KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

KAROLIS BARTKEVIČIUS

PILNŲ PAKETŲ ANALIZĖS METODO IR POŽYMIŲ DALIJIMOSI PLATFORMOS INTEGRACIJA TINKLO INCIDENTŲ APTIKIMO IR TYRIMO AUTOMATIZAVIMUI

Baigiamasis magistro darbas

Vadovas
prof. A. Venčkauskas

Konsultantas
M. Urkis

KAUNAS, 2018

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

PILNŲ PAKETŲ ANALIZĖS METODO IR POŽYMIŲ DALIJIMOSI PLATFORMOS INTEGRACIJA TINKLO INCIDENTŲ APTIKIMO IR TYRIMO AUTOMATIZAVIMUI

Baigiamasis magistro darbas
Informacijos ir informacinių technologijų sauga (kodas 621E10003)

Vadovas
(parašas) prof. A. Venčkauskas
(data)

Recenzentas
(parašas) dr. D. Rimkus
(data)

Projektą atliko
(parašas) Karolis Bartkevičius
(data) 2018 05 20

KAUNAS, 2018

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Karolis Bartkevičius

(Studento vardas, pavardė)

Informacijos ir informacinių technologijų sauga, M4096O21

(Studijų programos pavadinimas, kodas)

„Baigiamojo projekto pavadinimas“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 17 m. Sausio mėn. 02 d.

Kaunas

Patvirtinu, kad mano **Karolio Bartkevičiaus** baigiamasis projektas tema „Pilnų paketų analizės metodo ir požymių dalijimosi platformos integracija tinklo incidentų aptikimo ir tyrimo automatizavimui“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

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

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

Bartkevičius, K. „Pilnų paketų analizės metodo ir požymių dalijimosi platformos integracija tinklo incidentų aptikimo ir tyrimo automatizavimui“. Magistro baigiamasis projektas / vadovas prof. Algimantas Venčkauskas ; Kauno technologijos universitetas, Informatikos fakultetas, kompiuterių katedra.

Kaunas, 2018. 67 p.

SANTRAUKA

Šio darbo tikslas yra sukurti automatizuotą incidentų aptikimo ir tyrimo priemonę, integravus pilnųjų paketų gaudyklę bei srauto analizės įrankį „Moloch“ su incidentų tyrimo rezultatų ir požymių dalijimosi platforma – MISP. Tokios priemonės poreikis pagrindžiamas apžvelgus incidentų valdymo metodiką, šiuolaikinės tinklo saugos priemones ir nustačius, kad dauguma priemonių yra skirtos prevencijos, aptikimo ir grėsmių neutralizavimo etapams, tačiau incidentų tyrimo sritis vis dar paremta daugiausia rankiniu darbu ir primityviomis priemonėmis. Sėkmingai atlikus integraciją, sukurta priemonė leidžia žymiai efektyviau aptikti ir tirti incidentus, kadangi incidentų požymiai pažymimi automatiškai, atkrinta žmogiškosios klaidos faktorius. Darbo pabaigoje pateikiami 3 pagrindiniai integruotos priemonės pritaikymo scenarijai: naudojimas kartu su incidentų aptikimo ir prevencijos sistemomis didelės svarbos tinkluose, naudojimas kaip pagrindinė incidentų aptikimo priemonė mažos svarbos tinkluose ir tik incidentų aptikimo bei analizės dalies panaudojimas incidentų tyrimui post factum.

Bartkevicius, Karolis. *DEEP PACKET INSPECTION METHOD AND MALWARE INFORMATION SHARING PLATFORM INTEGRATION FOR NETWORK SECURITY INCIDENTS DETECTION AND ANALYSIS AUTOMATION*: *Master‘s* thesis in INFORMATION AND INFORMATION TECHNOLOGY SECURITY / supervisor assoc. prof. Algimantas Venčkauskas. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: IT Security, Network Security

Key words: Moloch, MISP, Deep Packet Inspection, Malware Detection, Network Security

Kaunas, 2018. 67 p.

SUMMARY

The goal of this paper is to create an incident detection and examination tool by integrating a full packet capturing, indexing, and database system Moloch with malware sharing platform MISP. The need for such tool is based on the conclusion of contemporary network security appliances analysis that most of them are aimed for incident prevention and protection, but the analysis and forensics parts are neglected. The integrated tool successfully detects incidents automatically and thus greatly improves investigation, because investigator no longer needs to enter queries and filters by hand, what also eliminates human error factor. In the final chapter, 3 possible use case scenarios are provided: using newly created tool along with IDPS, using it alone as low-cost but also less-effective IDS and using Moloch and MISP integration as an analysis tool for imported *pcap* files only.

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

TERMINŲ IR SANTRUMPŲ ŽODYNAS

IAPS – kaip incidentų aptikimo ir prevencijos sistema.
IAS – incidentų aptikimo sistema.
Informacijos saugos įvykis (angl. *event*) - bet koks veiksmas kompiuterių sistemoje: vartotojo prisijungimas prie kompiuterio, sesijos užmezgimas/nutraukimas, duomenų siuntimas ir t.t. arba identifikuotas reiškinys sistemoje, tarnyboje (angl. *service*) ar tinklo būsenoje, pranešantis apie galimą informacijos politikos ar kontrolės pažeidimą arba iš anksto nenumatytą situaciją, kuri gali būti susiejusi su sauga.
Informacijos saugos incidentas (angl. *incident*) dažniausiai apibrėžiamas kaip saugos politikos pažeidimas ar reali pažeidimo grėsmė. Arba „vienetinis nepageidaujamas ar netikėtas informacijos saugos įvykis arba jų serija, kuri indikuoja didelę tikimybę, kad gali būti pažeistos komercinės veiklos operacijos ir informacijos sauga“
Incidentų tyrimas (angl. *forensic, didigtal forensic, incident forensic*) yra svarbi IT saugos dalis, skirta teisingam žalos nustatymui (pavyzdžiui, kurios sistemos dalys buvo pažeistos, kurie duomenys nutekinti), kaltininkų identifikavimui, sistemos silpnų vietų radimui
Incidentų valdymas – procesų rinkinys, apimantis informacijos saugos incidentų aptikimą, pranešimą (raportavimą), tyrimą, vertinimą, jų sukeltų grėsmių neutralizavimą ir išvadų darymą.
IPA – išsamus paketų analizavimas.
IPS – incidentų prevencijos sistema.
Kontrolė – „Priemonė, mažinanti riziką. Tai gali būti bet koks procesas, politika, prietaisas ar veiksmas, kuris sumažina riziką. Tačiau gali būti, kad kontrolė ne visada sukels norimą efektą“
Paketas – tam tikro formato duomenų rinkinys, perduodamas komutuojamų paketų tinklais.
Srautas – duomenų, perduodamų tinklo jungtimi visuma.
SIAS – serverio incidentų aptikimo sistema.
TA – tėkmių analizė.
Tinklo sauga – informacijos ir informacinių technologijų saugos dalis, tirianti metodus, procedūras ir priemones kompiuterinių tinklų infrastruktūrai bei juose esančiai informacijai apsaugoti.
Tėkmė – duomenų rinkinys, aprašantis dalyvius, tarp kurių vyko komunikacija.
TIAS – tinklo incidentų aptikimo sistema.
TMIAS – tinklo mazgo incidentų aptikimo sistema.

ĮVADAS

„Pilnų paketų analizės metodo ir požymių dalijimosi platformos integracija tinklo incidentų aptikimo ir tyrimo automatizavimui“ yra „Informacijos ir informacinių technologijų saugos“ programos magistrinis baigiamasis darbas. Tiriama tinklo saugos technologijų sritis, o konkrečiau, pilnų paketų analize paremto metodo tyrimas ir taikymas IS saugos incidentų valdymo kontekste. Darbas skirtas tinklo administratoriams, saugos specialistams, incidentų tyrėjams, kurie domisi pilnų paketų analizės metodo taikymo galimybėmis bei diegimu tinkle.

Tradiciškai tinklo sauga buvo suvokiama kaip statinis procesas, kai diegiant kompiuterių tinklą nustatomos pažeidžiamiausios tinklo vietos, jos apsaugomos ugniasiene, sukuriama prieigos kontrolė, į vartotojų kompiuterius įdiegiama antivirusinė programa, o toliau rūpinamasi tik programinių sistemų atnaujinamais. Tačiau tinklams tampant vis sudėtingesniems ir didesniems, augant Interneto vartotojų ir į tinklą sujungtų įrenginių skaičiui, incidentų skaičius taip pat sparčiai auga. Negana to, augant incidentų skaičiui, atakos tuo pačiu tampa vis sudėtingesnės, sunkiau aptinkamos, įsilaužėliai dažnai būna geri savo srities specialistai, kurie yra motyvuoti, turi aiškius tikslus, įmantrias priemones ir pakankamai techninių bei finansinių resursų. Tradicinių priemonių, tokių kaip ugniasienės ir antivirusinės programos, nepakanka, ir tinklo saugos užtikrinimas tampa aktyviu, dinaminiu procesu, kai nuolat stebima tinklo veikla ir realiu laiku reaguojama į kylančias grėsmes.

Vis labiau plintant nuomonei, kad nepažeidžiamų kompiuterinių sistemų nėra, incidentų tyrimas tampa svarbia informacinių technologijų saugos dalimi, kai po aptikto įsilaužimo siekiama nustatyti įsilaužėlių tapatybes, vietą iš kur buvo vykdomas įsilaužimas, kokios priemonės naudotos, kokia padaryta žala ar nutekinta informacija, kokių tikslų buvo siekiama. Tokia informacija yra naudinga apsisaugant nuo panašių atakų ateityje, ieškant įsilaužėlių bei reguliuojant patirtą žalą.

Informacinių technologijų, o kartu ir kompiuterių tinklų saugai sparčiai vystantis, kuriamos naujos priemonės ir metodai, skirti kovoti su kylančiomis grėsmėmis, likviduoti įvykusių incidentų pasekmes, bei kaltininkų nustatymui. Vis dažniau greta ugniasienės galima išvysti įsilaužimo aptikimo ir prevencijos sistemas, svarbiuose tinklo taškuose įdiegtas monitoringo sistemas ir jutiklius, įmantriai paslėptus jaukus (angl. *honeypots*), centralizuotas tinklo valdymo sistemas. Daugėjant prieinamų tinklo saugos priemonių, joms tampant sudėtingesnėms, reikalaujančiomis nuolatinės priežiūros, auga ir keliami reikalavimai specialistams, diegiantiems bei prižiūrintiems šias sistemas, todėl didėja ir automatizavimo poreikis.

Šiuolaikinės tinklo saugos priemonės paremtos srauto analize, tačiau net ir turint pilnai įrašytą srautą, jo analizė yra sudėtingas, daug rankinio darbo reikalaujantis procesas. Todėl šio darbo tikslas yra integruoti pilnojo srauto analizavimo sistemą ir požymių dalijimosi platformą, automatizuojant potencialių incidentų aptikimą. Automatinis incidentų požymių atnaujinimas ir aptikimas daro analizę paprastesnę, sumažėja klaidų tikimybė, leidžia apdoroti didesnius kiekius informacijos. Srauto įrašymui ir analizei pasirinkta „Moloch“ programa, o incidentų dalijimosi platforma – MSIP, kurios laikomi geriausiais savo klasės įrankiais.

Darbas susideda iš 3 dalių: analizės, lyginamojo tyrimo ir „Moloch“ integravimo su MISP aprašymo. Keliami pagrindiniai uždaviniai:

1. apžvelgti incidentų valdymo ir tyrimo procesus šiuolaikiniame IS saugos kontekste;
2. apžvelgti šiuolaikines tinklo saugos technologijas, jas išnagrinėti ir klasifikuoti;
3. atlikti pilnųjų paketų gaudyklių rinkos apžvalgą;
4. atlikti „Moloch“ lyginamąją analizę su kitomis tinklo saugos priemonėmis;
5. nustatyti „Moloch“ metodo taikymo tikslingumą ir teikiamus privalumus, remiantis analizės metu gautais rezultatais;
6. integruoti „Moloch“ ir MISP, ištirti gautos sistemos teikiamus privalumus;
7. aptarti naujos, integruotos sistemos pritaikymo galimybes ir scenarijus.

1. INCIDENTŲ VALDYMO IR ŠIUOLAIKINIŲ TINKLO SAUGOS PRIEMONIŲ ANALIZĖ

1.1 Analizės tikslas

Analizės tikslas yra išsiaiškinti IS saugos incidento sąvoką, valdymo ir tyrimo procesus bei metodus, aptarti pagrindinius juos aprašančius standartus, taip pat apžvelgti pagrindines šiuolaikines tinklo saugos priemones, jų klasifikaciją, naudojamus tinklo stebėjimo metodus. Suskirsčius šiuolaikines tinklo saugos priemones į kategorijas, jos lyginamos tarpusavyje IS saugos incidentų valdymo kontekste pagal kontroles, siekiant nustatyti, kurie etapai padengiami pilnai, o kuriuose jaučiamas tinkamų priemonių trūkumas. Atliekama pilnųjų paketų gaudyklių rinkos apžvalga (daugiausiai dėmesio skiriant atviro kodo, nemokamiems sprendimams), produktai lyginami su tiriamuoju „Moloch“ metodu, pagrindžiant jo pasirinkimą šiam darbui.

