **`user-imitate.py`**. Jis veikia tokiu principu:

a) Laukia atsitiktinai sugeneruotą sekundžių skaičių (nustatoma maksimali riba);
b) Prisijungia prie nurodytos virtualios mašinos;
c) Atsitiktinai pasirenka vietą egzistuojančiame failų sistemos medyje;
d) Atsitiktinai pasirenka atlikti vieną iš šių veiksmų:
    a. Sukurti naują katalogą;
    b. Ištrinti esamą katalogą;
    c. Sukurti naują failą;
    d. Modifikuoti esamą failą;
    e. Ištrinti esamą failą.
e) Laukia (vis dar prisijungęs prie VM) atsitiktinai sugeneruotą sekundžių skaičių (nustatoma maksimali riba);
f) Atsijungia nuo VM;
g) Kartoja viską iš pradžių (3.2 pav.).



3.2 pav. Vartotojų veiksmus imituojančio scenarijaus user-imitate.py veikimo schema

Šis scenarijus paleidžiamas tuo pačiu metu iš 5-ių virtualių mašinų su skirtingais IP adresais ir jungiasi į tą pačią pagrindinę VM (jos parametrai: 5 GB HDD, 256 MB RAM), kuri bus tiriama. Prisijungimas vyksta SSH protokolo pagalba, veiksmai atliekami pasinaudojant standartines „Linux“ komandinės eilutės priemones: „**`mkdir`**“, „**`touch`**“, „**`echo`**“ ir kt. Tyrimo aiškumo vardan, visi

veiksmai atliekami šakniniu katalogu pasirinkus „**/home/nerijus/data/**“.

Kad būtų galima palyginti, scenarijus **user-imitate.py** sukuria 2-u žurnalo failus, kuriuose atitinkamai užfiksuojami visi faktiškai atlikti veiksmai su failų sistemos objektais bei įvykę IP prisijungimai (žr. 3.3 ir 3.4 pav.). Šiuos žurnalus vėliau naudoja scenarijus **zps-compare.py**.

```
1358303434,./data/folder1/,CREATED
1358303438,./data/folder1/file3,CREATED
1358303441,./data/folder1/file3,MODIFIED
1358303444,./data/folder1/folder4/,CREATED
1358303447,./data/folder1/file3,DELETED
```

3.3 pav. Scenarijaus `user-imitate.py` užfiksuoti faktiškai atlikti veiksmai

```
10.0.2.2,1358303432,1358303434
10.0.2.2,1358303434,1358303438
10.0.2.2,1358303438,1358303441
```

3.4 pav. Scenarijaus `user-imitate.py` užfiksuoti faktiški IP prisijungimai (pradžios ir pabaigos laikai)

Virtualių mašinų kopijos kas nustatytą laiko tarpą buvo kuriamos rankiniu būdu pasinaudojant virtualizacijos platformos „VirtualBox“ tam tikslui skirtu funkcionalumu ir talpinamos į pasirinktą katalogą tyrime naudojamame kompiuteryje, iš kurio buvo automatiškai apdorojamos iš anksto paleistos vykdymui Žurnalizavimo Paramos Sistemos. VM kopijos sukūrimui yra reikalingi sekantys žingsniai:

a) Rankiniu būdu sukurti VM kopiją „VirtualBox“ grafinės vartotojo sąsajos pagalba;
b) Rankiniu būdu ištrinti VM kopiją „VirtualBox“ grafinės vartotojo sąsajos pagalba;
c) Konvertuoti virtualų HDD iš VMDK į „gryną“ formatą įrankio **qemu-img** pagalba;
d) Sukurti RAM kopiją scenarijaus **make-ram-copy.sh** pagalba.

Ištrynimas reikalingas tam, kad visi atskiri virtualaus HDD failai būtų sulieti į vieną. Scenarijus **make-ram-copy.sh** yra sukurtas komandinės terpės (angl. *command shell*) „bash“ scenarijaus kūrimo galimybėmis. Jis virtualios RAM kopijos gavimui panaudoja įrankio „LiME“ funkcionalumą.

Po virtualios mašinos kopijos sukūrimo gauti 2-u failai **hdd-copy.raw** ir **ram-copy.lime** toliau apdorojami anksčiau aptarto scenarijaus **zps-snap.py** pagalba. Taip ŽPS priemonė užfiksuoja tarpinę informaciją, kurią po visų 10-ies VM kopijų išanalizavimo į palyginamąją tyrimo informaciją paverčia scenarijus **zps-log.py**.

Galiausiai iškvietus scenarijų **zps-compare.py** yra palyginama ŽPS žurnaluose užfiksuota informacija su faktiškais veiksmais, kuriuos atliko scenarijus **user-imitate.py**. Gauta išvestis,

parodyta 3.5 pav., ir yra reikalingi duomenys tyrimui. Atlikus 10 bandymų su kiekvienu pasirinktu kopijų kūrimo laiko periodu, buvo apskaičiuojamas atkurtos informacijos kiekio vidurkis.

```
FS entries found: 25 of 50 (50.00%)
IP entries found: 2 of 20 (1.00%)
```

3.5 pav. Scenarijaus `zps-compare.py` išvestis

Tiriant atkurtos informacijos kiekio priklausomybę nuo VM kopijų kūrimo dažnumo, kopijos buvo daromos kas 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55 ir 60 min. Tuo tarpu siekiant nustatyti, kaip atkurtos informacijos kiekis priklauso nuo vartotojų veiksmų kiekio, buvo nustatytas 5 min. periodas, o scenarijus **`user-imitate.py`** buvo koreguojamas po kiekvienos kopijos sukūrimo iteracijos taip, kad atliekamų veiksmų skaičius didėtų. Tai buvo įgyvendinta keičiant maksimalų laukimo laiko tarpą į atitinkamai 15, 13, 10, 9, 8, 7, 6, 5, 3 ir 1 min. Atkurtų veiksmų kiekis buvo skaičiuojamas po kiekvieno atlikto bandymo.

Tyrimas likusiais pjūviais, kaip ŽPS veikimo trukmė priklauso nuo virtualaus HDD ir RAM dydžių, buvo atliekamas nustačius 5 min. kopijų kūrimo ir 15 min. maksimalų vartotojų laukimo periodus. 3.4 lentelėje pateikiami skaitiniai dydžiai, kurie buvo naudojami tyrimo metu.

3.4 lentelė. Tyrime naudoti HDD ir RAM dydžiai

Virtualaus HDD dydis									
4 GB	4,5 GB	5 GB	5,5 GB	6 GB	6,5 GB	7 GB	7,5 GB	8 GB	8,5 GB
Virtualios RAM dydis									
256 MB	320 MB	384 MB	448 MB	512 MB	576 MB	640 MB	704 MB	768 MB	834 MB

Kiekviename iš šių dviejų tyrimų po kiekvienos kopijos sukūrimo buvo skaičiuojamas pagrindinio scenarijaus **`zps-snap.py`** veikimo laikas, tada „VirtualBox“ grafinės sąsajos pagalba keičiamas HDD arba RAM dydis, ir vėl viskas kartojama iš naujo. Taip gautos visos reikiamos veikimo trukmės skaitinės reikšmės, toliau atvaizduotos grafikuose.

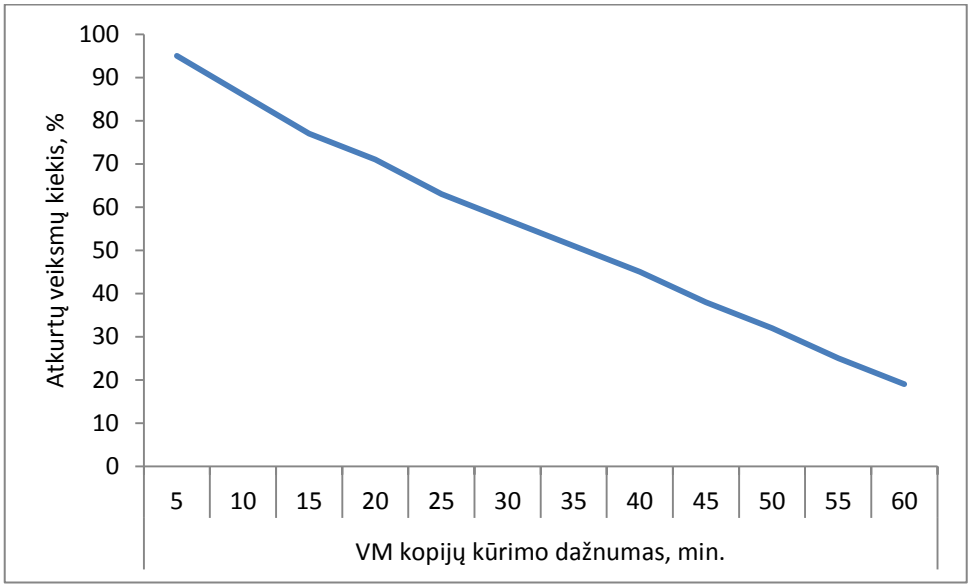
Ši aptarta tyrimo metodika sudarė palankias galimybes ištirti siūlomos Žurnalizavimo Paramos Sistemos efektyvumą.

3.2.3. Rezultatai

3.5 lent. ir 3.6 pav. yra pateikti su ŽPS atkurtos informacijos kiekio priklausomybės nuo virtualių mašinų kopijų kūrimo dažnumo tyrimo rezultatai. Juose atsispindi tiesinė priklausomybė: kuo rečiau yra kuriamos ir su ŽPS analizuojamos VM kopijos, tuo mažiau veiksmų pavyksta atkurti. Kopijas kuriant kas 5 min., pavyksta atkurti net 95% vartotojų veiksmų.

3.5 lentelė. Atkurtos informacijos kiekio priklausomybė nuo VM kopijų kūrimo dažnumo

VM kopijų kūrimo dažnumas, min.	5	10	15	20	25	30	35	40	45	50	55	60
Atkurtų veiksmų skaičius, %	95	86	77	71	63	57	51	45	38	32	25	19

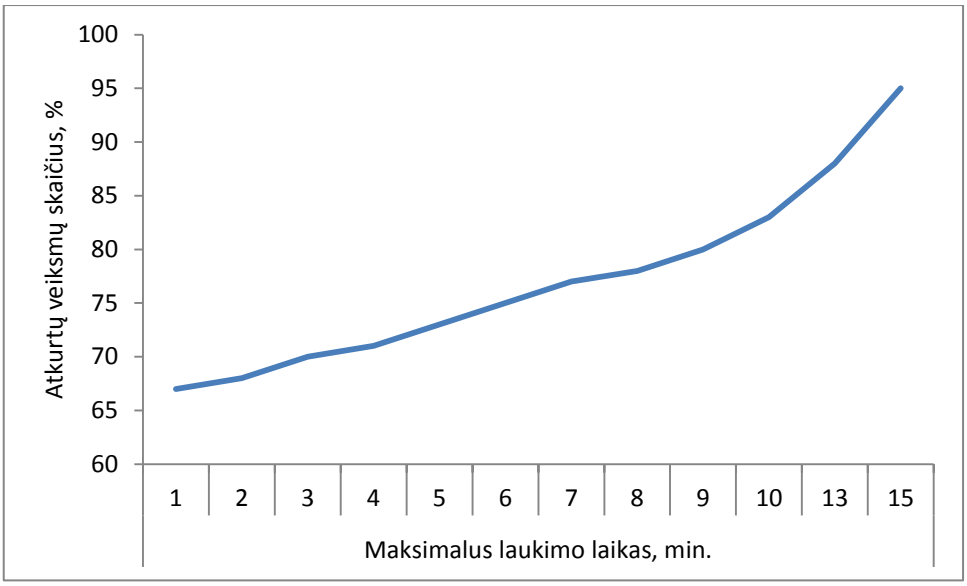


3.6 pav. Atkurtos informacijos kiekio priklausomybė nuo VM kopijų kūrimo dažnumo

Tuo tarpu priklausomybė tarp vartotojų veiksmų ir atkurtos informacijos kiekių yra eksponentinio pobūdžio – net ir stipriai didinant veiksmų kiekį, ŽPS sugeba atkurti pakankamai daug informacijos: daugiau kaip 65%; su sąlyga, kad VM kopijos yra kuriamos kas 5 min. (3.6 lent., 3.7 pav.).

3.6 lentelė. Atkurtos informacijos kiekio priklausomybė nuo maksimalaus vartotojų laukimo laiko (sąlygojančio vartotojų veiksmų kiekį)

Maksimalus laukimo laikas, min.	1	2	3	4	5	6	7	8	9	10	13	15
Atkurtų veiksmų skaičius, %	67	68	70	71	73	75	77	78	80	83	88	95

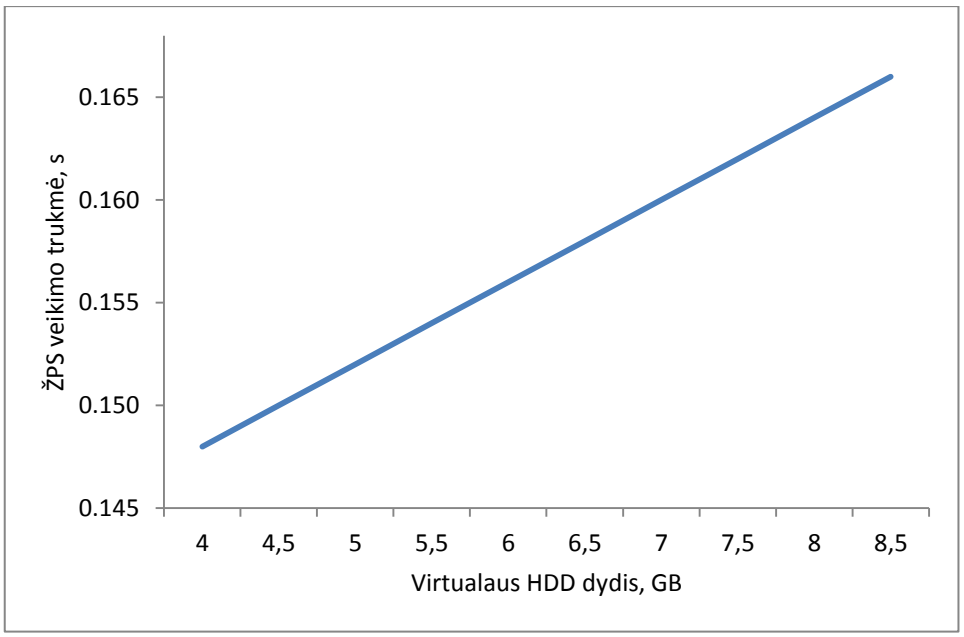


3.7 pav. Atkurtos informacijos kiekio priklausomybė nuo maksimalaus vartotojų laukimo laiko (sąlygojančio vartotojų veiksmų kiekį)

Svarbus Žurnalizavimo Paramos Sistemos efektyvumo rodiklis – duomenų apdorojimo greitis priklausomai nuo jų dydžio. Atlikus tyrimus su skirtingais virtualių mašinų HDD dydžiais, paaiškėjo, kad sistema veikia itin sparčiai: 8 GB dydžio duomenys buvo apdoroti vos per 164 milisekundes, ir tai yra vos 20 milisekundžių daugiau negu apdorojant du kartus mažiau talpų 4 GB dydžio HDD. Tikslios tyrimo metu gautos skaitinės reikšmės pateikiamos 3.7 lent., o 3.8 pav. jos atvaizduotos grafiškai.