1.2 Informacijos saugos įvykiai ir incidentai

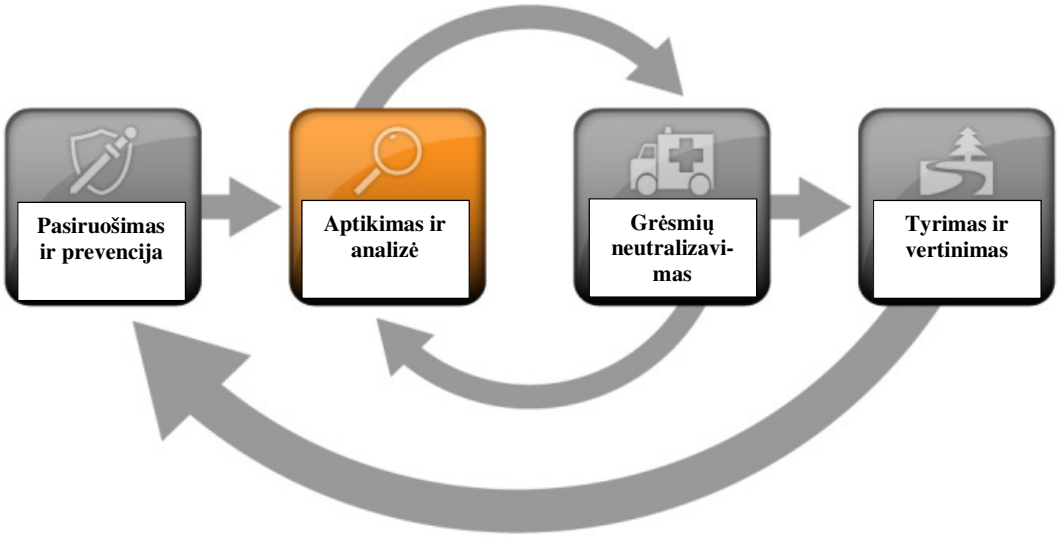
Kad ir kaip būtų apsaugota informacinė sistema, tačiau tobulų sistemų nėra ir informacijos saugos incidentai yra neišvengiami. Informacijos saugos įvykis (angl. *event*) - bet koks veiksmas kompiuterių sistemoje: vartotojo prisijungimas prie kompiuterio, sesijos užmezgimas/nutraukimas, duomenų siuntimas ir t.t. Informacijos saugos incidentas (angl. *incident*) dažniausiai apibrėžiamas kaip saugos politikos pažeidimas ar reali pažeidimo grėsmė. [1, p. 60]

„ISO 27000“ standartų serijoje informacijos saugos įvykis apibrėžiamas tiksliau: „identifikuotas reiškinys sistemoje, tarnyboje (angl. *service*) ar tinklo būsenoje, pranešantis apie galimą informacijos politikos ar kontrolės pažeidimą arba iš anksto nenumatytą situaciją, kuri gali būti susijusi su sauga“, ten pat informacijos saugos incidentas apibrėžiamas kaip „vienetinis nepageidaujamas ar netikėtas informacijos saugos įvykis arba jų serija, kuri indikuoja didelę tikimybę, kad gali būti pažeistos komercinės veiklos operacijos ir informacijos sauga“. [2, p. 4]

Įvykiai, kurie gali turėti įtakos sistemos būsenai, turėtų būti registruojami (angl. *logging*). Reikia pastebėti, kad incidento apibrėžimas priklauso nuo informacijos saugos politikos, todėl kas vienoje situacijoje yra normalus įvykis, kitoje gali būti reglamentuojamas kaip šiurkštus saugos politikos pažeidimas - incidentas. Pavyzdžiui, disko bendrinimas (angl. *sharing*) namų tinkle yra tikėtinas ir normalus įvykis, tuo tarpu organizacijos tinkle, kuriame saugomi slapti dokumentai, tai gali būti laikoma rimtu incidentu. Dažnai pasitaikantys incidentai: neautorizuoto prisijungimo prie sistemos bandymai, atsisakymo aptarnauti ataka (angl. *Denial of Service*), duomenų vagystės, neautorizuoti programinės įrangos pakeitimai ar naujos įrašymas. [1, p. 6]

Kompiuterių tinklų saugos kontekste įvykis yra bet koks veiksmas tinkle. Visos šiuolaikinės saugos priemonės analizuoja tinklo įvykius, vienaip ar kitaip lygina juos su žinomais incidentų požymiais ir, radus atitikimą, imasi prevencinių veiksmų (pavyzdžiui, ugniasienė draudžia prisijungimą prie tam tikro prievado) ar generuoja pranešimą apie galimą incidentą. Šiuolaikinės IT saugos priemonės yra nepajėgios pakankamai tiksliai nuspręsti, ar incidentas yra tikras ar ne, todėl dažnai galutinis sprendimas paliekamas kvalifikuotiems, atsakingiems asmenims. Tik atitinkamas žinias turintys žmonės pagal incidento požymius ir papildomą informaciją gali užtikrintai nuspręsti, ar grėsmė yra reali, ar tai organizuota ataka, o gal tiesiog nereikšmingas sistemos pranešimas, kurį galima ignoruoti.

1.2.1 Informacijos saugos incidentų valdymas



1.1 pav. Pagrindiniai incidentų valdymo etapai

Procesų rinkinys, apimantis informacijos saugos incidentų (toliau – incidentų) aptikimą, pranešimą (raportavimą), tyrimą, vertinimą, jų sukeltų grėsmių neutralizavimą ir išvadų darymą, vadinamas incidentų valdymu (angl. *management*), pagal „ISO/IEC“ standartų organizacijos naudojamą terminologiją, arba tvarkymu (angl. *handling*), pagal „NIST“ standartų organizacijos terminologiją [2, p. 5] [1, p. 1]. Toliau šiame darbe naudojamas bendras „incidentų valdymo“ terminas, neskiriant, kuria standartų organizacijos terminologija remiamasi. Incidentų valdymas gali būti kaip atskiras, savarankiškas standartas arba standartų sistemos (angl. *framework*) dalis. „NIST“ išleistoje specialioje publikacijoje „800-61: Kompiuterių sauga: incidentų tvarkymo gide“ pagrindiniai incidentų valdymo etapai (žr. 1.1 pav.) aprašomi detaliau:

1) **Pasiruošimas ir prevencija.** Pasiruošimo etape aprašoma incidentų valdymo politika, surenkama incidentų valdymo komanda [1, p. 22]. Per didelis įvykių ir incidentų kiekis gali apkrauti incidentų valdymo komandą, sumažinti jos reakcijos laiką ir efektyvumą, todėl svarbu pasirūpinti incidentų prevencija – IS ir tinklo sauga. Į prevencijos sąvoką taip pat įeina rizikų valdymas, galinių įrenginių apsauga, apsauga nuo kenkėjiškų programų, vartotojų mokymas ir švietimas. [1, p. 24]
2) **Aptikimas ir analizė**. Techniniai informacijos saugos įvykiai dažniausiai aptinkami saugos sistemų automatiškai. Tinklo saugos kontekste, tokių sistemų pavyzdžiai yra saugos priemonės – ugniasienės, IAS ir jaukai [1, pp. 26-27]. Ne visi įvykiai, apie kuriuos pranešta, yra teisingi. Pavyzdžiui, IAS gali klaidingai identifikuoti incidentą. Net jei pranešimas apie įvykį yra teisingas, tai nebūtinai reiškia, kad aptiktas incidentas, tai gali būti sistemos sutrikimas, žmogiškoji klaida ar kiti, su IS sauga nesusiję veiksniai. Nustatyti, ar įvykis yra incidentas, gali tik kvalifikuotas specialistas, turintis pakankamai kontekstinės informacijos sprendimui priimti. Analizės etapo tikslas yra nustatyti, įvykis yra incidentas ar ne. Nustačius, kad aptiktas incidentas, jis pirmiausia aprašomas, dokumentuojamas, prioretizuojamos ir eskaluojamas. [1, pp. 28-29]
3) **Grėsmių neutralizavimas**. Nustačius incidentą, jei jis dar vyksta, siekiama sustabdyti jo tolimesnį plitimą ir sumažinti daromos žalos mastus. Ne mažiau svarbus yra informacijos apie incidentą rinkimas ir išsaugojimas – žurnalų įrašai, srauto įrašai, paveiktų sistemų diskų kopijos ir kita, kas vėliau gali būti naudojama incidento tyrimui. Galiausiai, neutralizavus incidentą, atstatomos sistemos ir normalus darbas. [1, pp. 35-37]

4) **Tyrimas ir vertinimas**. Sėkmingai susitvarkius su incidentu, svarbu atlikti išsamų tyrimą, siekiant nustatyti incidento priežastis, pasekmes bei kaip apsisaugoti nuo panašių incidentų ateityje. [1, pp. 38-39]

1.2.2 Informacijos saugos incidentų tyrimas

Incidentų tyrimas (angl. *forensic, didigtal forensic, incident forensic*) yra svarbi IT saugos dalis, skirta teisingam žalos nustatymui (pavyzdžiui, kurios sistemos dalys buvo pažeistos, kurie duomenys nutekinti), kaltininkų identifikavimui, sistemos silpnų vietų radimui, paprastai atliekama incidentų valdymo tyrimo ir vertinimo etape. Tinklo saugos incidentų tyrimas (toliau – incidentų tyrimas) apsiriboja duomenų tyrimu, surinktu iš tinklo įrenginių ir srauto. Incidentų tyrimas susideda iš 4 pagrindinių etapų: [3, p. 16]

- **Duomenų rinkimas**. Duomenys (pilni paketai, tėkmės, registracijos įrašai) gaunami iš ugniasienių, maršrutizatorių, paketų gaudyklių, protokolų analizatorių, IAS, VPN serveriai ir kiti įrenginiai, kurie turi srauto analizavimo ir įvykių registravimo funkcijas. [3, pp. 65-68]. Iš šių priemonių, daugiausiai duomenų suteikia paketų gaudyklės. [3, p. 74]
- **Apžiūrėjimas ir filtravimas**. Aptikus incidentą, svarbu atrinkti su juo susijusią informaciją iš visų surinktų duomenų. Tai yra sudėtingas etapas, reikalaujantis plačių tinklo, protokolų ir IT žinių. Filtravimas atliekamas pasitelkiant specialią programinę įrangą rankiniu ar automatizuotu būdu. [3, pp. 71-72]
- **Analizė**. Šiuo etapu siekiama surasti incidento sukelėją ar šaltinį, naudojantis atrinktais duomenimis. [3, p. 30]
- **Raportavimas**. Surinktos informacijos apibendrinimas ir išvadų priėmimas. Šiame etape svarbu pabrėžti, kokių priemonių turi būti imtasi, kad išvengti panašių incidentų ateityje. [3, p. 31]

Šiuolaikiniame pasaulyje incidentai dažniausiai nebūna izoliuoti, vietiniai įvykiai, o apimantys daug tinklų, įmonių ir šalių. Efektyvesniam incidentų tyrimui ir raportavimui, yra sukurta specialių organizacijų, kurių kelios pagrindinės yra CERT[1] ir ENISA[2].

1.2.3 Kontrolės

Vienas svarbiausių sėkmingos incidentų prevencijos aspektų yra tinkamai parinktos kontrolės. „IEC/ISO 27000“ standarte kontrolė apibrėžiama kaip: „Priemonė, mažinanti riziką. Tai gali būti bet koks procesas, politika, prietaisas ar veiksmas, kuris sumažina riziką. Tačiau gali būti, kad kontrolė ne visada sukels norimą efektą“ [2, p. 3]. Jau vien iš apibrėžimo matosi, kad kontrolės negarantuoja apsaugos nuo incidentų, o tik sumažina jų riziką, todėl incidentų valdyme prevencija neapsiribojama, ir ji yra tik vienas iš kelių viso proceso etapų.

Galima rasti nemažai standartų, struktūrizuotų dokumentų (angl. *framework*) ir komercinių produktų aprašančių įvairias IS kontroles ir jų rinkinius: „ISO/IEC 27001“ standarte [4] aprašomos 114 kontrolės, suskirstytos į 14 funkcinių grupių, „NIST“ specialioje publikacijoje 800-53 „Saugos ir privatumo kontrolės federacinėms institucijoms ir organizacijoms“ [5] pateikiama 18 kontrolių grupių, CIS (angl. *„Center for Internet Security“*) organizacijos išleistame dokumente „CIS kritinės saugos kontrolės efektyviai kyber-gynybai“ [6] aprašoma 20 kontrolių grupių.

Šiame darbe daugiausia remiamasi CIS kontrolėmis, kadangi jos yra sukurtos remiantis jau esamomis kontrolėmis, stengiantis perimti geriausias savybes ir paaiškinti ar pataisyti mažiau aiškias ir konkrečias iš jų. CIS siūlo 20 kontrolių grupių, kurios suskirstytos į 3 pogrupius – bazines, pamatines ir organizacines.

[1] http://www.cert.org/
[2] https://www.enisa.europa.eu/

Bazinės grupės:

1. Turimos aparatinės įrangos inventorizacija ir kontrolė;
2. Turimos programinės įrangos inventorizacija ir kontrolė;
3. Nenutrūkstamas pažeidžiamumų valdymas (angl. *Vulnerability management*);
4. Kontroliuojamas administratoriaus teisių naudojimas;
5. Saugūs aparatinės ir programinės įrangos, mobiliųjų įrenginių, nešiojamų kompiuterių, asmeninių kompiuterių ir serverių nustatymai (angl. *Configuration*);
6. Žurnalinių įrašų stebėjimas, analizė ir auditavimas.

Pamatinės grupės:

7. Elektroninio pašto programų ir naršyklių apsauga;
8. Apsauga nuo kenksmingų programų;
9. Tinklo prievadų, protokolų, paslaugų ribojimas ir kontroliavimas;
10. Duomenų atstatymo galimybės;
11. Saugūs tinklo įrenginių (ugniasienių, maršrutizatorių, komutatorių ir t.t.) nustatymai;
12. Perimetro gynyba;
13. Duomenų apsauga;
14. Prieigos kontroliavimas pagal „poreikį žinoti“ (angl. *„Need to Know“*) principą;
15. Bevielio tinklo apsauga;
16. Paskyrų (angl. *Account*) kontroliavimas ir stebėjimas.

Organizacinės grupės:

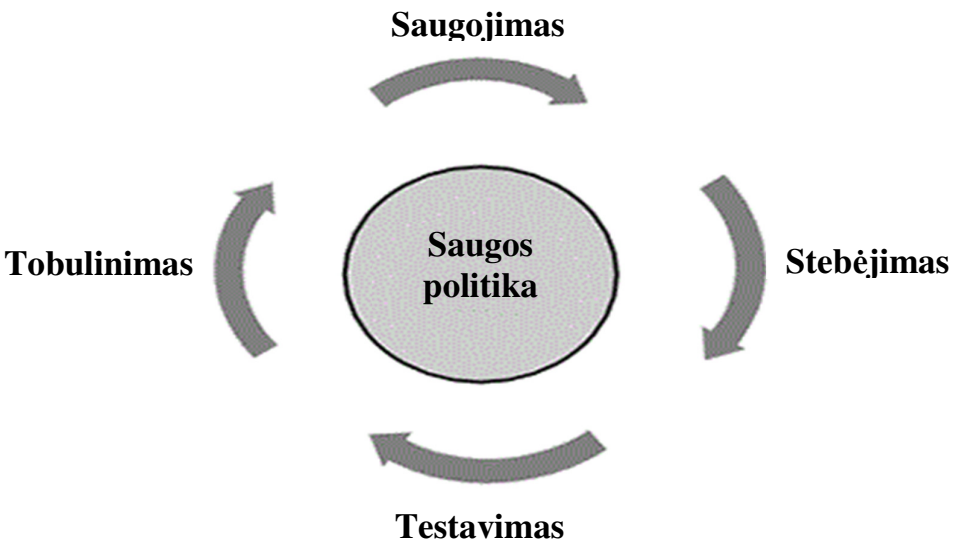
17. IS saugos mokymų programa;
18. Programinės įrangos sauga;
19. Incidentų valdymas;
20. Įsiskverbimo testai ir „raudonosios komandos“ (angl. *Red Team*) pratybos.

1.3 Šiuolaikinė tinklo sauga

Tinklo sauga yra informacijos ir informacinių technologijų saugos dalis, tirianti metodus, procedūras ir priemones kompiuterinių tinklų infrastruktūrai bei juose esančiai informacijai apsaugoti. Taikomuoju požiūriu, tinklų sauga apibrėžiama kaip organizacijos strategija, procedūrų, techninių ir programinių priemonių visuma [7] skirta kompanijos tinklui apsaugoti.

Tradiciškai tinklo sauga buvo daugiausiai orientuota į incidentų prevenciją. Tačiau nusikaltimų ir įsilaužimų elektroninėje erdvėje skaičiui nuolat ir sparčiai augant, atsirandant naujų grėsmių ir atakos vektorių [8], tinklo sauga taip pat tampa nuolatiniu, cikliniu procesu, nes siekiant efektyviai apsaugoti tinklus reikia nuolat tobulinti procedūras ir priemones. Tinklo įrangos gamybos kompanija „Cisco“ tinklo saugą apibrėžia kaip saugos politika paremtą nuolatinį procesą (žr. 1.2 pav.), kuris apima 4 etapus [9]:

1. Saugos priemonių diegimas;
2. Tinklo stebėjimas;
3. Tinklo testavimas;
4. Saugos politikos ir priemonių tobulinimas.



1.2 pav. „Cisco“ tinklo saugos procesas

Sugretinus „Cisco“ tinklo saugos ir „NIST“ incidentų valdymo procesus, matoma, kad tinklo sauga įsilieja į incidentų valdymo pirmą (pasiruošimo ir prevencijos), antrą (aptikimo ir analizės) etapus. Tačiau taip pat glaudžiai siejasi su trečiu (grėsmių neutralizavimo) bei ketvirtu (tyrimo ir vertinimo) etapais, kadangi tinklo saugos priemonių surinkta informacija – žurnaliniai įrašai, srautų informacija, pilnų paketų rinkiniai ir kiti duomenys, padeda nustatyti žalos mąstą ir atlikti išsamų incidentų tyrimą.