3.7 lentelė. ŽPS veikimo trukmės priklausomybė nuo virtualaus HDD dydžio

Virtualaus HDD dydis, GB	4	4,5	5	5,5	6	6,5	7	7,5	8	8,5
ŽPS veikimo trukmė, s	0,148	0,150	0,152	0,154	0,156	0,158	0,160	0,162	0,164	0,166

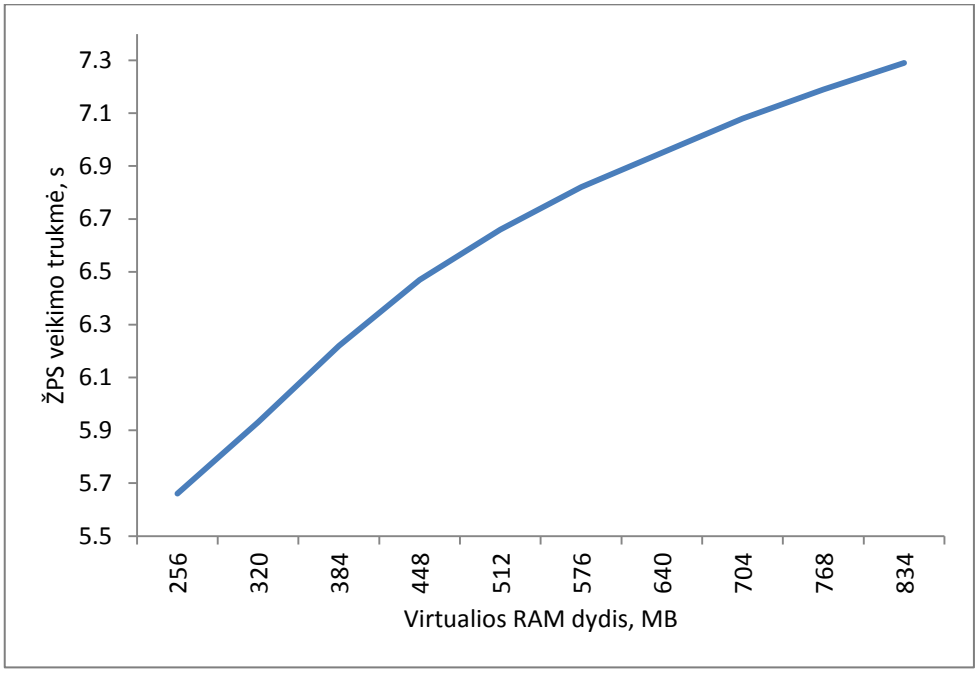


3.8 pav. ŽPS veikimo trukmės priklausomybė nuo virtualaus HDD dydžio

ŽPS sistemos dalis, apdorojanti su virtualia RAM susijusius duomenis, veikia lėčiau, tačiau pakankamai greitai. Analizuojant jos veikimo greičio priklausomybę nuo skirtingų RAM dydžių, paaiškėjo, kad sistema užtruko maždaug 5,5 s savęs inicializavimui, po kurio duomenų apdorojimas papildomai pridėjo po 1 s už kiekvienus papildomus 400 MB RAM atminties. Rezultatai pateikiami 3.8 lent. ir 3.9 pav.

3.8 lentelė. ŽPS veikimo trukmės priklausomybė nuo virtualios RAM dydžio

Virtualios RAM dydis, MB	256	320	384	448	512	576	640	704	768	834
ŽPS veikimo trukmė, s	5,66	5,93	6,22	6,47	6,66	6,82	6,95	7,08	7,19	7,29



3.9 pav. ŽPS veikimo trukmės priklausomybė nuo virtualios RAM dydžio

3.3.Išvados

Apibendrindami eksperimentinį tyrimą, galime padaryti tokias išvadas:

- ✓ Kad ištirti siūlomą metodiką, buvo pasinaudota virtualizacijos platforma „VirtualBox“ ir jos teikiamu „aktyvių“ virtualių mašinų kopijų kūrimo funkcionalumu, o vartotojų veiksmai su failų sistemos objektais buvo imituojami specialiai tam tikslui suprogramuotu scenarijumi;
- ✓ Žurnalizavimo Paramos Sistema parodė aukštus efektyvumo rodiklius: kiekvienoje iteracijoje užtrukdama vos 5,5-7,5 s, ji sugebėjo atkurti ne mažiau kaip 65% veiksmų priklausomai nuo vartotojų aktyvumo, kai virtualių mašinų kopijos būdavo kuriamos ir analizuojamos ne rečiau kaip kas 5 min.
- ✓ Esant sąlyginai nedideliam vartotojų aktyvumui ir kuriant virtualių mašinų kopijas kas 5 min., su Žurnalizavimo Paramos Sistema pavykdavo atkurti net 95% veiksmų.

4. IŠVADOS

Išanalizavus skaitmeninės teismo ekspertizės atlikimo skaičiavimų debesyje problemą ir pasiūlius bei ištyrus ją patobulinančią metodiką, galima padaryti tokias išvadas:

- ✓ Šiai dienai egzistuoja akivaizdi skaitmeninės teismo ekspertizės atlikimo skaičiavimų debesyje problema, ir yra siūloma įvairių metodų jai spręsti;
- ✓ Pristatyta metodika-įrankis, Žurnalizavimo Paramos Sistema, padeda užfiksuoti ir vėliau atkurti vartotojų veiksmus kaip įkalčius skaičiavimų debesies saugyklose, nepriklausomai nuo to, ar šiuo klausimu debesies paslaugos tiekėjas siūlo kokį nors funkcionalumą (paprastai jis neegzistuoja išvis);
- ✓ ŽPS yra pagrįsta „Python“ programavimo kalba apjungiant dvi nemokamas atvirojo kodo programines priemones, skirtas skaitmeninės teismo eskpertizės atlikimui – „The Sleuth Kit“ ir „The Volatility Framework“: TSK pagalba iš virtualių mašinų kopijų yra išgauna visa reikalinga su fizine failų sistema susijusi informacija, o TVF suteikia galimybes surasti reikiamus artefaktus operatyviojoje atmintyje;
- ✓ ŽPS įgyvendina kitų autorių pasiūlytą unifikuotą audito įrašų kūrimo formatą tokio pobūdžio aplinkoms ir sukuria taip vadinamą save apsirašančių duomenų efektą, kuris, manoma, yra būtinas žingsnis siekiant efektyviai tirti nusikaltimus skaičiavimų debesyje;
- ✓ Tyrimo metu ŽPS pademonstravo aukštą veikimo efektyvumą: kiekvienoje iteracijoje užtrukdama vos 5,5-7,5 s, ji sugebėjo atkurti ne mažiau kaip 65% veiksmų priklausomai nuo vartotojų aktyvumo, kai virtualių mašinų kopijos būdavo kuriamos ir analizuojamos ne rečiau kaip kas 5 min.;
- ✓ Esant sąlyginai nedideliam vartotojų aktyvumui ir kuriant virtualių mašinų kopijas kas 5 min., su Žurnalizavimo Paramos Sistema pavykdavo atkurti net 95% veiksmų;
- ✓ Šie magistrinio darbo rezultatai buvo pristatyti 18-toje tarpuniversitetinėje magistrantų ir doktorantų konferencijoje „Informacinė visuomenė ir universitetinės studijos (IVUS 2013)“, kurios iniciatyva taip pat buvo išpublikuotas šio magistrinio darbo pasekoje parašytas autorių straipsnis: „Nusikalstamos veiklos „skaičiavimų debesies“ saugyklose atkūrimo metodas“. Jo kopija pateikta A priede.

Ateityje būtų naudinga ištirti Žurnalizavimo Paramos Sistemos galimybes ir efektyvumą dirbant su milžiniško dydžio duomenų kiekiais (skaičiuojant terabaitais), kurie paprastai sutinkami skaičiavimų debesies technologijos kontekste.

LITERATŪRA

1. D. BIRK. Technical Challenges of Forensic Investigations in Cloud Computing Environments. Systematic Approaches to Digital Forensic Engineering (SADFE), 2011, IEEE Sixth International Workshop.
2. R. CHOW et al. Controlling Data in the Cloud: Outsourcing Computation without Outsourcing Control. CCSW'09, November 13, 2009, Chicago, Illinois, USA, Copyright 2009 ACM.
3. J. C. F. CRUZ, T. ATKINSON. Digital Forensics on a Virtual Machine. 49th ACM Southeast Conference, March 24-26, 2011, Kennesaw, GA, USA, 2011 ACM.
4. J. C. F. CRUZ, T. ATKINSON. Evolution of Traditional Digital Forensics in Virtualization. ACM Southeast Conference, March 29-31, 2012, Tuscaloosa, AL, USA, 2012 ACM.
5. X. FU, Zh. LING, W. YU, J. LUO. Cyber Crime Scene Investigations (C2SI) through Cloud Computing: Proceedings of the 2010 IEEE 30th International Conference on Distributed Computing Systems Workshops.
6. S. L. GARFINKEL. Digital Forensics Research: The Next 10 Years". Elsevier Ltd. Digital Investigation Magazine, No 7, 2010.
7. B. GROBAUER, Th. SCHRECK. Towards Incident Handling in the Cloud: Challenges and Approaches. CCSW'10, October 8, 2010, Chicago, Illinois, USA, Copyright 2010 ACM.
8. R. K. L. KO et al. Trustcloud: Framework for Accountability and Trust in Cloud Computing, 2011 IEEE.
9. R. LU, X. LIN, X. LIANG, X. Sh. SHEN. Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing. ASIACCS'10 April 13–16, 2010, Beijing, China, 2010 ACM.
10. S. MARSTON et al. Cloud Computing - The Business Perspective. Decision Support Systems 51, 2011.
11. R. MARTY. Cloud Application Logging for Forensics. SAC'11 March 21-25, 2011, TaiChung, Taiwan, Copyright 2011 ACM.
12. D. REILLY, C. WREN, T. BERRY. Cloud Computing: Pros and Cons for Computer Forensic Investigations. International Journal Multimedia and Image Processing (IJMIP), Volume 1, Issue 1, March 2011.
13. K. RUAN, J. CARTHY, T. KECHADI, M. CROSBIE. Cloud Forensics: An Overview. Advances in Digital Forensics, 7th IFIP International Conference on Digital Forensics, Orlando, Florida, June 2011.
14. M. TAYLOR, J. HAGGERTY, D. GRESTY, R. HEGARTY. Digital Evidence in Cloud Computing Systems. Computer Law & Security Review 26, 2010, p. 304-308.
15. C. WANG, Q. WANG, K. REN, W. LOU. Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing. IEEE INFOCOM 2010.

16. S. D. WOLTHUSEN. Overcast: Forensic Discovery in Cloud Environments. 2009 Fifth International Conference on IT Security Incident Management and IT Forensics, 2009 IEEE.
17. A. A. YAVUZ, P. NING. BAF: An Efficient Publicly Verifiable Secure Audit Logging Scheme for Distributed Systems. 2009 IEEE.
18. W. ZHOU et al. Towards Data-Centric View of Cloud Security. CloudDB 2010, October 30, 2010, Toronto, Ontario, Canada, 2010 ACM.
19. Y. ZHU, H. WANG, Z. HU, G. AHN, H. HU, S. S. YAU. Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds. SAC'11 March 21-25, 2011, TaiChung, Taiwan, 2011 ACM.
20. Access US [žiūrėta 2013-05-13]. Prieiga per internetą: <http://www.accessus.net/business-services/cloud-computing>.
21. AccessData [žiūrėta 2013-05-13]. Prieiga per internetą: <http://www.accessdata.com/products/digital-forensics/ftk>.
22. Cisco Global Cloud Index: Forecast and Methodology, 2010–2016, White Paper [žiūrėta 2013-05-13]. Prieiga per internetą: <http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns1175/Cloud_Index_White_Paper.html>.
23. Guidance Software [žiūrėta 2013-05-13]. Prieiga per internetą: <http://www.guidancesoftware.com>.
24. P. MELL, T. GRANCE. The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standards and Technology, Special Publication 800-145, September 2011 [žiūrėta 2013-05-13]. Prieiga per internetą: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>.
25. K. KENT, S. CHEVALIER, T. GRANCE, H. DANG. Guide to Integrating Forensic Techniques into Incident Response: Recommendations of the National Institute of Standards and Technology, Special Publication 800-86, August 2006 [žiūrėta 2013-05-13]. Prieiga per internetą: <http://csrc.nist.gov/publications/nistpubs/800-86/SP800-86.pdf>.
26. The Sleuth Kit [žiūrėta 2013-05-13]. Prieiga per internetą: <http://www.sleuthkit.org>.
27. The Volatility Framework [žiūrėta 2013-05-13]. Prieiga per internetą: <https://code.google.com/p/volatility>.

PRIEDAI

Priedas A. Išpublikuoto straipsnio „Nusikalstamos veiklos „skaičiavimų debesies“ saugyklose atkūrimo metodas“ kopija

INFORMACINĖS TECHNOLOGIJOS • IVUS 2013 • ISSN 2029-249X • eISSN 2029-4824

Nusikalstamos veiklos „skaičiavimų debesies“ saugyklose atkūrimo metodas

Nerijus Saikauskas
Kauno technologijos universitetas
Kompiuterių katedra, Studentų g. 50
Kaunas, Lietuva
nerijus.saikauskas@stud.ktu.lt

Santrauka. ***„Skaičiavimų debesies“ (angl. cloud computing) technologijos sukūrimas suteikė galimybę padidinti kompanijų veiklos efektyvumą, tačiau sukėlė ir naujų problemų, viena kurių – skaitmeninės teismo ekspertizės (angl. digital forensics) atlikimas nutolusioje aplinkoje. Apskritai teigiama, kad jeigu „skaičiavimų debesies“ paslauga nefiksuoja tinkamų audito įrašų, nustatyti įkalčius tampa sunku arba tiesiog neįmanoma. Deja, paprastai šiam tikslui siūlomas funkcionalumas yra gana ribotas arba iš viso neegzistuoja. Šiame straipsnyje siūlome naują metodą-įrankį – žurnalizavimo paramos sistemą (ŽPS), padedančią užfiksuoti ir atkurti vartotojų veiksmus kaip įkalčius „skaičiavimų debesies“ saugyklose. ŽPS įgyvendina kitų autorių pasiūlytą unifikuotą audito įrašų formatą tokio pobūdžio aplinkoms ir sukuria save aprašančių duomenų efektą, kuris, manoma, yra svarbus žingsnis siekiant efektyviai tirti nusikaltimus „skaičiavimų debesies“ saugyklose. Metodo efektyvumas yra įvertinamas apskaičiuojant ŽPS atkuriamų veiksmų skaičiaus priklausomybę nuo virtualių mašinų (angl. virtual machine) kopijų kūrimo dažnumo ir vartotojų atliekamų veiksmų skaičiaus.***

Reikšminiai žodžiai: „skaičiavimų debesis“, virtuali mašina, skaitmeninė teismo ekspertizė, saugykla, veiksmų atkūrimas, paramos sistema.

I. ĮŽANGA

Paskutiniais keleriais metais įvyko reikšmingų IT pasikeitimų, vienas kurių – „skaičiavimų debesies“ (angl. *cloud computing*) sukūrimas. Šios technologijos atsiradimas ėmė keisti būdus, kaip organizacijos įsigyja kompiuterijos. Patikimi tiekėjai, tokie kaip „Microsoft“, „Amazon“, „Google“, „Yahoo“ ir kiti, pasiūlė kompanijoms efektyvias galimybes reikiamų IT išteklių pirkti iš trečiųjų šalių ir tokiu būdu pagrindinį dėmesį sutelkti į savo esminį verslą [1].

„Skaičiavimų debesis“ – tai „modelis, suteikiantis visur pasiekiamą, patogią, „pagal pareikalavimą“ tinklo prieigą prie bendrų, konfigūruojamų kompiuterių išteklių (tinklų, serverių, saugyklų, vartotojo programinės įrangos, paslaugų ir kt.), kurie gali būti sparčiai kviečiami minimaliomis valdymo pastangomis ar paslaugos tiekėjo įsikišimu“ [2] (1 pav.).

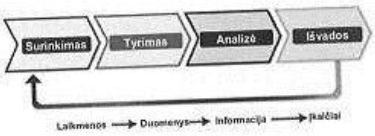
„Skaičiavimų debesies“ paslauga sparčiai populiarėja. Prognozuojama, kad metinis globalus jos IP srautas išaugs 3,5 karto – nuo 1,2 zetabaitų 2012 m. iki 4,3 zetabaitų (tai sudaro maždaug 4 600 000 000 terabaitų) 2016 m. ir sudarys du trečdalius visų duomenų centrų srauto [4].