Toliau apžvelgiamos pagrindinės šiuolaikinės tinklo saugos priemonės, jų pritaikymas ir konkrečių kontrolių atitikimas joms. Remiamasi anksčiau išvardintomis 20 CIS kontrolių.

1.3.1 Tinklo saugos priemonių klasifikacija

Griežtos, visuotinai priimtos tinklo saugos priemonių klasifikacijos nėra, įvairūs šaltiniai pateikia įvairias klasifikacijas, priklausomai nuo to, kokiu aspektu ir kokiame kontekste sistemos yra nagrinėjamos. Šiame darbe nesiekiama sukurti išsamios tinklo saugos priemonių klasifikacijos, o tik išskiriami bruožai, reikalingi tolimesniam sprendimų palyginimui, todėl apibrėžiama klasifikacija pagal paskirtį, naudojamą metodą ir konfigūraciją.

Pagal paskirtį šiuolaikinės tinklo saugos priemonės dažniausiai skirstomos į ugniasienes, incidentų aptikimo (angl. *Intrusion Detection Systems*) ir prevencijos (angl. *Intrusion Prevention Systems*) sistemas, jaukus (angl. *Honeypots*) [10, p. 200]. Kai kuriuose šaltiniuose [11] [12] kaip atskira kategorija išskiriama tinklo stebėjimo sistemos (angl. *Network Monitoring System*), tačiau tai nėra visiškai atskira kategorija, o labiau kelių, anksčiau minėtų, priemonių rinkinys su papildomais įrankiais. Šios rolės nėra griežtai atskirtos, ir vienas įrenginys gali atlikti kelias paskirtis, pavyzdžiui, ugniasienė, kuri kartu yra ir įsilaužimo aptikimo sistema.

Visos išvardintos priemonės savo darbą atlieka stebėdamos tinklo srautą. Stebėjimui dažniausiai naudojami 2 metodai: tėkmių (angl. *flows*) analizavimas ir išsamus paketų analizavimas (angl. *Deep Packet Inspection*). Šie metodai plačiau aptariami 1.4 skyriuje.

Pagal konfigūraciją, tinklo saugos sistemos skirstomos į taškines (angl. *host-based*), paskirstytas (angl. *distributed*) ir centralizuotas (angl. *centralised*). Taškinės sistemos yra tokios, kurios susideda iš vieno įrenginio ir yra įdiegtos viename tinklo taške. Tokių sistemų pavyzdys yra ugniasienės, kurios dažniausiai statomos tinklo ar svarbių tarnybinių stočių prieigose ir veikia savarankiškai. Paskirstytos sistemos gali susidaryti iš daugelio įrenginių – sensorių, kurie gali būti geografiškai nutolusiuose tinkluose ir kuriuose įdiegiami savarankiški, analogišką paskirtį turintys

agentiniai procesai. Jei paskirstytieji agentiniai procesai veikia ne savarankiškai, o tik renka duomenis ir juos siunčia į centrinį įrenginį ar procesą apdorojimui ir sprendimo priėmimui, tokios sistemos vadinamos centralizuotomis. Centralizuotų sistemų principu paprastai veikia įsilaužimo aptikimo sistemos.

1.3.1.1 Ugniasienės

Ugniasienės yra tinklo saugos priemonių kategorija, kurios stebi ir kontroliuoja įeinantį/išeinantį tinklo srautą pagal iš anksto aprašytas taisykles [13, p. 32]. Ugniasienės paprastai statomos tinklo prieigose kaip barjeras, atskiriantis sąlyginai saugų vidinį tinklą nuo išorinio tinklo (Interneto) grėsmių [14, p. 40]. Ugniasienės yra priemonė įmonės saugos politikai (angl. *IT security policy*) įgyvendinti, nurodant IP adresų rėžius, protokolus, duomenų tipus ir kitas sąlygas, kurioms esant leidžiamas ar draudžiamas tam tikras srautas saugomame tinkle [15, p. 7].

Ugniasienių kategorija yra plati, jos gali būti tiek atskiri fiziniai įrenginiai, tiek programiniai sprendimai, tiek į maršrutizatorius integruoti procesai [11, p. 250]. Pagal konfigūraciją, ugniasienės paprastai būna taškinės. Paprasčiausia ugniasienes skirstyti pagal darbo lygmenis:

- Tinklo lygmens ugniasienės pagal OSI modelį [16] dirba tinklo (trečiame) ir transporto (ketvirtame) lygmenyse. Šios kategorijos ugniasienės naudoja dalinį paketų analizavimo metodą siekiant nustatyti, ar paketo antraštėje esanti informacija (IP adresai, TCP/UDP prievadai) atitinka kurią nors aprašytą taisyklę, ir priima atitinkamą sprendimą. Toliau šiame darbe, jei nenurodoma kitaip, yra kalbama apie tinklo lygmens ugniasienes.
- Aplikacijos lygmens ugniasienės pagal OSI modelį dirba aplikacijos (septintajame) lygmenyje. Šios kategorijos ugniasienės taip pat remiasi taisyklių lentele sprendimų priėmimui, tačiau analizuoja daugiau informacijos (pvz., HTTP užklausas, FTP komandas ir t.t.), gaunamos pilno paketų analizavimo metodu. [11, pp. 253-254]

Ugniasienės yra pagrindinė tinklo saugos priemonė, todėl jos dažniausiai minimos ir CIS kritinių saugumo valdiklių siūlomuose metoduose. Pagal CIS, ugniasienės rekomenduojamos naudoti išsamių įrašų (angl. *logs*) apie srautus rinkimui (CSC-6) [17, p. 25], neaiškių ir pavojingų Interneto puslapių bei elektroninių laiškų filtravimui (CSC-7) [17, p. 28], nebūtinų veiklai prievadų, protokolų ir servisų draudimui (CSC-9) [17, p. 34], perimetro gynybai (CSC-12) [17, p. 42], aplikacijos lygmens taikomųjų programų žinomų atakų filtravimui (CSC-18) [17, p. 63].

Pagrindiniai ugniasienių privalumai:

1. Sėkmingai filtruoja srautą;
2. Gali blokuoti protokolus ir servisus, kurie yra lengvai pažeidžiami;
3. Atskiria vidinį tinklą nuo išorinio;
4. Gali rinkti išsamią statistiką apie srautą. [10, p. 201]

Pagrindiniai ugniasienių trūkumai:

1. Nepatogus, rankinis taisyklių konfigūravimas;
2. Pasyvi saugos priemonė, todėl negali aptikti ir priešintis atakoms;
3. Neapsaugo nuo atakų iš tinklo vidaus;
4. Neapsaugo nuo nežinomų aplikacijos lygmens atakų, pavyzdžiui, virusų, kirminų, Trojos arklių ir t.t. [10, p. 201]

1.3.1.2 Incidentų aptikimo sistemos

Incidentų aptikimas yra procesas, kurio metu siekiama identifikuoti ir atskirti kenksmingas veiklas tinkle ar jame esančiuose įrenginiuose nuo įprastos veiklos [18, p. 4]. Incidentų aptikimo sistemos (toliau – IAS), kurios sugeba ne tik aptikti atakas, bet ir reaguoti į jas, vadinamos incidentų prevencijos sistemomis. Dauguma šiuolaikinių sistemų gali atlikti tiek aptikimo, tiek prevencijos

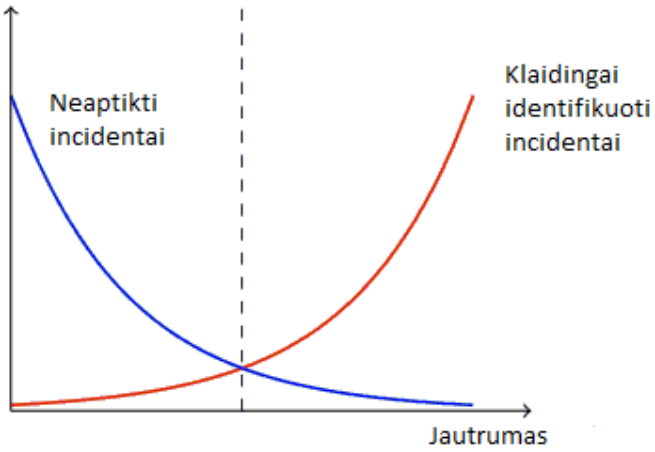
funkcijas; tokios sistemos bendrai žinomos kaip incidentų aptikimo ir prevencijos sistemos – IAPS (angl. *Intrusion Detection and Prevention Systems – IDPS*) [11, p. 294]. Šiame darbe, kaip ir daugelyje šaltinių, IAS naudojamas kaip bendras terminas, apimantis ir IAPS. Tikslesni terminai naudojami tik tada, kai reikia pabrėžti IAS/IAPS skirtumus ar kalbant apie savybes, būdingas konkrečiai grupei. IAS gali būti realizuotos tiek kaip atskiras įrenginys, tiek kaip programinės sistemos.

IAS skirstomos pagal konfigūraciją, naudojamą tinklo/įrenginio stebėjimo metodą ir tai, kaip priimami sprendimai. Pagal konfigūraciją IAS skirstomos:

- Serverio incidentų aptikimo sistemos (angl. *Host Intrusion Detection System – HIDS*). Ši IAS grupė periodiškai tikrina svarbias serveryje esančias laikmenas ir tikrina, ar jos nebuvo modifikuotos, pakeistos, ištrintos. Aptikus pažeidimą, serverio incidentų aptikimo sistema (toliau – SIAS) praneša administratoriui. Iš kitų tinklo saugos priemonių SIAS išsiskiria tuo, kad nestebi ir neanalizuoja bendro srauto, todėl tik sąlyginai priskiriama tinklo saugos priemonių grupei. SIAS iš kitų tinklo saugos priemonių išsiskiria ir savo teisiniu statusu, kadangi analizuoja ne srautą, o vidinę sistemą, kuri nėra taip griežtai reglamentuojama įstatymais.
- Tinklo mazgo incidentų aptikimo sistemos (angl. *Network Node Intrusion Detection System – NNIDS*). Tinklo mazgo incidentų aptikimo sistema (toliau – TMIAS) taip pat yra diegiama galiniame tinklo įrenginyje, tačiau nuo SIAS skiriasi tuo, kad stebi ne vidines laikmenas, o įeinantį/išeinantį paketų srautą.
- Tinklo incidentų aptikimo sistemos (angl. *Network Intrusion Detection Systems – NIDS*). Kaip ir TMIAS, tinklo incidentų aptikimo sistema (toliau – TIAS) stebi paketų srautą, tačiau ne vieno konkretaus įrenginio, o visos tinklo dalies, prie kurios yra prijungta. [19, p. 3]

Pagal sprendimo priėmimo metodiką, IAS skirstomos:

- Požymio aptikimu (angl. *Signature-based*) paremtos IAS tiria, ar naujai gauti duomenys atitinka žinomų atakų požymius. Tokios sistemos pasižymi tikslumu, mažu neteisingai identifikuotų grėsmių skaičiumi (angl. *false positive*) (žr. 1.3 pav.), tačiau jų veikimui reikia išsamios atakų požymių duomenų bazės.
- Anomalijų aptikimu (angl. *anomaly-based*) paremtos IAS lygina duomenis su normaliu veikimo modeliu (angl. *model of normality*), ieškant nukrypimų, vadinamų anomalijomis. Aptikus anomaliją, generuojamas pranešimas ir įspėjamas sistemos administratorius. Normalaus veikimo modelio kūrimui ir duomenų analizei pasitelkiami neuroniniai tinklai, statistinė analizė, Markovo modelis. Šio tipo IAS didžiausias privalumas yra tas, kad jos gali aptikti ir nežinomas atakas, kas paprastai yra neįmanoma naudojant kitus metodus. Pagrindinis anomalijų aptikimo metodo trūkumas yra tas, kad galimas neteisingas neįprastos, tačiau normalios veiklos interpretavimas kaip anomalijos, dėl to šios sistemos paprastai pasižymi didesniu neteisingai identifikuotų grėsmių skaičiumi. [18, p. 14]



1.3 pav. IAS jautrumo derinimo problemos iliustracija

Pagal tinklo stebėjimo metodą, IAS skirstomos:
- Paketų analizavimą (angl. *packet inspection*).
- Tėkmių analizavimą.

Šie metodai išsamiau apžvelgiami 1.3.2 skyriuje.

CIS incidentų aptikimo sistemas rekomenduoja naudoti ugniasienių silpnųjų vietų kompensavimui. IAS taikomas kenksmingų programų (angl. *malware*) aptikimui (CIS-8) [17, p. 31] ir tinklo gynybai, ypač nuo iš vidaus kylančių grėsmių (CIS-12) [17, p. 43].

Pagrindiniai IAS privalumai:
- Efektyviai aptinka sudėtingesnes atakas.
- Incidentų aptikimas realiu laiku.
- Leidžia centralizuotai stebėti tinklo saugumo būklę. [10, p. 202]

Pagrindiniai IAS trūkumai:
- Nėra visiškai patikimos, todėl reikia rinktis tarp tikslumo ir neužfiksuotų incidentų skaičiaus. Didinant IAS jautrumą, didėja aptinkamų atakų skaičius, tačiau neišvengiamai auga ir neteisingai identifikuotų incidentų skaičius (angl. *false positives*). Mažinant IAS jautrumą, auga neaptiktų incidentų skaičius (angl. *false negatives*).
- Nustačius incidentą, reikalingas žmogiškasis įsikišimas sprendimo priėmimui dėl tolimesnių veiksmų.
- Norint didinti tikslumą, reikia rinkti ir analizuoti daugiau informacijos, tuo pačiu didėjant ir resursų poreikiui. [10, p. 202]

1.3.1.3 Jaukai

Jaukas yra specialiai paruoštas serveris, sistema ar tinklas, siekiant privilioti įsilaužėlį ir rinkti informaciją apie jo naudojamus metodus ir jį patį [20]. Jaukai turi tik vieną vartotoją – administratorių, todėl bet koks kitas bandymas prisijungti prie sistemos traktuojamas kaip saugumo pažeidimo incidentas. Teisinis šio metodo statusas nėra apibrėžtas, todėl jaukus rekomenduojama naudoti atsakingai [21, p. 91]. Incidentų aptikimui naudojamos panašios priemonės kaip IAS [11, p. 325]. Dėl savo specifikos, sudėtingumo ir neaiškaus teisinio statuso, jaukai šiame darbe plačiau nenagrinėjami.

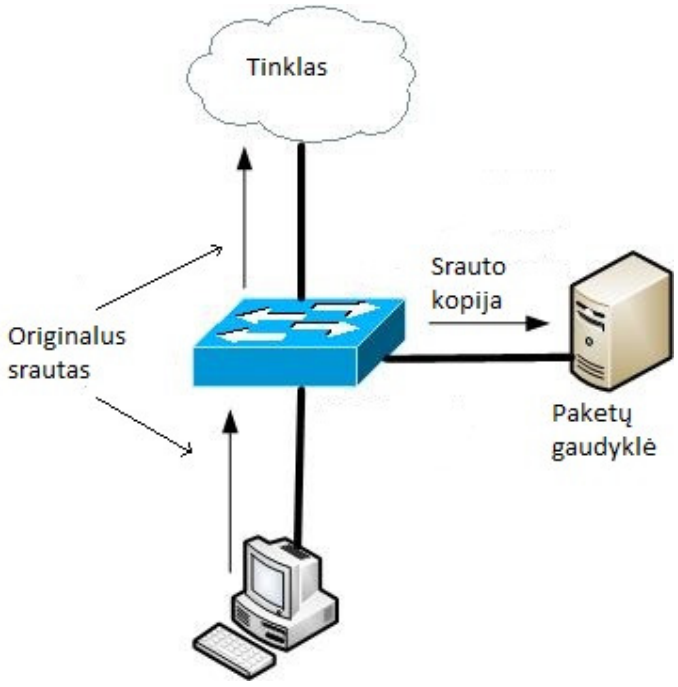
1.3.2 Srauto stebėjimo metodai

Visos tinklo saugos priemonės stebi duomenų srautus visame tinkle, jo dalyje ar galiniame įrenginyje. Pagal tai, kaip interpretuojami duomenų srautai, išskiriami du metodai: paketų analizė ir tėkmių analizė [22, p. 1].

Paketų analizės metodu tinklo srautas interpretuojamas ir analizuojamas paketais. Priklausomai nuo to, kuri paketo dalis naudojama analizei, išskiriami lygiai – dalinis (angl. *partial*) ir pilnas (angl. *full*), dar kitaip vadinamas išsamus, paketų analizavimas. Daliniam paketų analizavimui naudojamas ne pilnas paketas, o tam tikras laukų skaičius, dažniausiai adresai ir protokolų antraštės. Išsamus paketų analizavimo metodas naudoja visus paketo laukus [23, p. 14].

Paketai gaudomi dviem būdais. Pirmasis yra paketų kopijavimas maršrutizatoriuje ar komutatoriuje (žr. 1.4 pav.) panaudojant specialią funkciją – prievado dubliavimą (angl. *port spanning*). Naudojant prievado dubliavimą, paketai, einantys per vieną prievadą, yra nukopijuojami ir išvedami į atskirą prievadą, prie kurio prijungtas pilnųjų paketų gaudyklės (angl. *full packet capture*) – toliau PKG – įrenginys. Antrasis būdas yra specialaus pasiklausymo įrenginio (angl. *tap*) įterpimas į tinklo jungtį. Pasiklausymo įrenginys yra pasyvus, netrikdantis tinklo darbo, leidžiantis skaityti paketus tiesiogiai iš tinklo jungties [22, p. 3].

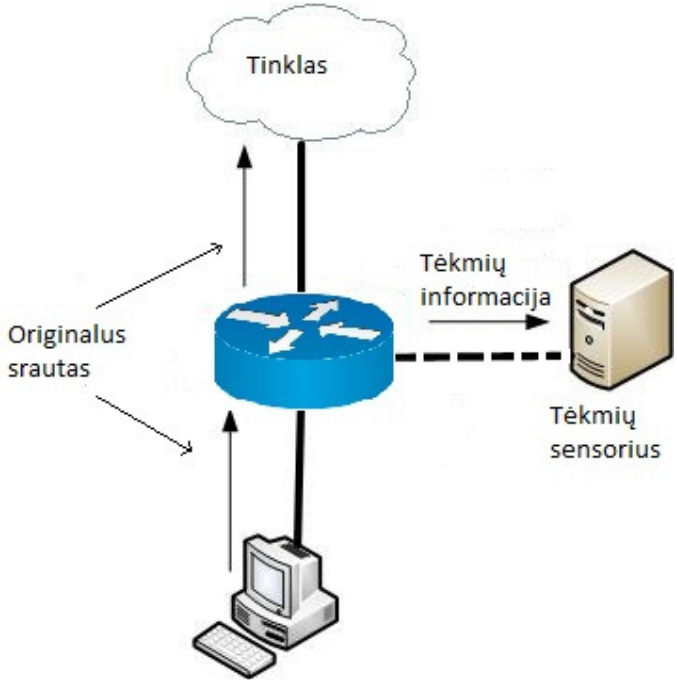
Pasiklausymo metodas yra pranašesnis už prievado dubliavimo metodą, nes maršrutizatorius (komutatorius) prioritetą teikia savo pagrindiniam darbui – paketų maršrutizavimui (komutavimui), todėl esant didelei apkrovai ir trūkstant resursų, dalis paketų gali likti nenukopijuoti. Kitas prievado dubliavimo trūkumas yra tas, kad dėl paketų buferizacijos ir iškraipymų nukopijuotas srautas nebūtinai bus identiškas esančiam tinkle [24, p. 7]. Dėl aptartų priežasčių prievado dubliavimo metodas rekomenduojamas naudoti nedideliuose, mažų srautų tinkluose. Esant sudėtingesnei tinklo topologijai, dideliems srautams paprastai diegiami pasiklausymo įrenginiai [22, p. 3].



1.4 pav. Pilnų paketų gaudymo schema

Antrasis metodas yra paremtas tėkmių analize. Tėkmė yra duomenų rinkinys, aprašantis dalyvius, tarp kurių vyko komunikacija. Į šį rinkinį įeina: abiejų įrenginių IP adresai, prievadų numeriai, naudotas protokolas, persiųstų duomenų kiekis ir kiti, papildomi duomenys. Tėkmės gali būti generuojamos maršrutizatoriaus, šakotuvo ar serverio (žr. 1.5 pav.). Jų formatas nėra standartizuotas, todėl rinkoje yra daug konkuruojančių komercinių variantų: „Cisco“ sukurtas

„NetFlow“, „Juniper Networks“ – „Jflow“, „3Com/HP“ – „NetStream“ ir kiti. Atviro kodo sprendimai dažnai naudoja „sFlow“ formatą. [22, p. 2]



1.5 pav. Tėkmių rinkimo schema, kai jas generuoja maršrutizatorius

Kiekvienas šių metodų turi savų privalumų ir trūkumų. Bendru atveju, išsamus paketų analizavimas (toliau – IPA) suteikia pilną informaciją apie srautą, tačiau paketų rinkimas, saugojimas, analizavimas reikalauja daug resursų, ir dėl to yra nepraktiškas ar neįmanomas esant dideliems srautams. [18, p. 7]

Tėkmių analizė (toliau – TA) atlieka dalinį srauto analizavimą, todėl reikalauja žymiai mažiau resursų (tėkmių generuojamas srautas yra 0.1 – 0.2 % nuo pilno srauto), tačiau nepateikia tiek informacijos kiek IPA [18, p. 6]. Imant telefoninių pokalbių analogiją, TA pateiktų informaciją apie tai, kas ir kam skambino bei kiek laiko truko pokalbis, tuo tarpu IPA papildomai išsaugotų visą pokalbio turinį. TA neskirta pakeisti IPA, o naudojama kaip papildoma priemonė išankstinio incidentų aptikimo sistemose, esant dideliems duomenų srautams. [25, p. 2]

Srauto ir tėkmių analizavimo metodai dažniausiai naudojami kaip sudėtinė didesnės sistemos dalis, pavyzdžiui, incidentų aptikimo sistemose, tačiau gali būti realizuojami ir kaip pavienės programos. Kai šie metodai realizuojami kaip pavienės programos, jos, pagal CIS, gali būti naudojamos duomenų apsaugai, aptinkant ir filtruojant tam tikrus raktažodžius, nesankcionuotai užšifruotus duomenis sraute (CSC-12) [17, pp. 47-48].

1.3.3 Srauto stebėjimą reglamentuojantys įstatymai

Tobulėjančios tinklo stebėjimo techninės priemonės, pinganti duomenų saugyklų talpa suteikia ne tik naujų galimybių, tačiau kelia pavojų ir gali būti išnaudojamos cenzūrai, disidentų persekiojimui, masiniam šnipinėjimui, asmens duomenų vagystei ir kitai veiklai, varžančiai žmogaus teises, prieštaraujančiai Interneto neutralumo principui, pažeidžiančiai asmens privatumą [26, p. 5]. Siekiant apsaugoti vartotojus, kuriami įstatymai, reglamentuojantys tinklo stebėjimą ir klientų duomenų naudojimą. Šiame darbe nagrinėjamas tik tinklo stebėjimas, siekiant užtikrinti IT saugą ir tirti saugumo pažeidimus; į cenzūros, srauto formavimo, vartotojų veiklos sekimą Internete ir kitų veiklų teisinį reglamentavimą nesigilinama.

Lietuvoje asmens duomenų tvarkymą reglamentuoja asmens duomenų teisinės apsaugos [27], elektroninių ryšių [28] ir kibernetinio saugumo įstatymai [29]. Teisinė bazė priklauso nuo objekto ir subjekto santykio, todėl išskiriami keli nagrinėjami scenarijai: tinklo stebėjimas iš viešųjų elektroninių paslaugų tiekėjo ir privačios kompanijos tinklo pozicijų. Fiziniams asmenims asmens duomenų apsaugos įstatymas netaikomas [21, 1 straipsnis, 4 dalis]: „4. Šis įstatymas netaikomas, jeigu asmens duomenis tvarko fizinis asmuo ir tik asmeniniams poreikiams, nesusijusiems su verslu ar profesija, tenkinti“, todėl asmeniniam naudojimui visos privataus tinklo srauto stebėjimo ir analizavimo priemonės yra leidžiamos, tačiau tik turint nuosavybės teisės į tinklą, kadangi priešingu atveju, tai yra traktuojama kaip prisijungimas prie elektroninių ryšių tinklų, kuris be savininko (ūkio subjekto) sutikimo yra draudžiamas[22, 42 straipsnis].

Pirmuoju atveju, viešųjų elektroninių paslaugų tiekėjui yra keliami griežtesni asmens duomenų tvarkymo reikalavimai. Elektroninių ryšių įstatymo 61-jame straipsnyje, 1-oje dalyje teigiama: „1. Draudžiama be faktinių elektroninių ryšių paslaugų naudotojų sutikimo klausytis, įrašyti, kaupti ar kitu būdu perimti pranešimų turinį ir srauto duomenis ar su jais susipažinti, išskyrus atvejus, kai tai galima teisėtai daryti pagal šio Įstatymo 66 ir 77 straipsnius. Be faktinių elektroninių ryšių paslaugų naudotojų sutikimo draudžiama atskleisti elektroninių ryšių tinklais perduodamų pranešimų turinį ir (ar) susijusius srauto duomenis arba sudaryti sąlygas sužinoti tokią informaciją ir (ar) susijusius srauto duomenis, išskyrus įstatymo nustatytus atvejus“. Čia esminės sąvokos yra pranešimas[3] ir srauto duomenys[4], pagal kurių apibrėžimą, tinklo stebėjimas tiek tėkmių, tiek paketų analizės metodu būtų draudžiamas. Tačiau sekančioje dalyje numatoma papildoma išlyga: „2. Šio straipsnio 1 dalies nuostatos nedraudžia nepažeidžiant konfidencialumo principo laikinai išsaugoti perduodamus pranešimus, jei tai būtina paslaugoms (pavyzdžiui, balso paštui, elektroniniam paštui ir kitoms) teikti <...>“. Apibendrinant, teisinis stebėjimo sistemų saugos tikslais statusas priklauso nuo to, ar jos gali būti laikomos būtinomis saugiam paslaugų tiekimui užtikrinti. Bet kuriuo atveju, į elektroninių paslaugų tiekimo sutartį reikėtų įtraukti sąlygas, numatančias srauto stebėjimą ir laikiną saugojimą, išimtinai paslaugų saugumo užtikrinimui.

77 straipsnis numato, kad Interneto paslaugų tiekėjas privalo įstatymų numatyta tvarka teikti reikalingą informaciją atitinkamoms institucijoms, tačiau apie prievolę rinkti papildomus duomenis ar priemones neužsimenama.

Antruoju atveju, kai stebimas įmonės vidinis tinklas, yra svarbūs keli aspektai. Jei kompanija neužsiima viešųjų elektroninių ryšių ar elektroninių paslaugų tiekimu, jos vidinis tinklas nėra laikomas viešu, todėl elektroninių ryšių įstatymo devintasis skirsnis („Asmens duomenų tvarkymas ir privatumo apsauga“) [28, pp. 40-41] nėra taikomas, ir įmonės tinklą galima stebėti visomis priemonėmis. Tačiau reikia atsižvelgti į tinklą naudojančius klientus ir darbuotojus, nes asmens duomenų apsaugos įstatymas bet kuriuo atveju yra galiojantis.

Lietuvos Respublikos darbo kodeksas nustato, kad darbo tvarką darbovietėje nustato vidinės darbo tvarkos taisyklės, kuriomis remiantis galimas srauto stebėjimas, bet griežto darbo vietos IT sistemų stebėjimo teisinio reglamentavimo nėra, todėl vadovaujamasi teismų praktika. Teismai pabrėžia, kad darbuotojo statusas nepanaikina jo teisės į privatumą net ir naudojant įmonės komunikacijos priemones, todėl nors vidinį įmonės srautą stebėti teisiškai leidžiama, negalima to naudoti asmeninių darbuotojų duomenų rinkimui. Šis darbdavio-darbuotojo santykis nėra griežtai apibrėžtas ir vadovaujamasi proporcingumo principu. [30, pp. 198-203]

Europos Sąjungos teisėje taip pat nėra aiškaus IT saugos priemonių apibrėžimo ir reglamentavimo, todėl vadovaujamasi bendruoju duomenų apsaugos reglamentu [31], kuris draudžia bet kokį nesankcionuotą asmens duomenų rinkimą, automatinį ar ne. Šis reglamentas priimtas 1995

[3] Pranešimas – informacija, kuri perduodama arba kuria, naudojantis viešosiomis elektroninių ryšių paslaugomis, apsikeičia baigtinis viešųjų elektroninių ryšių paslaugų naudotojų skaičius. Pranešimu nelaikoma informacija, perduodama elektroninių ryšių tinklais kaip dalis transliavimo paslaugos, išskyrus tą jos dalį, kurią sudaro individualizuotos informacijos perdavimas identifikuojamam abonentui arba viešųjų elektroninių ryšių paslaugų naudotojui.

[4] Srauto duomenys – duomenys, tvarkomi siekiant perduoti informaciją elektroninių ryšių tinklu ir (arba) tokio perdavimo apskaitai.

m., prieš Interneto paplitimą, todėl nėra tinkamas šiuolaikinėms problemoms spręsti. Teisinių priemonių atnaujinimui, 2016 m. balandžio 14 d. buvo priimtas naujas bendrasis duomenų apsaugos reglamentas [32], kuris įsigalios nuo 2018 m. gegužės 25 d. ir pakeis dabar galiojantį reglamentą. Naujajame reglamente taip pat griežtai ginama teisė į asmens privatumą, draudžiamas nesankcionuotas asmeninių duomenų rinkimas, tačiau 49-oje priežastyje numatoma išlyga, leidžianti tvarkyti asmens duomenis: „valdžios institucijų, kompiuterinių incidentų tyrimo tarnybų, kompiuterių saugumo incidentų tyrimo tarnybų, elektroninių ryšių tinklų bei paslaugų teikėjų ir saugumo technologijų bei paslaugų teikėjų atliekamas asmens duomenų tvarkymas tik tiek, kiek tai yra būtina ir proporcinga siekiant užtikrinti tinklo ir informacijos saugumą, t. y. tinklo ar informacinės sistemos nustatyto patikimumo laipsnio atsparumą trikdžiams arba neteisėtiems ar tyčiniams veiksmams, kuriais pažeidžiamas saugomų ar persiunčiamų asmens duomenų prieinamumas, autentiškumas, vientisumas, konfidencialumas ir susijusių paslaugų, kurias teikia tie tinklai ir sistemos arba kurios per juos prieinamos, saugumą, laikomas teisėtu atitinkamo duomenų valdytojo interesu. Tai galėtų, pavyzdžiui, užkirsti kelią neteisėtai prieigai prie elektroninių ryšių tinklų ir kenkimo programų kodų platinimui, taip pat sustabdyti atkirtimo nuo paslaugos atakas ir neleisti pakenkti kompiuterių bei elektroninių ryšių sistemoms”.