1 pav. „Skaičiavimų debesies“ modelis [3]

Nežiūrint į privalumus, „skaičiavimų debesis“ kelia įvairių abejonių ir rūpesčių. Ypač baiminamasi dėl esamų galimybių paslaugos tiekėjui būti užpultam elektroninės erdvės nusikaltėlių, kurie gali pavogti klientų informaciją. Galima tiek tiesioginė, tiek netiesioginė (pasinaudojant kliento puse) „debesies“ infrastruktūros kompromitacija, egzistuoja įvairios „debesies“ pažeidžiamos vietos, kyla paslaugos tiekėjo šnipinėjimo grėsmė, be to, dėl nutolusių išteklių, sunkiai sprendžiamas audito ir skaitmeninės teismo ekspertizės atlikimo klausimas [5].

Skaitmeninė teismo ekspertizė (angl. *digital forensics*) 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*) (2 pav.). Per pastarąjį dešimtmetį labai išaugo nusikaltimų, atliekamų elektroninėje erdvėje, skaičius. Todėl, įsikūrė daug kompanijų, kurios siūlo produktų, padedančių teisėsaugos organams nustatyti, kas, ką, kur, kada ir kaip padarė nusikaltimus kompiuterinėmis priemonėmis. Skaitmeninė teismo ekspertizė, kaip sfera, išsivystė būtent tam, kad būtų užtikrintas tinkamas elektroninių įkalčių pristatymas teismui [6].

Straipsnio II skyriuje aprašoma veiksmų kaip įkalčių atkūrimo „skaičiavimų debesyje“ problematika, III skyriuje nusakomas mūsų siūlomas metodas – žurnalizavimo paramos



2 pav. Skaitmeninės teismo ekspertizės procesas

sistema (ŽPS), – kurią pasitelkus įvykus nusikaltimui būtų galima atkurti vartotojų veiksmus „skaičiavimų debesies“ saugyklose pasinaudojant esamomis atvirojo kodo programinėmis priemonėmis, IV skyriuje įvertinamas metodo efektyvumas ir pateikiamos išvados.

II. Veiksmų kaip įkalčių atkūrimo „skaičiavimų debesyje“ problematika

A. Skaitmeninė teismo ekspertizė „skaičiavimų debesies“ kontekste

Kyla daug klausimų, ar „skaičiavimų debesis“ gali būti tinkamai ištirtas skaitmeninės teismo ekspertizės tikslams. Viena vertus, šios technologijos tinkamumas tokioms užduotims nėra pakankamai išanalizuotas. Kita vertus, toks gebėjimas nėra esminis jos reikalavimas – kadangi tarptautinės teisinės bazės šiuo klausimu nėra, tai laikoma prabanga, kurią gali arba negali būti įsidiegę paslaugos tiekėjai savo nuožiūra [7].

Kalbant apie techninius skaitmeninės teismo ekspertizės atlikimo „skaičiavimų debesyje“ aspektus, yra galimi 3 įkalčių šaltiniai:

1) „Skaičiavimų debesies“ dalis (vadinamoji virtuali instancija), kurią naudoja klientas. Yra galimybės „gyvai“ tirti vis dar veikiančias virtualias instancijas [8, 9] arba padaryti jų kopijas tam tikru laiko momentu (angl. *snapshot*) ir išanalizuoti jas atskirai.

2) Kompiuterių tinklas. Skirtingi tinklo OSI[1] „sluoksniai“ galėtų suteikti įvairios naudingos informacijos apie juose veikiančius protokolus, kurie užtikrina duomenų srautų tarp „skaičiavimų debesies“ ir kliento apsikeitimą. Deja, tipiškas „skaičiavimų debesies“ paslaugų tiekėjas nesuteikia jokių tinklo komponentų audito įrašų (angl. *logs*).

3) Kliento sistema. Paprastai tai yra tik naršyklė, veikianti kliento kompiuteryje, kuri leidžia pasiekti visas reikiamas „skaičiavimų debesies“ paslaugas [10].

Vertinant iš skaitmeninės teismo ekspertizės pusės, pagrindinis „skaičiavimų debesies“ privalumas yra centralizuoti duomenys: vienoje vietoje esančią informaciją lengviau kontroliuoti. Kitas teigiamas dalykas – milžiniški kompiuterijos ištekliai, galintys saugoti neribotą tiriamos informacijos kiekį ir daug sparčiau atlikti sudėtingų skaičiavimų reikalaujančius veiksmus, pavyzdžiui, nulaužti slaptažodžius. Be to, „skaičiavimų debesies“ mastas suteikia neribotas galimybes sukurti ir suderinti audito įrašų kūrimo mechanizmus taip, kaip to pageidauja esama kliento situacija, kad ir kokia kompleksiška ji būtų ir reikalautų išteklių [1]. Kaip parodė [11], „skaičiavimų debesies“ paslauga gali tapti nusikaltimo tyrimo įrankiu netgi pati savaime.

Vis dėlto „skaičiavimų debesis“, vertinant jį iš skaitmeninės teismo ekspertizės pozicijų, turi ir esminių trūkumų, kurių pagrindinis – nutolę duomenys, kurių fizinė buvimo vieta nėra tiksliai žinoma [12]. Tiriant tokiomis aplinkybėmis, yra didelė tikimybė prarasti svarbius nusikaltimo artefaktus. Pavyzdžiui, gali nebūti galimybių pasiekti „skaičiavimų debesies“ duomenų centruose esančius registro įrašus (angl. *registry entries*), laikinuosius failus (angl. *temporary files*) ir laikinąją atmintį (angl. *memory*). Jeigu duomenys yra atsisiunčiami iš „skaičiavimų debesies“, gali būti prarasta ir metainformacija.

Yra ir daugybė kitų problemų, susijusių su teismo ekspertizės atlikimu „skaičiavimų debesyje“, tokių kaip įrankių nebuvimas [1], teisiniai aspektai (duomenų centrų išsidėstymas keliose šalyse kelia jurisdikcijos klausimą), įkalčių stabilumas ir kitos, kurias aktyviai siūloma spręsti mokslininkų bendruomenei [13]. Tai yra vis dar besivystanti sritis, ir aktyviai kuriami įvairūs metodai jai patobulinti [1].

B. Audito įrašai kaip priemonė atkurti veiksmus

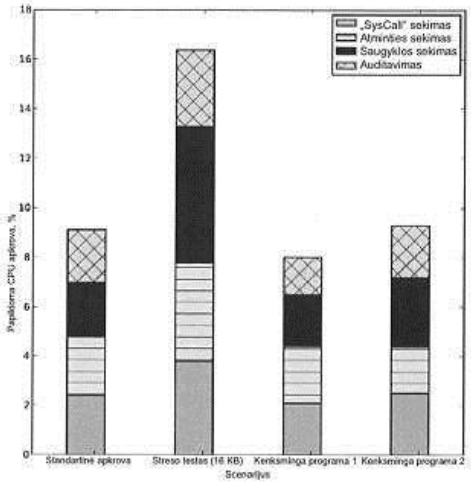
Jeigu „skaičiavimų debesies“ paslauga nefiksuoja tinkamų audito įrašų, nustatyti įkalčius tampa sunku arba tiesiog neįmanoma [7]. Šiam klausimui skiriama daugiausiai dėmesio.

Siūloma pereiti nuo apsaugos, kuri veikia iš sistemos perspektyvos, prie efektyvesnės, kuri veikia atskaitos taškų pasirinkdama failą. Tai reiškia, kad duomenys, saugomi „skaičiavimų debesyje“, turi būti patys save apibūdinantys, kad būtų galima atsekti visą veiklą nuo jų sukūrimo iki sunaikinimo, nepriklausomai nuo aplinkos apribojimų [14]. W. Zhou ir kt. [15] apibūdina pagrindinius iššūkius, susijusius su šios idėjos, dar vadinamos duomenų centriškumu (angl. *data-centric view*), įgyvendinimu, ir siūlo galimą platformą (angl. *framework*) apsikeisti tokia informacija „skaičiavimų debesyje“ tarp skirtingų sistemų, kuri padėtų išsaugoti reikiamus metaduomenis.