Griežto teisinio reguliavimo, konkrečiai apibrėžiančio tinklo saugos priemonių naudojimą kol kas nėra. Nepavyko rasti ir konkrečių teisminės praktikos pavyzdžių. Tačiau ES ir Lietuvos teisėje bei teismų praktikoje matoma tendencija, siekianti apsaugoti asmens duomenis nuo neteisėto naudojimo, atskleidimo ir saugojimo, o informacijos saugumo užtikrinimo priemonėms jokių išimčių numatyta nėra, todėl tenka vadovautis bendromis teisinėmis normomis, kurios neatspindi šiuolaikinėmis technologijomis paremto pasaulio. Situacija turėtų pasikeisti 2018 m. įsigaliojus naujajam bendram duomenų apsaugos reglamentui (BDAR), kuris pripažįsta ribotą asmens duomenų naudojimą, siekiant užtikrinti IT saugumą ir efektyvų incidentų tyrimą [33, p. 282].

1.3.4 Tinklo saugos priemonių palyginimas pagal kontroles

Apibendrintas šiuolaikinių tinklo saugos priemonių palyginimas pagal jų atitinkamas CIS CSC kontroles, pateikiamas 1.1 lentelėje. Čia pateikiamos tik tos kontrolės, kurias atitinka bent viena tinklo saugos priemonė. Iš apibendrinimo daroma išvada, kad ugniasienė vis dar yra pagrindinė tinklo saugos priemonė, o kitos, nors ir naudojamos vis dažniau, atitinka mažiau kontrolių.

1.1 lentelė. Tinklo saugos priemonių palaikomos kontrolės

	Ugniasienė	**TMIAS/ TIAS**	**PKG**	**TA sistema**
2.10 Fizinis ar loginis programų su padidinta rizika atskyrimas	+	-	-	-
6.2 Audito žurnaliniai įrašai	+	+	+	-
6.3 Detalūs žurnaliniai failai	+	+	+	-

6.4 Pakankamai vietos diske žurnaliniams failams saugoti	+	+	+	-
6.7 Reguliari žurnalinių įrašų peržiūra	+	+	+	-
7.4 Universaliųjų adresų (angl. *URL*) blokavimas pagal sąrašą	+ (programų lygmens)	+	-	-
7.5 Prenumeruoti kenksmingų universaliųjų adresų sąrašo atnaujinimo paslaugą	+ (programų lygmens)	+	-	-
7.6 Visų universaliųjų adresų fiksavimas žurnale	+ (programų lygmens)	+	+	-
7.7 Žinomų kenksmingų nuorodų blokavimas	+ (programų lygmens)	+	-	-
8.2 Kenkėjiškų programų požymių atnaujinimas	-	+	-	-
12.1 Tinklo ribų nustatymas	+	-	-	-
12.3 Prisijungimų blokavimas iš žinomų kenkėjiškų IP adresų	+	-	-	-
12.5 Pilnųjų paketų įrašymas	-	+ (priklausomai nuo nustatymų)	+	-
12.6 Tinklo IAS naudojimas	-	+	-	-

12.7 Tinklo IPS naudojimas	-	+	-	-
12.8 Tėkmių informacijos rinkimas	-	-	-	+
12.9 Programų lygmens ugniasienės naudojimas	+ (programų lygmens)	-	-	-
14.1 Tinklo segmentavimas	+	-	-	-
14.2 Srauto filtravimas tarp virtualių tinklų	+	-	-	-
14.3 Komunikacijos draudimas tarp galinių įrenginių	+	-	-	-

1.3.5 Paketų gaudyklių rinkos apžvalga

Šiame darbe kuriamam integracijos sprendimo pagrindui bus naudojama pilnųjų paketų gaudyklė. Svarbiausi kriterijai, kuriuos turi atitikti PKG:

- Sukauptų paketų indeksavimas greitai paieškai ir analizei;
- Plečiama modulinė architektūra;
- Grafinė vartotojo sąsaja su integruotais analizės įrankiais.

Rinkoje yra nemažai pilnųjų paketų įrašymui skirtų įrankių, nuo paprasčiausių atviro kodo, komandinės eilutės pagrindu veikiančių iki sudėtingų, kompleksinių komercinių sistemų. Daugumos jų veikimas paremtas „libpcap“ („Linux“ operacinėje sistemoje) ir „WinPcap“ („Windows“ OS pritaikyta „libpcap“ versija) bibliotekomis, skirtomis paketų dubliavimui OS branduolyje, jų nukreipimui analizei ar įrašymui į diską.

Šiame darbe pilnųjų paketų analize paremto metodo pagrindu iš anksto pasirinktas „Moloch“, todėl nesiekiama atlikti išsamios rinkos analizės. Toliau tik trumpai apžvelgiamos kelios populiariausios programos rinkoje ir jų galimybių palyginimas su „Moloch“.

1.3.5.1 “Wireshark“

„Wireshark“, anksčiau žinomas kaip „Ethereal“, yra turbūt labiausiai paplitęs ir žinomas paketų įrašymo ir protokolų analizės įrankis. Tai yra įrankių rinkinys, kurį sudaro komandinės eilutės programos („tshark“, „dumpcap“, „pcap-filter“ ir t.t.), skirtos paketų dubliavimui, tekstinei analizei ir

įrašymui į diską kartu su grafine vartotojo sąsaja, skirta interaktyviam duomenų srauto tyrimui, pasitelkiant įvairius filtrus ir automatizuotas analizės priemones.[5]

1.3.5.2 „Stenographer“

Tai *„Google“* sukurta pilnų paketų gaudyklė, orientuota į naudojimą su IAS. Iš kitų paketų gaudyklių *„Stenographer“* išsiskiria tuo, kad beveik neturi analizės priemonių, o didžiausias dėmesys skiriamas kuo spartesniam paketų rašymui į diską, iš kurio, specialių užklausų pagalba, nuskaitomi paketai, atitinkantys norimus parametrus. Oficialiame aprašyme[6] teigiama, kad *„Stenographer“*:

- **skirtas** greitam paketų rašymui į diską (~10 Gbps naudojant tinkamą įrangą);
- **skirtas** disko valdymui, siekiant išsaugoti kuo daugiau paketų (trūkstant vietos, ištrinami seniausi paketai);
- **skirtas** tik mažos dalies paketų skaitymui iš disko (~1 %);
- **neskirtas** sudėtingai srauto analizei (TCP sesijos atstatymui ir pan.);
- **neskirtas** didelės dalies (daugiau nei 1 %) įrašytų paketų skaitymui iš disko, kadangi didžioji dalis disko resursų naudojama rašymui, siekiant neprarasti paketų.

1.3.5.3 „netsniff-ng“

Tai dar vienas *„Linux“* operacinei sistemai skirtas komandinės eilutės programų rinkinys[7], sudarytas iš šių komandų:

- **„netsniff-ng“**: pilnų paketų gaudyklė, veikianti „nulio kopijų“ (angl. *zero-copy*) principu, rašanti paketus į diską standartiniu „pcap“ formatu ir gebanti retransliuoti (angl. *replay*) sukauptus paketus;
- **„trafgen“**: paketų generatorius;
- **„mausezahn“**: dar vienas paketų generatorius, skirtas „Cisco“ operacinės sistemos pagrindu veikiantiems tinklo įrenginiams;
- **„Berkeley Packet Filter compiler“ - „bpfc“** : paketų filtrų kompiliatorius ir asembleris;
- **„ifpps“**: „Linux“ branduolio tinklo statistikos įrankis;
- **„flowtop“**: sujungimų sekimo įrankis;
- **„curvetun“**: IP tunelis;
- **„astraceroute“**: autonominių sistemų tyrimo įrankis.

„netsniff-ng“ labiausiai žinomas būtent dėl pilnų paketų gaudyklės įrankio, kuriuo pavadintas visas įrankių rinkinys. Ši pilnų paketų gaudyklė, dėl savo mažos apimties ir sisteminių reikalavimų, dažnai sutinkama kaip įvairių IAS pagrindas.

1.3.5.4 „PcapDB“

„PcapDB“ yra atviro kodo, paskirstyta (angl. *distributed*), optimizuota paieškai paketų gaudyklė, sukurta kaip konkurentas komercinėms sistemoms[8]. Šioje sistemoje, skirtingai nei daugelyje „pcap“ pagrindu veikiančių paketų gaudyklių, paketai yra suskirstomi sekomis (angl. *flows*) ir indeksuojami. Tam naudojami du diskų masyvai:

- **Indeksavimo diskas**, kuriama laikoma indeksuota srauto informacija. Jei šiam disko tipui priskiriamas daugiau nei vienas diskas, automatiškai sukuriamas RAID1 masyvas;

[5] https://www.wireshark.org/
[6] https://github.com/google/stenographer
[7] http://netsniff-ng.org/
[8] https://github.com/dirtbags/pcapdb

- **Paketų diskas**, į kurį rašomi sugauti paketai. Jei priskiriama daugiau nei vienas diskas, rekomenduojama sukurti RAID5 masyvą.

1.3.5.5 n2disk

Viena populiariausių komercinė pilnų paketų gaudyklė, sukurta „ntop“ kompanijos. Skirta itin didelių srautų išsaugojimui (iki 40 Gbps). Savo veikimo principu „n2disk“ yra panaši į „tshark“ ir kitas atviro kodo paketų gaudykles, tačiau pasižymi aukštu optimizacijos lygiu ir paketų indeksavimu. Pagal maksimalią įrašomo srauto spartą, yra 3 skirtingos programos versijos:

- „n2disk1g“, skirta iki 1 Gbps srauto įrašymui;
- „n2disk5g“, skirta iki 5 Gbps srauto įrašymui;
- „n2disk“, skirta iki 40 Gbps srauto įrašymui.

„ntop“ taip pat siūlo specialų įrenginį – „nBox Recorder“, optimizuotą paketų gaudyklei ir įrašyta „n2disk“ programine įranga.

1.3.5.6 „Moloch“

„Moloch“[9] yra nemokama, atviro kodo[10], lengvai plečiama paketų gaudymo, indeksavimo ir archyvavimo sistema, sukurta Andy Wick ir Eoin Miller, dirbusių „AOL“ kompanijos incidentų reagavimo skyriuje. Iš kitų rinkoje esančių paketų gaudyklių („tcpdump“, „ngrep“, „Wireshark“ ir kitų) „Moloch“ išsiskiria tuo, kad yra pilnas sprendimas, leidžiantis ne tik gaudyti paketus, bet juos indeksuoti, archyvuoti ir analizuoti per saugią žiniatinklio prieigą. „Moloch“ plačiau aprašomas ir tiriamas tolimesniuose skyriuose.

1.3.5.7 Paketų gaudyklių palyginimas

Apibendrintas populiariausių paketų gaudyklių palyginimas pateiktas 1.2 lentelėje. Iš palyginimo daroma išvada, kad „Moloch“ geriausiai atitinka keliamus reikalavimus ir turi vieną laisviausių atviro kodo licencijų.

1.2 lentelė. Pilnųjų paketų gaudyklių palyginimas

Pavadinimas	**Grafinė vartotojo sąsaja**	**Bazinis indeksavimas**	**Protokolo lygmens indeksavimas**	**Plečiama architektūra**	**Integruoti srauto analizės įrankiai**	**Licenzija**
„Wireshark“	Yra	Yra	Nėra	Nėra	Yra	GPLv2
„Stenographer“	Nėra	Yra	Nėra	Nėra	Nėra	Apache v2
„netsniff-ng“	Nėra	Nėra	Nėra	Nėra	Yra	GPLv2
„PcapDB“	Nėra	Yra	Nėra	Yra	Nėra	BSD 2
„n2disk“	Nėra	Yra	Nėra	Nėra	Nėra	Komercinė
„Moloch“	Yra	Yra	Yra	Yra	Yra	Apache v2

[9] http://molo.ch/

[10] https://github.com/aol/moloch

1.4 Analizės išvados

Analizės dalyje apžvelgti pagrindiniai incidentų valdymo ir tyrimo standartai bei kiti struktūrizuoti dokumentai (angl. *framework*). Išvardintos pagrindinės kontrolės ir jų grupės, pagal kurias bus lyginamos tinklo saugos priemonės ir tiriamos jų pritaikymo galimybės.

Aptartos pagrindinės šiuolaikinės tinklo saugos priemonės ir įvesta jų klasifikacija pagal tipą, naudojamą tinklo stebėjimo metodą, konfigūraciją ir kontrolių priskyrimą, kuri leidžia priemones lyginti tarpusavyje. Apžvelgti du pagrindiniai tinkle stebėjimo metodai (paketų ir tėkmių analizavimas), kuriuos svarbu teisingai pasirinkti, priklausomai nuo siekiamų rezultatų. Išnagrinėjus Lietuvos ir ES teisinę bazę, nustatyta problematinių srauto stebėjimo sričių, į kurias privaloma atsižvelgti norint išvengti teisinių problemų, naudojant tinklo saugos sistemas, paremtas srauto analizavimu. Atlikus rinkos apžvalgą, pagrįstas „Moloch“ ne tik kaip paketų gaudyklės, bet kaip ir galingos analizės platformos pasirinkimas.

Palyginus tinklo saugos priemones tarpusavyje, nustatyta pagrindinė problema – dauguma jų yra orientuota į incidentų prevenciją ir aptikimą, o analizės priemonių visai nėra arba jos minimalios, nors šiuolaikinė tinklo saugos metodika ir pabrėžia incidentų tyrimo būtinybę ir svarbą. Atsižvelgus į tokias analizės išvadas, keliamas pagrindinis šio darbo tikslas – sukurti tokį tinklo saugos metodą, kuris užpildytų incidentų tyrimo nišą.

Tolimesniame darbe didžiausias dėmesys bus skiriamas incidentų tyrimui, naudojantis tinklo saugos priemonių surinkta informacija. Skirtingos tinklo saugos priemonės skirtos skirtingoms kontrolėms padengti, generuoja skirtingą informaciją ir jos kiekį, todėl svarbu nustatyti, kokiu atveju kokią priemonę tikslinga naudoti.

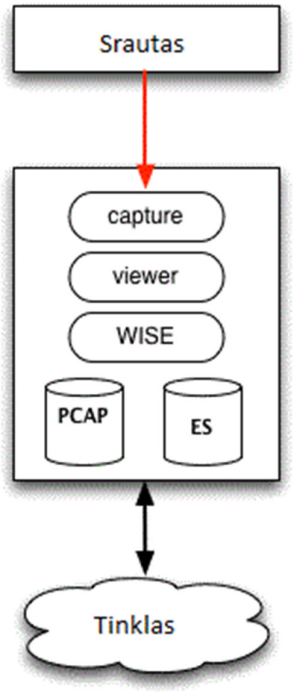
2. „MOLOCH“ METODO TAIKYMAS IR TYRIMAS

Pats „Moloch“ nėra tinklo saugos priemonė, nes savaime tinklo neapsaugo, o tik įrašo srautą ir suteikia įrankius jį analizuoti. Nors „Moloch“ neskirtas pakeisti IAS ar kitas tinklo saugos priemones, tačiau gali puikiai jas papildyti. Naudojami standartiniai PCAP paketų saugojimo ir JSON sesijos informacijos formatai, leidžia „Moloch“ praplėsti ar integruoti su kitomis sistemomis.[11]

Struktūriškai „Moloch“ sistema susideda iš 3 pagrindinių komponentų, kurie tuo pačiu yra ir atskiri sisteminiai procesai:

1. **capture** – „C“ kalba parašyta daugiagijinė (angl. *threaded*) aplikacija, kuri stebi srautą, PCAP formatu įrašo pagautus paketus į diską ir siunčia SPI (angl. *System Packet Interface*) meta duomenis į **elasticsearch** procesą.
2. **viewer** – „node.js“ aplikacija, kuri persiunčia PCAP duomenis iš sensoriaus ir suteikia prieigą per žiniatinklį.
3. **elasticsearch** – duomenų bazės indeksavimo ir paieškos sistema.[12]
4. WISE (angl. *With Intelligence See Everything*) neprivalomas modulis skirtas papildomų duomenų persiuntimui iš kitų priemonių (pvz., vietinių ar nutolusių laikmenų, kitų, komercinių sprendimų).[13]

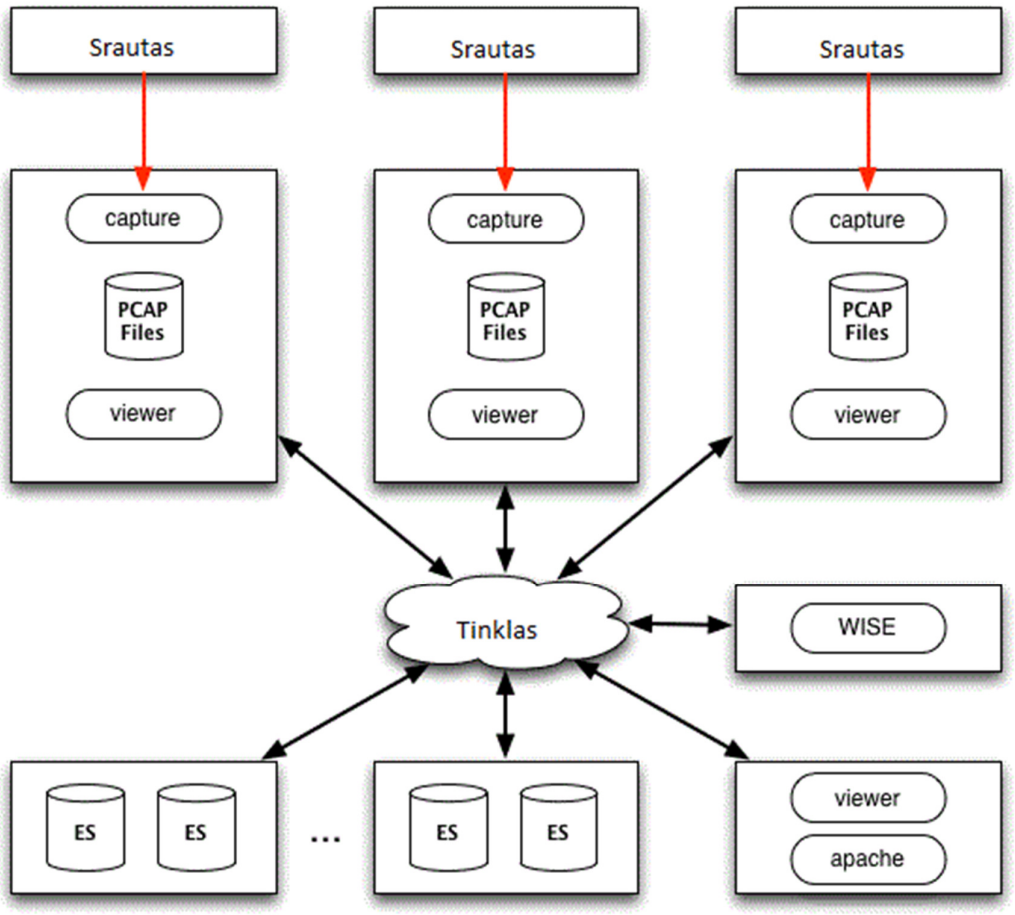
„Moloch“ išdėstymo konfigūracija yra lanksti, tinkanti tiek nedideliam vietiniam tinklui, tiek didelių, nutolusių tinklų stebėjimui. Paprasčiausiu atveju naudojama taškinė konfigūracija ir visi 3 komponentai yra diegiami į vieną serverį, kuriame stebimas vienas srautas (žr. 2.1 pav.)



2.1 pav. Taškinės „Moloch“ konfigūracijos schema

[11] https://github.com/aol/moloch#what-is-moloch
[12] https://github.com/aol/moloch#id2
[13] https://github.com/aol/moloch/wiki/WISE

Sudėtingesniu atveju (žr. 2.2 pav.) naudojami keli serveriai – sensoriai, kuriuose veikia paketų gaudymo (**capture**) ir peržiūros procesai (**viewer**), bei kuriuose PCAP formatu saugojami paketai. Atskirame serveryje ar serveriuose veikia **elasticsearch** procesai. Žiniatinklio prieigai gali būti diegiamas "Apache" serveris.[14] Gali būti ir daugiau „Moloch" išdėstymo tinkle konfigūracijų, tačiau šiame darbe apsiribojama dvejomis išvardintomis.



2.2 pav. Paskirstytos „Moloch" konfigūracijos schema

„Moloch" sisteminiai resursų reikalavimai priklauso nuo naudojamos konfigūracijos ir stebimo srauto. Oficialiai pateikiami reikalavimai 1 Gbps srauto stebėjimui[15]:

1. Kiekvienam **capture** ir **viewer** procesui:
   - Dedikuota tinklo sąsaja valdymui;
   - 1 dedikuotas procesoriaus branduolys operacinei sistemai ir po vieną kiekvienai stebimai tinklo sąsajai;
   - 10 GB operatyviosios atminties kiekvienai stebimai tinklo sąsajai;
   - 10 TB/dienai vietos diske PCAP failams laikyti.

2. Kiekvienam **elasticsearch** procesui:
   - 30 – 64 GB operatyviosios atminties;

[14] https://github.com/aol/moloch/wiki/Architecture
[15] https://github.com/aol/moloch#advanced-configuration

- Rekomenduojama naudoti atskirą serverį nuo **capture** ir **viewer** procesų.

2.1 „Moloch“ metodo taikymas

Pilnųjų paketų gaudymo sistemos įrengimas yra techniškai sudėtinga ir daug resursų reikalaujanti užduotis, todėl pirmiausia svarbu nustatyti saugotiną perimetrą ir sensorių išdėstymą taip, kad būtų generuojama mažiausi srauto duomenų kiekiai, tačiau neprarandama incidentų analizei svarbi informacija. Vienareikšmiškai atsakyti į klausimą: „kaip išdėstyti sensorius?“, neįmanoma, kadangi tai priklauso nuo konkretaus tinklo topologijos ir stebimo objekto specifikos, tačiau svarbiausia atsižvelgti į šiuos veiksnius:

- **Teisėtumas**. Ar už srauto stebėjimą atsakingas subjektas turi teisę stebėti ir kaupti perduodamus duomenis? Ar perduodami duomenys neturi kažkokių specialių apribojimų ar reikalavimų?
- **Vientisumas**. Ar galima stebimą objektą apibrėžti vienu perimetru? Ar visas objektas yra viename fiziniame ar loginiame tinkle?
- **Pasitikėjimas**. Kur yra galimi grėsmės sukėlėjai (angl. *threat actors*)? Ar saugomasi tik nuo išorės grėsmių sukėlėjų, ar tik nuo vidinių, ar nuo abiejų?
- **Skaidrumas**. Ar stebimi srauto duomenys nėra šifruoti? Ar informacija nepaslėpiama, neiškreipiama?

2.1.1 Teisėtumo veiksnys

Prieš diegiant „Moloch“, būtina nustatyti, ar renkamos informacijos saugojimas nepažeidžia teisės normų (žr. skyrelį „1.3.3 Srauto stebėjimą reglamentuojantys įstatymai“) ir ar galimai sukaupta informacija nesukelia papildomų rūpesčių bei reikalavimų. Pavyzdžiui, PCI DSS (angl. *The Payment Card Industry Data Security Standard*) standartas, skirtas mokėjimo kortelės turėtojų duomenų apsaugai, nurodo, kad bet kuri sistema, perduodanti, sauganti ar kitaip apdorojanti mokėjimo kortelės turėtojų duomenis, automatiškai patenka į standarto reglamentuojamą aplinką (angl. *environment*). Tai reiškia, kad norint atitikti PCI DSS keliamus reikalavimus, mokėjimo kortelės turėtojų duomenis valdantis subjektas kartu turėtų užtikrinti ir visos „Moloch“ sistemos atitikimą standartui bei atlikti reikiamus patikrinimus ar auditus, kas sąlygotų didesnes išlaidas.

2.1.2 Vientisumo veiksnys

Visą stebimą objektą reikia suskirstyti į vientisas zonas, kuriose stebimi visi įėjimai ir išėjimai. „Moloch“ sistemos reikalaujami resursai priklauso nuo stebimo srauto kiekio, todėl apibibrėžiant zonas taip pat reikia stebėti, kad nebūtų nereikalingų sensorių (pavyzdžiui, tarp dviejų patikimų serverių). Zonų viduje neturi likti grėsmės sukėlėjų.

2.1.3 Pasitikėjimo veiksnys

Visi grėsmės sukėlėjai nuo stebimo objekto turėtų būti atskiriami bent vienu sensoriumi. Neturi likti neapsaugotų vektorių tarp grėsmės sukėlėjų ir stebimos informacijos. Tai galioja ir vidiniams grėsmės sukėlėjams, todėl, jei, pavyzdžiui, stebima įmonės vidinė duomenų bazė, kurioje laikomi konfidencialūs duomenys ir kuriems keliami griežti saugumo reikalavimai, tokiu atveju, šią duomenų bazę reikėtų išskirti atskira zona, apsaugančia ir nuo vidinio duomenų nutekėjimo.

2.1.4 Skaidrumo veiksnys

Prieš pradedant kaupti pilnus paketus, svarbu įsitikinti, kad jie bus naudingi, nes nėra tikslo kaupti duomenis, kurių vis tiek nebus galima panaudoti analizei. Tai galėtų būti šifruotas srautas be galimybės jį iššifruoti, pavyzdžiui, virtualus privatusis tinklas (VPT). Tokiu atveju, sensorių reikėtų

statyti prieš duomenų užšifravimo įrenginį įėjime ir už duomenis iššifruojančio įrenginio (VPT šliuzo) išėjime.

Kitas srauto informaciją iškraipantis veiksnys gali būti tinklo įrenginiai. Pavyzdžiui, ugniasienė, kurioje naudojama NAT funkcija, pakeičia vidinius tinklo adresus į vieną ar daugiau išorinių, todėl pastatant sensorių prieš tokią ugniasienę be papildomos informacijos bus neįmanoma nustatyti, kuriam konkrečiam įrenginiui vidiniame tinkle srautas buvo skirtas.

2.1.5 Resursų reikalavimai

Diegiant saugos sistemas, prieš tai svarbu paskaičiuoti, ar tai yra ekonomiškai prasminga. Informacija nėra vienodos vertės, o vienas esminių IT saugos principų teigia, kad saugos priemonių kaina neturi viršyti saugomos informacijos vertės (reikia pabrėžti, kad vertė čia nebūtinai suprantama pinigine prasme, pavyzdžiui, jautrūs asmeniniai duomenys dažnai nėra vertingi pinigine prasme, tačiau gali turėti didelę reikšmę ir padaryti daug žalos juos neleistinai panaudojus ar paviešinus). IT saugos priemonių kaštai susideda iš dviejų pagrindinių dalių:

- Diegimo kaštai (trumpalaikiai): programinės ir aparatinės įrangos kaina, pasirengimo ir diegimo darbai.
- Eksploatacijos kaštai (ilgalaikiai): programinės ir aparatinės įrangos licencijų atnaujinimas, prižiūrinčio personalo užmokestis, suvartojamos elektros, aparatinės įrangos atnaujinimas, taisymas ir t.t.

2.2 „Moloch“ ir kitų tinklo saugos priemonių reakcijos į incidentus tyrimas

Tinklo saugos priemonių pagrindinė paskirtis, priklausomai nuo jos tipo, yra incidentų prevencija, aptikimas, blokavimas ir analizė, todėl šioje dalyje bus tiriama konkrečių priemonių reakcija į labiausiai paplitusius incidentų tipus (atakas) kiekviename šių etapų. Kompiuterių tinklo incidentus galima skirstyti pagal tipą, kurių labiausiai paplitę yra [34]:

- **Kenkėjiškos programos** (angl. *malware*), kurios infekuoja aukos kompiuterį ir dažnai įtraukia į kenkėjų tinklus (angl. *botnet*) savininkui to nežinant. Infekuotos sistemos dažnai naudojamos kaip platforma kitoms atakoms vykdyti. Tai atakų kategorija, į kurią taip pat įeina ir virusai (savarankiškai plintančios kenksmingos programos), šnipinėjimo programos ir kitos, tačiau visos jos naudoja kompiuterių tinklą plitimui, komandų gavimui iš kontroliuojančio serverio, pavogtos informacijos išsiuntimui ir t.t. Pastaruoju metu ypač išpopuliarėjo išpirkos reikalaujantys virusai (angl. *ransomware*), kurie patekę į sistemą, užšifruoja visus prieinamus duomenis ir už jų iššifravimą reikalauja išpirkos. Kartais tokie virusai išskiriami į atskirą kategoriją, tačiau jų veikimo principas toks pats kaip kitų kenkėjiškų programų, todėl šiame darbe pateikiamas bendroje kenkėjiškų programų kategorijoje;
- **Prieigos sutrikdymo atakos** (angl. *Denial of Service – DoS*) ir **paskirstytos sutrikdymo atakos** (angl. *Distributed Denial of Service – DDoS*), kuriomis siekiama sutrikdyti sistemos ar tinklo darbą, sugadinti prieigą prie paslaugų. Dažnai naudojamos kitoms atakoms užmaskuoti;
- **Sukčiavimas** (angl. *phising*) skirtas konfidencialios informacijos išviliojimui apgaulės būdu, pavyzdžiui, siunčiant elektroninį laišką ir apsimetant auditoriumi prašant įmonės finansinių duomenų. Sudėtingesniais sukčiavimo atvejais gali būti naudojami netikri internetiniai puslapiai, imituojantys kitus, žinomus puslapius, dažniausiai siekiant išvilioti prisijungimo duomenis, pavyzdžiui, internetinio banko ar elektroninio pašto paskyros;
- **Brukalų siuntimas** (angl. *spam*) griežtu požiūriu nėra IS incidentas, nes nepažeidžiamas nei vienas iš trijų pagrindinių informacijos saugos principų (konfidencialumas, vientisumas, prieinamumas), tačiau apkrauna sistemas, užteršia elektroninio pašto dėžutes, taip pat dažnai naudojamas sukčiavimo laiškams platinti.