Pabrėžiama, kad dabartinių pasyvių „skaičiavimų debesies“ apsaugos metodų nepakanka siekiant atsekti su pačiais save apibūdinančiais duomenimis susijusius veiksmus. Todėl būtina aktyviai daryti audito įrašus, tenkinant tokius reikalavimus:

- *Sekti failus.* Stebimi sisteminiai failų sistemos skaitymo / rašymo kreipiniai (angl. *call*) padėtų atsekti ir susieti virtualias ir fizines failų atminties vietas. (Tyrimais įrodyta, kad dėl šios papildomos funkcijos virtualizacijos platformos procesoriaus apkrova padidėtų iki 20 % (3 pav.) [16].) Failų vietos keitimo sekimas (įskaitant tinklo audito įrašus) sugeneruotų papildomos naudingos informacijos skaitmeninės teismo ekspertizės tyrėjui.
- *Sekti duomenis.* Šie dažnai būna išsidėstę keletame failų sistemos objektų (failų, katalogų), tad integralesnis (bendresnis) audito įrašų darymas padėtų sukaupti reikiamų įrodymų platesniame kontekste.

[1] Atvirųjų sistemų sujungimo tinklu modelis, apibrėžiantis duomenų perdavimą atviraisiais nevienalyčiais tinklais [24].

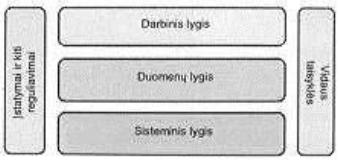


3 pav. Procesoriaus apkrovos padidėjimas papildomai apdorojant sisteminius failų sistemos skaitymo / rašymo kreipinius skaitmeninės teismo ekspertizės tikslams

- *Sekti informaciją.* Kadangi būtent informacija, o ne duomenys ar failai yra svarbiausia organizacijos vertybė, užfiksuoti įrašai, parodantys, kaip failai ir duomenys transformavosi į informaciją, yra neįkainojamos vertės tiriant saugumo incidentus.
- *Sekti informaciją ir duomenų srautus.* Aukšto lygio rizika, tokia kaip sprendimų priėmimo informacija, taip pat turėtų būti kontroliuojama.

Šiuos reikalavimus autoriai išdėstė vadinamojo „TrustCloud" koncepcijoje, kuri vaizdžiai apibrėžia centralizuotą požiūrį į audito įrašų darymą (4 pav.) [14].

Sunkumų kelia tai, kad nėra oficialaus standarto, kokiu formatu įrašai turėtų būti fiksuojami siekiant vėliau efektyviai ištirti nusikaltimą „skaičiavimų debesyje". R. Marty [17] siūlo priimti bendras gaires, kuriose apibrėžiama konkreti sintaksė. Autorius mano, kad audito įrašais turėtų būti fiksuojamas bent įvykio laikas, susijusi aplikacija, vartotojas, sesijos ID, svarbumo reitingas (angl. *severity*), priežastis ir kategorija, nes tai padėtų atsakyti į klausimus: kada, kas, ką ir kodėl. O įrašų sintaksę siūloma grįsti laukų grupavimu (angl. *key-value pair*), nes tai leistų efektyviai apdoroti audito įrašus papildomomis programinėmis priemonėmis (5 pav.).



4 pav. „TrustCloud" koncepcija

```
time=2010-05-13 13:03:47.123231PDT,
session_id=08BaswoAAQgAADVDG3IAAAAD,
severity=ERROR,user=pixlcloud_zrlram,
object=customer,action=delete,status=failure,
reason=does not exist
```

5 pav. Siūloma sintaksė audito įrašų „skaičiavimų debesyje" fiksavimui

„Skaičiavimų debesies" audito įrašai negalės būti panaudoti teismo procese, jeigu nebus užtikrintas jų integralumas. Ši problema, iškilusi „skaičiavimų debesies" saugyklų kontekste, yra sprendžiama siūlymais įtraukti trečiąją šalį, kuri, apsikeisdama specialiomis žinutėmis su vartotoju ir „skaičiavimų debesimi", papildomai išsaugotų ir įrašų privatumą [18]. Tuo užsiimti, manoma, galėtų oficialiai pripažinta audito kompanija (6 pav.), tačiau [20] bei [21] nurodoma ir metodų, kurie užtikrina paskirstytose sistemose (angl. *distributed systems*) esančių įrašų integralumą, neįsikišant pašaliniams.

III. ŽURNALIZAVIMO PARAMOS SISTEMA

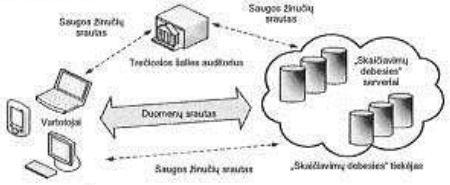
Siūlome naują metodą-įrankį – *žurnalizavimo paramos sistemą*, kuri, pasinaudodama atvirojo kodo priemonėmis, padeda pirmiausia užfiksuoti, o vėliau – atkurti vartotojų veiksmus kaip įkalčius „skaičiavimų debesies" saugyklose. ŽPS skirta fiksuoti nustatytiems duomenims, nepriklausomai nuo to, ar egzistuoja koks nors „skaičiavimų debesies" saugyklų paslaugos funkcionalumas šiuo klausimu, ar ne. ŽPS vieta nagrinėjamos problemos kontekste pavaizduota 7 pav.

Sistema suprojektuota taip, kad veiktų kaip atskiras komponentas, savarankiška programinė įranga, kurią savo pasirinkimu galėtų įdiegti bet kokia privati kompanija, kuri naudojasi lokaliomis „skaičiavimų debesies" saugyklomis. Ją funkcionalumo prasme sudaro 2 dalys:

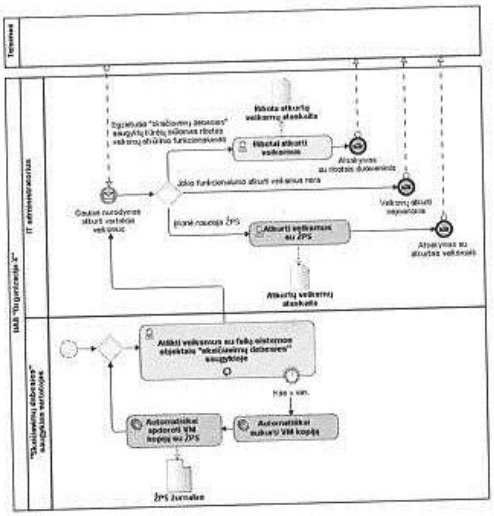
1) Veiksmus fiksuojanti. Apdoroja naujas virtualios mašinos (angl. *virtual machine*) kopijas ir papildo ŽPS audito žurnalą.

2) Veiksmus atkurianti. Atkuria pasirinkto laikotarpio veiksmus.

Tiek veiksmus fiksuojanti, tiek veiksmus atkurianti dalis gali būti įdiegta į bet kokią pasirinktą „Windows" arba „Linux" kompiuterinę sistemą pagal patogumą – svarbiausia, kad iš jų būtų galima pasiekti paruoštas VM kopijas ir vėliau ŽPS sukurtą audito žurnalą (norint atkurti veiksmus). Suteikiamas tinkamai paruoštų ir į nustatytą katalogą įdėtų „aktyvių" virtualių mašinų kopijų (angl. *snapshot*) apdorojimo funkcionalumas, atsiribojant nuo kopijų kūrimo klausimo ir su juo susijusios problematikos. VM kopijos turi tenkinti šiuos reikalavimus:



6 pav. Audito įrašų integralumo užtikrinimas pasinaudojant trečiąja šalimi [19]



7 pav. Žurnalizavimo paramos sistemos vieta problemos kontekste (BPMN modelis)

- Turi būti 2-u failai – virtualaus HDD kopija ir virtualios RAM kopija.
- Tiek HDD, tiek RAM kopija turi būti išsaugota (esant reikalui, konvertuota) vadinamuoju „grynu“ (angl. *raw*) formatu.

Žurnalizavimo paramos sistema yra pagrįsta „Python“ programavimo kalba jungiant dvi nemokamas atvirojo kodo programines priemones: „The Sleuth Kit“ (TSK) [22] ir „The Volatility Framework“ (TVF) [23]. TSK leidžia iš virtualaus HDD nuskaityti šiuos failų sistemos objektų (failų ir katalogų) metaduomenis:

- sukūrimo data;
- redagavimo data;
- ištrynimo data.