Brukalai paprastai yra aptinkami ir filtruojami pašto serveryje esančia specialia programine įranga;

- **Infekuotų kompiuterių tinklai** (angl. *botnet*) yra naudojami kaip platforma įvairioms atakoms (DDoS, duomenų vagystei, brukalų siuntimui ir t.t.) vykdyti be jų savininkų žinios ar sutikimo. Tokie tinklai paprastai valdomi iš centrinio valdymo serverio (angl. *comand and control Server*), todėl paprasčiausias būdas su jais kovoti yra šių kontrolinių serverių blokavimas ir naikinimas;
- **Programų pažeidžiamumų išnaudojimas** (angl. *exploit*) atakos dažniausiai vyksta sesijos ir aukštesniuose OSI lygmenyse, išnaudojant programavimo klaidas (angl. *bug*) neteisėtai prieigai gauti ir programos ar visos sistemos darbo sutrikdymui;
- **Vidinių sistemos vartotojų keliama grėsmė** (angl. *insider threat*) paprastai pasireiškia, kai teisėtą prieigą turintis vidinis vartotojas siekia pasisavinti ar išsiųsti konfidencialią informaciją iš apsaugoto tinklo. Tai vienas pavojingiausių incidentų, kadangi potenciali žala, atskleidus konfidencialius įmonės duomenis, nutekinus intelektinę nuosavybę ar pan., gali būti labai didelė. Kadangi vidinių sistemos vartotojų keliama grėsmė yra paplitusi ir pavojinga, su ja kovoti yra kuriama speciali programinė įranga – duomenų praradimo apsauga (angl. *data loss protection*).

Šiame darbe, naudojantis sukurta testine aplinka, bus atkuriama atakų eiga ir tiriama, kaip kiekviena sistema į jas reaguoja, kokią informaciją surenka ir kaip sėkmingai gali būti atliekama incidento analizė, pasinaudojus šia informacija.

2.2.1 Tiriamų sistemų ir aplinkos aprašymas

Tyrimui atrinkta po vieną tinklo saugos priemonę iš kiekvienos kategorijos: ugniasienių, IAS, tėkmių analizės ir srauto analizės. Renkamasi buvo iš nemokamų, atviro kodo sistemų, atsižvelgiant į naudojimo paplitimą ir prieinamos dokumentacijos kiekį. Atrinktų priemonių santrauka pateikta 2.1 lentelėje.

2.1 lentelė. Tinklo saugos priemonės, atrinktos tyrimui

Priemonė / Požymis	**„ pfSense“**	**„nfcapd“, „nfdump“, „nfsen“**	**„Security Onion“**			**„Moloch“**
			„Suricata“	**„Snort“**	**„Bro“**	
Klasė	Tinklo (perimetro) Ugniasienė	Tėkmių analizatorius	Požymių aptikimu paremta TMIAS	Požymių aptikimu paremta TMIAS	Anomalijų aptikimu paremta TMIAS	IPA
Sisteminiai reikalavimai	žemi	žemi	aukšti	aukšti	aukšti	aukšti
Palaikomos konfigūracijos	Taškinė	Taškinė, centralizuota	Taškinė, centralizuota	Taškinė, centralizuota	Taškinė, centralizuota	Taškinė, paskirstyta, centralizuota
Versija	2.4.3	1.3.6	14.04.5.13			0.20.2
Teisinis statusas Lietuvoje ir EU	Leistinas	Neaiškus	Neaiškus	Neaiškus	Neaiškus	Neaiškus
Kaina	nemokama	nemokama	nemokama	nemokama	nemokama	nemokama
Licenzija	Apache 2.0	BSD	GPLv2	GPLv2	BSD	Apache 2.0

Testinė aplinka kuriama virtualizacijos technologijų pagalba. „VirtualBox“ programa sukuriamos virtualios mašinos, kuriose instaliuojama reikiama programinė įranga. Visas darbe naudojamų virtualių mašinų sąrašas pateiktas 2.2 lentelėje. Šios mašinos tarpusavyje nesusiejusios, veikia savarankiškai ir jų tyrimas atliekamas paeiliui.

2.2 lentelė. Virtualios testinės aplinkos aprašymas

Eil. Nr.	VM pavadinimas	OS	Pagrindiniai moduliai	Sisteminiai parametrai
1	Moloch_standalone	Ubuntu Server 16.04 LTS	*capture*, *viewer, elasticsearch*	4 GB RAM, 2 CPU, 30 GB HDD
2	pfSense	FreeBSD	*pfSense*	4 GB RAM, 2 CPU, 30 GB HDD
3	SecOnion	Ubuntu Server 16.04 LTS	*netsniff-ng, Surricata, Snort, Bro*	8 GB RAM, 4 CPU, 30 GB HDD
4	nfsen	Ubuntu Server 16.04 LTS	*nfcapd, nfdumo, nfsen*	4 GB RAM, 2 CPU, 30 GB HDD

Incidentų aptikimo galimybių tyrimas atliekamas keliais būdais. Incidentai, kuriuos galima nesunkiai atkartoti, tiriami parsisiuntus[16] įrašytus incidentų srauto duomenis PCAP formatu ir juos importuojant į tiriamą sistemą virtualioje aplinkoje. „nfsen“ atveju, naudojama tėkmių informacija, todėl prieš importuojant srauto duomenis reikia konvertuoti. Tam naudojama „nfsen“ įrankių pakete esanti **softflowd** komanda. Tokiu būdu tirtas kenkėjiškų programų aptikimas, siunčiamų laikmenų atstatymo matomumo ir atkūrimo galimybės (imituojant vidinių vartotojų keliamas grėsmes), elektroninio laiško siuntimas (tiriant, ar matomas laiško turinys ir įtartinos nuorodos jame), infekuotų kompiuterių tinklų nustatymas.

Incidentai, kurių atkartojimo galimybės yra ribotos, tiriami teoriškai, skaitant dokumentaciją ir tikrinant, ar aptinkami susiję požymiai. Tokių incidentų pavyzdžiai yra paskirstyta sutrikdymo ataka, brukalų siuntimas, programų pažeidžiamumų išnaudojimas.

2.2.2 „pfSense“ ugniasienė

„pfSense“ yra „FreeBSD“ operacinės sistemos pagrindu sukurta nemokama, atviro kodo ugniasienė. Lankstus diegimo būdas leidžia „pfSense“ naudoti tiek kaip programinę įrangą bendros paskirties fiziniame ar virtualiame serveryje, tiek diegti į specializuotą aparatinę įrangą. Tai nėra vien perimetro apsaugos ugniasienė, „pfSense“ dažnai naudojama kaip bevielio tinklo prieigos taškas (angl. *Wireless Access Point*), DHCP serveris, DNS serveris, taip pat galima įdiegti IAS kaip papildomus paketus, tačiau šiame tyrime apsiribojama tik perimetro ugniasienės galimybėmis.

„pfSense“ paprastai naudojama taškinėje konfigūracijoje, diegiant ją į fizinį, virtualų, debesijos (angl. *cloud*) serverį ar naudojant kartu su specialia aparatine įranga. Baziniai sisteminiai reikalavimai, palyginus su kitomis šiuolaikinėmis priemonėmis, yra maži:

- 1 GHz branduolys;
- 1 GB operatyviosios atminties.

„pfSense“ pateisina savo kaip ugniasienės galimybes ir yra gera prevencijos priemonė nuo daugelio tipo atakų, tačiau tuo jos panaudojimas incidentų valdyme ir apsiriboja, nes nėra

[16] https://www.hybrid-analysis.com/recent-submissions?filter=file

automatinės incidentų aptikimo ir blokavimo galimybės, suteikiamos tik minimalios incidentų analizės priemonės – įvykio žurnalo įrašai su prisijungimų informacija. Detalesnis „pfSense“ tyrimo aprašymas pateiktas 2.3 lentelėje.

2.3 lentelė. „pfSense“ incidentų valdymo galimybių tyrimas

Etapas / Tipas	Prevencija	Aptikimas	Blokavimas	Analizė
Kenkėjiškos programos	**Ribota**. Gali padėti stabdyti automatinį plitimą uždarant kenkėjiškos programos plitimui naudojamus prievadus, tačiau bendru atveju, tai nėra efektyvi priemonė prieš kenkėjiškas programas.	**Ribota**. Aktyvios incidentų aptikimo galimybės nėra, iš įvykių žurnalų galima matyti tik bandymus pažeisti iš anksto nustatytas prevencijos taisykles.	**Nėra.**	**Ribota**. Įjungus išsamaus įvykių registravimą į žurnalą, galima matyti, susijungimų adresus, prievadus ir laikus.
Paskirstytos sutrikdymo atakos	**Nėra.** Kadangi ataka vykdoma iš daug šaltinių, neįmanoma jų visų užblokuoti. Taip pat atakos mąstai kartais būna tokie dideli, kad perkrauna bet kokias apsaugos priemones.			
Sukčiavimas	**Nėra.** Sukčiavimas vykdomas aukštesniuose OSI lygmenyse ir naudojant socialinę inžineriją, todėl ugniasienės, dirbančios tinklo lygmenyje, yra neefektyvios prieš tokio tipo atakas.			
Brukalų siuntimas	**Ribota**. Galima užblokuoti žinomus brukalus siuntinėjančius serverius ir pašto dėžutes, tačiau jų nuolat atsiranda naujų, todėl tokia priemonė nėra efektyvi.			

Infekuotų kompiuterių tinklai	**Ribota**. Galima užblokuoti žinomus valdymo serverius, tačiau jų adresas gali keistis, todėl tokia priemonė nėra efektyvi.			
Programų pažeidžiamumų išnaudojimas	**Nėra**. Tokio tipo atakos vykdomos aukštesniuose OSI lygmenyse, todėl ugniasienė prieš jas yra neefektyvi.			
Vidinių sistemos vartotojų keliama grėsmė	**Ribota**. Galima užblokuoti prieigą prie tam tikrų resursų, tačiau informaciją nesunkiai galima nutekinti kitais kanalais, todėl ugniasienė prieš tokio tipo atakas yra neefektyvi.			

2.2.3 „Security Onion“ saugos priemonių rinkinys

„Security Onion“ yra „Ubuntu“ operacinės sistemos modifikacija, skirta darbui su tinklo sauga, stebėjimu ir incidentų tyrimui. „Security Onion“ yra tinklo saugos priemonių, analizės įrankių ir būsenos stebėjimo įrankių rinkinys (angl. *Network Monitoring System*), į kurį surinkti populiariausi atviro kodo sprendimai.

„Security Onion“ sudaro trys pagrindinės posistemės:

- Pilnų paketų gaudyklei naudojamo „netsniff-ng“[17];
- Incidentų aptikimo sistemos;
- Analizės įrankiai.

„Security Onion“ sudaro kelios skirtingų klasių IAS – trijų tipų tinklo incidentų aptikimo sistemos ir viena serverio incidentų aptikimo sistema:

- Požymio aptikimu paremtomis TMIAS naudojamos „Snort“[18] ir „Suricata“[19] sistemos;
- Anomalijų aptikimu paremta TMIAS naudojama „Bro“[20] sistema;
- SIAS naudojama „OSSEC“[21] sistema.

Tarp „Security Onion“ pateikiamų įrankių yra įvykių analizės programa „Squil“[22], palengvinanti IAS sistemų sugeneruotų incidentų ir įvykių stebėjimą bei analizavimą. „Squert“[23]

17 http://netsniff-ng.org/
18 http://snort.org/
19 http://suricata-ids.org/
20 http://bro-ids.org/
21 http://www.ossec.net/
22 http://sguil.sourceforge.net/

programa leidžianti prisijungti prie „Squil“ duomenų bazės nuotoliniu būdu. „ELSA“ (angl. *Enterprise Log Search and Archive*) skirta palengvinti darbą su sisteminiais įrašais (angl. *logs*), generuoti ir išsiųsti įspėjamuosius pranešimus (angl. *alerts*).

Architektūriškai „Security Onion“ susideda iš dviejų dalių – sensoriaus, kuriame renkami pilni paketai ir serverio, kuriame atliekama surinktų paketų analizė. Kaip ir „Moloch“, „Security Onion“ konfigūracija yra lanksti, pritaikoma įvairaus dydžio ir sudėtingumo tinklams. Paprasčiausia konfigūracija yra vieno fizinio įrenginio ar vienos virtualios mašinos, kurios atveju tiek sensorių, tiek serverio procesai veikia lygiagrečiai. Centralizuotos konfigūracijos atveju sensoriai yra išdėstomi atskiruose įrenginiuose ar virtualiose mašinose, o analizė atliekama centriniame serveryje. Hibridinė konfigūracija yra abiejų anksčiau aprašytų konfigūracijų mišinys, kai viename įrenginyje ar virtualioje mašinoje veikia serveris ir sensoriai, tačiau prie to dar prijungti papildomi, išoriniai sensoriai.

Sisteminiai „Security Onion“ reikalavimai varijuoja, priklausomai nuo to, kokie servisai naudojami, kiek ir kokio intensyvumo srautai stebimi, kiek sensorių naudojama ir kitų tinklo bei srauto parametrų. Oficialiai pateikiamas orientacinis sisteminių reikalavimų pavyzdys vienam 1 Gbps pilnai prisotintam srautui stebėti, kai naudojamos „Snort“ ir „Bro“ IAS:

- 10 procesoriaus branduolių (5 skirti „Snort“ procesams ir 5 skirti „Bro“ procesams);
- 128 – 256 GB operatyviosios atminties;
- 10 TB/dienai vietos diske.

„Security Onion“ incidentų valdymo galimybių tyrimo rezultatai pateikti 2.4 lentelėje.

2.4 lentelė. „Security Onion“ incidentų valdymo galimybių tyrimas

Etapas / Tipas	Prevencija	Aptikimas	Blokavimas	Analizė
Kenkėjiškos programos	**Ribota**. „Security Onion“ neturi ugniasienės funkcionalumo, tačiau dalis prevencijos gali būti sukonfigūruota IAS moduliuose (pavyzdžiui, uždrausti IP adresai).	**Yra**. Kenkėjiškos programos gali būti aptiktos tiek pagal požymius, tiek pagal anomalijas.	**Yra.** Aptikus kenkėjišką programą, jos prisijungimai gali būti blokuojami automatiškai.	**Ribota**. Nors „Security Onion“ turi nemažai analizės priemonių, tačiau dauguma jų skirtos IAS įvykių ir žurnalų analizei. Tinklo srauto analizė paremta „Wireshark“ programa, todėl paveldėja ir jos trūkumus – lėta, reikalaujanti daug rankinio darbo ir netinkanti didelių duomenų kiekių analizei.
Paskirstytos sutrikdymo atakos		**Yra**. Tokio tipo atakos nesunkiai nustatomos pagal stipriai išaugusį tinklo srautą į vieną ar kelis prisijungimo taškus.	**Nėra.** Kadangi ataka vykdoma iš daug šaltinių, neįmanoma jų visų užblokuoti. Taip pat atakos mąstai kartais būna tokie dideli, kad perkrauna bet kokias apsaugos priemones.	
Sukčiavimas		**Ribota.** Galima aptikti kai kuriuos sukčiavimo atvejus pagal įtartinas nuorodas, tačiau tai nėra efektyvi priemonė.	**Ribota**. Galima blokuoti aptiktas įtartinas nuorodas, tačiau tai nėra efektyvi priemonė.	