TVF suteikia galimybę surasti reikiamus artefaktus operatyviojoje atmintyje. ŽPS atveju iš virtualios RAM kopijos yra nuskaitomi prie VM esamu momentu prisijungę IP adresai. Virtualių mašinų kopijos yra kuriamos nustatytu periodu (žr. IV skyrių), ir tiek TSK, tiek TVF sugeneruoti duomenys išsaugomi ŽPS audito žurnaluose, kurių formatas atitinka [17] reikalavimus ir kuriuose dėl patogumo naudojamas „Unix“ laiko formatas (8, 9 pav.).

```
time=1358303434,object=./data/folder1/,action=created
time=1358303438,object=./data/folder1/file3,action=created
time=1358303441,object=./data/folder1/file3,action=modified
time=1358303444,object=./data/folder1/folder4/,action=created
time=1358303447,object=./data/folder1/file3,action=deleted
```

8 pav. Audito žurnalo „zps-fs.log“ turinys

```
time=1358269391,object=[10.0.2.2,65,7,11,15]
time=1358293119,object=[192.168.14.22]
time=1358299579,object=[10.0.2.15,192.168.0.14,213.15.26.5]
```

9 pav. Audito žurnalo „zps-ip.log“ turinys

Kadangi „The Sleuth Kit“ nuskaito failų sistemos objektų metaduomenis, visais atvejais yra pateikiama galutinė ir užtikrinta informacija apie su objektais atliktus veiksmus. Turint omenyje, kad vartotojams paprastai yra suteikiama prieiga ne prie absoliučiai visų, o tik specialiai tam skirtų failų sistemos išteklių (atskirų skaidinių arba katalogų), žurnalizavimo paramos sistemos darbas yra optimizuojamas konfigūracijoje nurodant atitinkamas failų sistemos dalis, kurių veiksmus reikia fiksuoti. O „The Volatility Framework“ išsaugo visus prie VM prisijungusius tinklo IP adresus, nepriklausomai nuo to, ar jie gali būti susiję su TSK dalies užfiksuotais failų sistemos objektų pokyčiais, ir pateikia tai kaip papildomą informaciją bylos nagrinėtojams. Aptarta ŽPS veikimo schema pateikiama 10 pav.

Veiksmams atkurti ŽPS įrankis nuskaito audito žurnalą, kuriame yra išsaugota ŽPS surinkta informacija. Galima surasti įrašus pagal:

- datą;
- IP adresą;
- failų sistemos objektą;
- veiksmą (sukūrimas, redagavimas, ištrynimas).

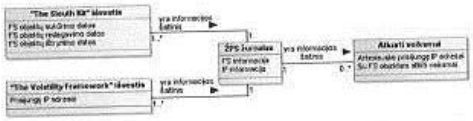
11 pav. pateiktas atkurtų veiksmų, susijusių su failu „file1“, audito įrašų pavyzdys. Išvestyje įrašomi prieš failų sistemos objektų pokyčius arčiausiai užfiksuoti IP adresai. Gauname atkuriamąją informaciją apie objektą, nepriklausomai nuo to, kokiose sistemose ar priemonėse jis buvo naudojamas. Tokiu būdu įgyvendiname [14] ir [15] pasiūlytą save aprašančių duomenų koncepciją, kuri, kaip buvo aptarta II skyriuje, yra svarbus žingsnis siekiant efektyviai tirti nusikaltimus „skaičiavimų debesies“ saugyklose.

Kitame skyriuje pateikiamas žurnalizavimo paramos sistemos efektyvumo įvertinimas.

IV. ŽPS EFEKTYVUMO ĮVERTINIMAS

Siūlomas ŽPS metodas-įrankis buvo tiriamas šiais aspektais:

- Atkurtos informacijos kiekio priklausomybė nuo VM kopijų kūrimo (ir atitinkamai apdorojimo pasitelkiant žurnalizavimo paramos sistemą) dažnumo.



10 pav. ŽPS veikimo schema (koncepcinis duomenų modelis)

```
time=1358225932,object=[10.0.2.15]
time=1358228301,object=./data/file1,action=created
time=1358229629,object=./data/file1,action=modified
time=1358233118,object=[10.0.2.15,192.168.0.147,213.15.26.5]
time=1358234629,object=./data/file1,action=modified
time=1358234660,object=./data/file1,action=deleted
```

11 pav. Atkurti veiksmai, susiję su failų sistemos objektu „file1"

- Atkurtos informacijos kiekio priklausomybė nuo vartotojų veiksmų dažnumo.

Buvo naudojamas kompiuteris „HP nx6125" (I lent.) su „Lubuntu 12.10" 64 bitų operacine sistema („Linux" branduolio versija – „3.5.0-21-generic"), kurioje įdiegta „Oracle VirtualBox 4.1.18" virtualizacijos platforma. Sukurtos 6-ios virtualios mašinos su „Debian 4.3.5-4" 32 bitų operacine sistema („Linux" branduolio versija – „2.6.32-5-686"). Viena buvo naudojama kaip tyrimo objektas – būtent jos kopijos ir buvo kuriamos bei analizuojamos, o likusios buvo panaudotos kaip skirtingų 5-ių vartotojų kompiuterių imitavimas, kuriais buvo jungiamasi prie pagrindinės VM ir atliekami veiksmai su failų sistemos objektais. ŽPS sistema buvo leidžiama iš pagrindinės kompiuterio OS.

I LENTELĖ. TYRIME NAUDOTAS KOMPIUTERIS

Komponentas	Techniniai duomenys
Procesorius	AMD *Turion 64 Mobile Technology* ML-30 (1.6-GHz, 1-MB L2 *cache*)
Pagrindinė plokštė	ATI RADEON XPRESS 200M Chipset
RAM	2048-MB 333-MHz DDR SDRAM
HDD	50-GB 5400 rpm
Vaizdo korta	*Integrated* ATI MOBILITY RADEON X300 (128-MB *allocated system memory*)

Siekiant imituoti vartotojų veiksmus realiu laiku, „Python" programavimo kalba buvo sukurtas scenarijus (angl. *script*):

1) Laukia atsitiktinai sugeneruotą sekundžių skaičių (nustatoma maksimali riba).

2) Prisijungia prie nurodytos virtualios mašinos.

3) Atsitiktinai pasirenka vietą egzistuojančiame failų sistemos medyje.

4) Atsitiktinai pasirenka atlikti vieną iš šių veiksmų:

a) Sukurti naują katalogą.

b) Ištrinti esamą katalogą.

c) Sukurti naują failą.

d) Modifikuoti esamą failą.

e) Ištrinti esamą failą.

5) Laukia (vis dar prisijungęs prie VM) atsitiktinai sugeneruotą sekundžių skaičių (nustatoma maksimali riba).

6) Atsijungia nuo VM.

7) Kartoja viską iš pradžių.

Šis scenarijus paleidžiamas tuo pačiu metu iš 5-ių virtualių mašinų su skirtingais IP adresais ir jungiasi prie tos pačios pagrindinės VM (jos parametrai: 5 GB HDD, 256 MB RAM), kuri bus tiriama. Prisijungimas vyksta pagal SSH protokolą, veiksmai atliekami naudojant standartines „Linux" komandinės eilutės priemones: „mkdir", „touch", „echo" ir kt. Kad būtų galima palyginti, scenarijus visus faktiškai atliktus veiksmus ir įvykusių prisijungimų laikus išsaugo audito žurnaluose.

Virtualių mašinų kopijos nustatytu periodu buvo kuriamos rankiniu būdu, pasinaudojant virtualizacijos platformos „VirtualBox" tam skirtu funkcionalumu ir dedamos į pasirinktą katalogą tyrime naudojamame kompiuteryje, iš kurio buvo automatiškai apdorojamos iš anksto paleistos vykdyti žurnalizavimo paramos sistemos.

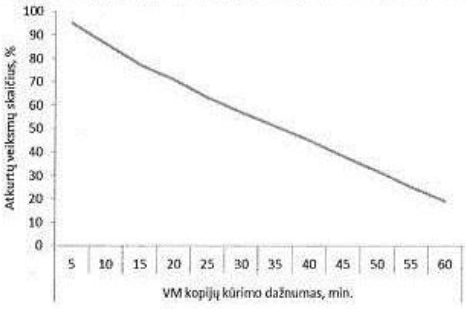
Tiriant atkurtos informacijos kiekio priklausomybę nuo VM kopijų kūrimo dažnumo, kopijos buvo daromos kas 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55 ir 60 min. Maksimalus vartotojų veiksmus imituojančių scenarijų laukimo laikas buvo nustatytas vienodas – po 15 min. Kiekvienas kopijos kūrimo laikas buvo tiriamas kartojant bandymą po 10 kartų. Rezultatai pateikiami 12 pav.

Siekiant nustatyti, kaip atkurtos informacijos kiekis priklauso nuo vartotojų veiksmų kiekio, buvo nustatytas 5 min. VM kopijų kūrimo dažnumas, o vartotojų veiksmus imituojantys scenarijai buvo koreguojami po kiekvienos kopijos sukūrimo iteracijos taip, kad atliekamų veiksmų skaičius didėtų. Tai buvo įgyvendinta keičiant maksimalų laukimo laiką į atitinkamai 15, 13, 10, 9, 8, 7, 6, 5, 3 ir 1 min. Atkurtų veiksmų skaičius buvo nustatomas po kiekvieno atlikto bandymo, rezultatai pateikiami 13 pav.

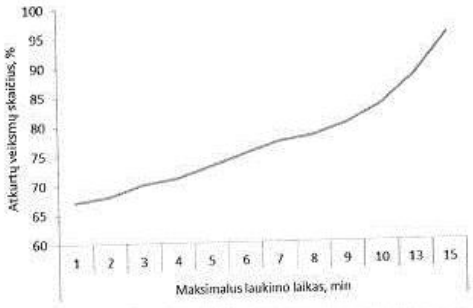
Tyrimo rezultatuose atsispindi tiesinė virtualių mašinų kopijų kūrimo dažnumo ir atkurtos informacijos kiekio priklausomybė. Kopijas kuriant kas 5 min., naudojant žurnalizavimo paramos sistemą pavyksta atkurti net 95 % vartotojų veiksmų. O priklausomybė tarp vartotojų veiksmų ir atkurtos informacijos kiekio yra eksponentinė – net ir labai didinant veiksmų skaičių, ŽPS sugeba atkurti gana daug informacijos (daugiau kaip 65 %; su sąlyga, kad VM kopijos yra kuriamos kas 5 min.).

V. IŠVADOS

Šiame straipsnyje pristatėme naują metodą-įrankį – žurnalizavimo paramos sistemą, padedančią užfiksuoti ir atkurti vartotojų veiksmus kaip įkalčius „skaičiavimų debesies"



12 pav. Atkurtos informacijos kiekio priklausomybė nuo VM kopijų kūrimo dažnumo



13 pav. Atkurtos informacijos kiekio priklausomybė nuo maksimalaus vartotojų laukimo laiko (sąlygojančio vartotojų veiksmų skaičių)

saugyklose. „Python" programavimo kalba jungianti kelias atvirojo kodo programines priemones skaitmeninei teismo ekspertizei atlikti, ŽPS įgyvendina kitų autorių pasiūlytą unifikuotą audito įrašų formatą tokio pobūdžio aplinkoms ir sukuria save aprašančių duomenų efektą, kuris, manoma, yra svarbus žingsnis sprendžiant veiksmų atkūrimo „skaičiavimų debesies" saugyklose problemą. Pateiktas metodo efektyvumo įvertinimas atskleidžia atkuriamų veiksmų skaičiaus priklausomybę nuo virtualių mašinų kopijų kūrimo dažnumo ir vartotojų atliekamų veiksmų skaičiaus: analizuojant virtualių mašinų kopijas kas 5 min., žurnalizavimo paramos sistema leidžia atkurti net 65–95 % veiksmų, priklausomai nuo juos sukūrusių vartotojų aktyvumo.

Ateityje planuojama išmatuoti ŽPS veikimo trukmės priklausomybę nuo virtualaus HDD ir RAM dydžio.

LITERATŪROS SĄRAŠAS

[1] D. Reilly, C. Wren, T. Berry, „Cloud Computing: Pros and Cons for Computer Forensic Investigations", International Journal Multimedia and Image Processing (IJMIP), Volume 1, Issue 1, March 2011.

[2] P. Mell, T. Grance, „The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standards and Technology", Special Publication 800-145, September 2011. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

[3] Access US. http://www.accessus.net/business-services/cloud-computing.

[4] Cisco Global Cloud Index: Forecast and Methodology, 2010–2016, White Paper. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns1175/Cloud_Index_White_Paper.html.

[5] R. Chow, et al., „Controlling Data in the Cloud: Outsourcing Computation without Outsourcing Control", CCSW'09, November 13, 2009, Chicago, Illinois, USA, Copyright 2009 ACM.

[6] K. Kent, S. Chevalier, T. Grance, H. Dang, „Guide to Integrating Forensic Techniques into Incident Response: Recommendations of the National Institute of Standards and Technology", Special Publication 800-86, August 2006. http://csrc.nist.gov/publications/nistpubs/800-86/SP800-86.pdf.

[7] M. Taylor, J. Haggerty, D. Gresty, R. Hegarty, „Digital Evidence in Cloud Computing Systems", Computer Law & Security Review 26, 2010, pp. 304-308.

[8] J. C. F. Cruz, T. Atkinson, „Digital Forensics on a Virtual Machine", 49th ACM Southeast Conference, March 24-26, 2011, Kennesaw, GA, USA, 2011 ACM.

[9] J. C. F. Cruz, T. Atkinson, „Evolution of Traditional Digital Forensics in Virtualization", ACM Southeast Conference, March 29-31, 2012, Tuscaloosa, AL, USA, 2012 ACM.

[10] D. Birk, „Technical Challenges of Forensic Investigations in Cloud Computing Environments", Systematic Approaches to Digital Forensic Engineering (SADFE), 2011, IEEE Sixth International Workshop.

[11] X. Fu, Zh. Ling, W. Yu, J. Luo, „Cyber Crime Scene Investigations (C2SI) through Cloud Computing", Proceedings of the 2010 IEEE 30th International Conference on Distributed Computing Systems Workshops.

[12] S. L. Garfinkel, „Digital Forensics Research: The Next 10 Years", Elsevier Ltd. Digital Investigation Magazine. No. 7, 2010.

[13] S. D. Wolthusen, „Overcast: Forensic Discovery in Cloud Environments", 2009 Fifth International Conference on IT Security Incident Management and IT Forensics, 2009 IEEE.

[14] R. K. L. Ko, et al., „Trustcloud: Framework for Accountability and Trust in Cloud Computing", 2011 IEEE.

[15] W. Zhou, et al., „Towards Data-Centric View of Cloud Security", CloudDB 2010, October 30, 2010, Toronto, Ontario, Canada, 2010 ACM.

[16] S. Krishnan, K. Snow, F. Monrose, „Trail of Bytes: Efficient Support for Forensic Analysis", CCS'10, October 4–8, 2010, Chicago, Illinois, USA, Copyright 2010 ACM.

[17] R. Marty, „Cloud Application Logging for Forensics", SAC'11 March 21-25, 2011, TaiChung, Taiwan, Copyright 2011 ACM.

[18] Y. Zhu, H. Wang, Z. Hu, G. Ahn, H. Hu, S. S. Yau, „Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds", SAC'11 March 21-25, 2011, TaiChung, Taiwan, 2011 ACM.

[19] C. Wang, Q. Wang, K. Ren, W. Lou, „Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing", IEEE INFOCOM 2010.

[20] A. A. Yavuz, P. Ning, „BAF: An Efficient Publicly Verifiable Secure Audit Logging Scheme for Distributed Systems", 2009 IEEE.

[21] R. Lu, X. Lin, X. Liang and Sh. Shen, „Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing", ASIACCS'10 April 13-16, 2010, Beijing, China, 2010 ACM.

[22] The Sleuth Kit. http://www.sleuthkit.org.

[23] The Volatility Framework. https://code.google.com/p/volatility.

[24] V. Dagienė, G. Grigas, T. Jevsikova, „Enciklopedinis kompiuterijos žodynas", 2-as papildytas leidimas, Vilnius: TEV, 2008.