[23] http://www.squertproject.org/

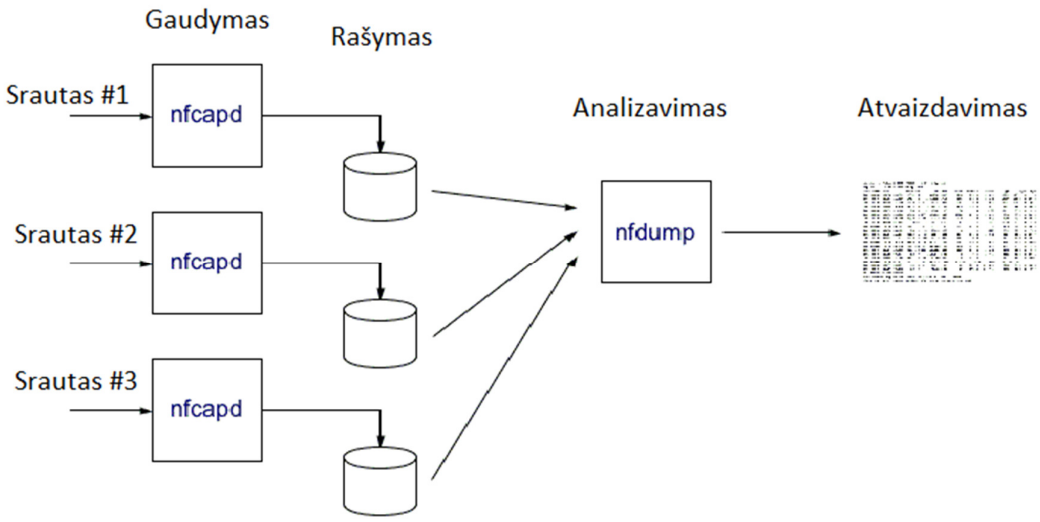
Brukalų siuntimas		**Ribota.** Galima aptikti kai kuriuos brukalų serverius pagal žinomus IP adresus, tačiau tai nėra efektyvi priemonė.	**Ribota.** Galima blokuoti žinomus brukalus siunčiamus serverius, tačiau tai nėra efektyvi priemonė.	
Infekuotų kompiuterių tinklai		**Yra.** „Security Onion“ gan efektyviai aptinka ir blokuoja kontroliuojančius serverius, pagal požymius ar anomalijas.		
Programų pažeidžiamumų išnaudojimas		**Yra.** „Security Onion“ gali aptikti ir blokuoti žinomus kenksmingus paketų turinius (angl. *payload*) bei aptikti anomalijas protokolų ar programų lygmenyje.		
Vidinių sistemos vartotojų keliama grėsmė		**Ribota.** Galima aprašyti taisykles, pagal kurias atskiriama konfidenciali informacija, tačiau tai nėra standartinis funkcionalumas, ir reikalauja specifinių sistemos bei pačios informacijos turinio žinių.		

2.2.4 „nfcapd“, „nfdump“ ir „nfsen“ rinkinys

Kombinuotas „nfcapd“, „nfdump“ ir „nfsen“ sprendimas[24] (žr. 2.3 pav.) savo paskirtimi šiek tiek panašus į „Moloch“, tačiau vietoj išsamaus paketų analizės metodo naudojamas tėkmių analizės metodas. Gali būti naudojamas tiek taškinėje konfigūracijoje, tiek centralizuotoje.

„nfcapd“ modulis yra sensorius, kuris renka tėkmių informaciją, gaunamą iš tinklo įrenginio (maršrutizatoriaus, šakotuvo), ir įrašo į laikmenas. Kiekvienos tėkmės stebėjimui reikia atskiro „nfcapd“ proceso. „nfdump“ yra „C“ kalba parašytas komandinės eilutės įrankis, skirtas analizuoti ir filtruoti „nfcapd“ surinktą tėkmių informaciją. „nfsen“ yra skirtas surinktai tėkmių informacijai atvaizduoti grafiškai. [35]

[24] http://nfsen.sourceforge.net/



2.3 pav. Taškinės konfigūracijos „nfcapd“, „nfdump“ ir „nfsen“ sprendimo schema

Originaliai numatyta taškinė (vieno serverio) konfigūracija, kurioje galima stebėti kelis prievadus skirtingais „nfcapd“ procesais ir analizuoti vienu „nfdump“ procesu. Galima ir centralizuota modifikacija, kai sensoriai išdėstomi atskiruose įrenginiuose ar virtualiose mašinose, juose leidžiami „nfcapd“ procesai, o serveryje veikiantys „nfdump“ ir „nfsen“ analizuoja ir atvaizduoja surinktą tėkmių informaciją, tačiau ji nėra oficialiai palaikoma[25].

Sisteminiai reikalavimai, palyginus su pilnų paketų analizavimo metodą naudojančiomis sistemomis, nėra dideli, didžiausia įtaka veikimui turi laikmenų įrašymo sparta į diską. Pateikiama pavyzdinė sistema [35] 1 Gbps srauto tėkmių analizei ir saugojimui:

- Dviejų branduolių procesorius po 3 GHz;
- 2 GB operatyviosios atminties;
- 25 GB/dienai vietos diske.

„Nfsen“ incidentų valdymo galimybių tyrimo rezultatai pateikti 2.5 lentelėje.

2.5 lentelė. „Nfsen“ incidentų valdymo galimybių tyrimas

Etapas / **Tipas**	**Prevencija**	**Aptikimas**	**Blokavimas**	**Analizė**
Kenkėjiškos programos	**Nėra.**	**Ribotas.** Galima aprašyti taisykles (pavyzdžiui, stipriai išaugęs sujungimų skaičius, jungimasis su tam tikrais adresais ir t.t.), pagal kurias parodomas pranešimas ar perspėjamas	**Nėra.**	**Ribota.** Kenkėjiškas programas galima nustatyti pagal įtartinus sujungimus.
Paskirstytos sutrikdymo atakos				**Yra**. Tokio tipo atakos nesunkiai nustatomos pagal stipriai išaugusį tinklo srautą į vieną ar kelis

[25] https://github.com/vytautas/nfdump

		administratorius (pvz. elektroniniu paštu).		prisijungimo taškus.
Sukčiavimas				**Ribota.** Kai kuriuos atvejus galima aptikti pagal įtartinas nuorodas.
Brukalų siuntimas				**Ribota.** Galima nustatyti tik žinomus brukalus siunčiančius serverius.
Infekuotų kompiuterių tinklai				**Yra.** Pagal įtartinus sujungimus galima nustatyti valdantįjį serverį.
Programų pažeidžiamumų išnaudojimas				**Ribota.** Galima nustatyti, tik kur buvo jungtasi iš tam tikro šaltinio.
Vidinių sistemos vartotojų keliama grėsmė				**Ribota.** Galima nustatyti, tik kur buvo jungtasi iš tam tikro šaltinio.

2.2.5 „Moloch“ pilnųjų paketų gaudyklė

„Moloch“ metodas išsamiai aprašytas 2.1 skyriuje „Moloch“ metodo taikymas“, o 2.6 lentelėje pateikti jo incidentų valdymo tyrimo rezultatai.

2.6 lentelė. „Moloch“ incidentų valdymo galimybių tyrimas

Etapas / Tipas	Prevencija	Aptikimas	Blokavimas	Analizė
Kenkėjiškos programos	**Nėra.**	**Ribotas.** Galima įkelti laikmeną su požymiais, kurie bus aptinkami, tačiau aptikimo galimybės nėra išvystytos.	**Nėra.**	**Yra.** Kenkėjiškas programas galima nustatyti pagal siunčiamą turinį (angl. *payload*), taip pat pagal įtartinus sujungimus.
Paskirstytos sutrikdymo atakos				**Yra**. Tokio tipo atakos nesunkiai nustatomos pagal stipriai išaugusį tinklo srautą į

				vieną ar kelis prisijungimo tašką (-us).
Sukčiavimas				**Yra.** Laiškų ir siunčiamų laikmenų analizė gali tiksliai atkurti laiško ar laikmenos turinį.
Brukalų siuntimas				**Yra.** Laiškų ir siunčiamų laikmenų analizė gali tiksliai atkurti laiško ar laikmenos turinį. Taip pat galima nustatyti tokius laiškus siunčiančius serverius ar e. pašto dėžutes.
Infekuotų kompiuterių tinklai				**Yra.** Pagal įtartinus ir siunčiamus pranešimus tarp įrenginių sujungimus galima nustatyti valdantįjį serverį.
Programų pažeidžiamumų išnaudojimas				**Yra.** Galima pilnai atkurti siunčiamų pranešimų turinį jį tirti, nustatant bandymus išnaudoti programų pažeidžiamumus.
Vidinių sistemos vartotojų keliama grėsmė				**Yra.** Galima atkurti siunčiamų laiškų ir laikmenų turinį ar atlikti jų paiešką pagal maišos funkcijos reikšmę, taip nustatant, kokia informacija iš kur ir į kur buvo siunčiama.

2.3 „Moloch“ metodo taikymo ir tyrimo išvados

Palyginus „Moloch“ su kitomis tinklo saugos priemonėmis, galima teigti, kad tai galingas tinklo srauto analizės įrankis, leidžiantis surinkti ir apdoroti didelius paketų kiekius beveik realiu laiku. Didžiausi „Moloch“ privalumai, palyginus su kitomis sistemomis, yra lanksti konfigūracija, kuri leidžia efektyviai analizuoti nuo mažiausio srauto (<100 Mbps), naudojant paprasčiausią personalinį ar mini kompiuterį, iki dešimtis gigabitų per sekundę skaičiuojančius stuburinių tinklų srautus, naudojant specializuotą galingą įrangą. Integruota „ElesticSearch“ duomenų bazė realiu laiku sauganti ir indeksuojanti paketus, leidžia atlikti analizę greičiau ir lanksčiau nei kitomis tirtomis sistemomis, paketus saugančiomis tiesiai į diską ar reliacinę duomenų bazę.

Ištyrus tinklo saugos priemones pagal IS incidentų valdymo galimybes įvairiuose etapuose, aiškiai matosi, kad „pfSense“ ugniasienė yra orientuota į incidentų prevenciją, „Security Onion“ IAS rinkinys į incidentų aptikimą ir blokavimą, tačiau turi ir neblogas analizės galimybes, „nfsen“ tėkmių analizės įrankis skirtas paviršutiniškai, bet greitai analizei, o „Moloch“ orientuotas tik į išsamią, didelių kiekių srauto duomenų analizę. Nors savo analizės galimybėmis „Moloch“ lenkia visas kitas tirtas priemones, naudotis juo nėra paprasta ar patogu dėl automatinio incidentų aptikimo trūkumo, todėl norint efektyviai naudotis „Moloch“ kaip analizės įrankiu, reikia specifinių žinių. Dėl šios priežasties incidentų požymių aptikimo efektyvumas tiesiogiai priklauso nuo operatoriaus kompetencijos ir pastabumo, ir tai yra didžiausias „Moloch“ trūkumas, trukdantis ją panaudoti kaip tinklo saugos priemonę.

Siekiant ištaisyti nustatytą „Moloch“ automatizacijos trūkumą, sekančioje dalyje bus aprašoma „Moloch“ ir požymių dalijimosi platformos MISP integracija bei incidentų požymių aptikimo automatizavimas, remiantis viešomis incidentų požymių duomenų bazėmis.

3. „MOLOCH“ SISTEMOS TOBULINIMAS AUTOMATINIAM INCIDENTŲ APTIKIMUI

3.1 Kenkėjiškų programų informacijos dalijimosi platforma – MISP

Tinklo saugos priemonės incidentus aptinka pagal požymius ir tam tikras jų kombinacijas (pavyzdžiui, įtartina byla siunčiama iš neaiškaus elektroninio pašto adreso). Dažniausiai naudojami požymiai ir incidentų tipai, kuriuos jie padeda aptikti pateikiami 3.1 lentelėje.

3.1 lentelė. Incidentų požymių ir tipų ryšys

Požymis / **Incidento Tipas**	**Pradinis (angl. *Source*) IP adresas**	**Galutinis (angl. *Destination*) IP adresas**	**Vardas (angl. *Hostname*)**	**Nuoroda (URL)**	**E. pašto adresas**	**Turinio maišos funkcijos reikšmė (pvz. MD5)**
Kenkėjiškos programos	Sekant plitimą atvirkštine tvarka, galima nustatyti pirminį šaltinį	Dažnai kenkėjiškos programos valdomos iš serverio, kuris pasiekiamas per tam tikrą, iš anksto nurodytą, IP adresą, vardą ar nuorodą.			-	Kenkėjiškos programos platinamos kaip paleidžiamos laikmenos (pvz. *.exe* tipo).
Paskirstytos sutrikdymo atakos	-	Paprastai atakuojamas konkretūs taikiniai, juos pasiekiant pagal IP adresą ar vardą.		-	-	-
Sukčiavimas	-	-	-	Nurodomas nuoroda, kuri panaši į žinomos svetainės adresą (pvz. vietoj *bankas.lt/saskaita* nurodomas *bankas.io/saskaita*).	Gali būti bandoma impersonifikuoti kokios nors įmonės atstovą sukuriant panašų elektroninį pašto adresą (pvz. vietoj *vardas@ktu.edu* naudojamas *vardas@kttu.edu*).	-
Brukalų siuntimas	Pašto serveriai, siunčiantys brukalus, gali būti aptikti ir blokuojami	-	-	-	Paprastai iš vienos dėžutės siunčiami dideli kiekiai laiškų skirtingiems adresatams.	Žinomus brukalus galima atskirti pagal jų turinį.

	pagal IP adresą.					
Infekuotų kompiuterių tinklai	-	Valdymo serveris pasiekiamas per tam tikrą, iš anksto nurodytą, IP adresą, vardą ar nuorodą.			-	-
Programų pažeidžiamumų išnaudojimas	Pagal šaltinio IP adresą galima sekti veiksmus.	-	-	Bandoma vykdyti ataką per URL parametrus, pavyzdžiui, įterpti Sql užklausą.	-	Siunčiami tam tikro formato duomenys ar paketas.
Vidinių sistemos vartotojų keliama grėsmė	Nustatomas konkretus įrenginys.	-	Bandoma pasiekti tinklapius laikmenų įkėlimui išoriniame tinkle.		Bandoma siųsti konfidencialius dokumentus į pašto dežutę, esančią ne kompanijos serveryje.	Žinomų konfidencialių dokumentų judėjimą tinkle galima sekti pagal jų maišos reikšmę.

MISP (angl. *Malware Information Sharing Platform – MISP*) yra atviro kodo sprendimas kompiuterinių incidentų ir kenkėjiškų programų požymių rinkimui, saugojimui, katalogavimui ir dalijimuisi tarp skirtingų organizacijų bei nepriklausomų tyrėjų. MISP projektas vystomas grupės nepriklausomų programuotojų, Belgijos gynybos ministerijos, NATO kompiuterinių incidentų reagavimo centro (angl. *NATO Computer Incident Response Capability Technical Centre – NCIRC*) ir Liuksemburgo kompiuterinių incidentų reagavimo centro (angl. *Computer Incident Response Center Luxembourg – CIRCL*). Šiuo metu CIRCL daugiausiai ir kuruoja MISP projektą, rengia mokymus, organizuoja konferencijas, veikia kaip informacijos dalijimosi centras. Dažniausiai naudojami yra tokie požymiai:

- Atakuotojo IP adresas;
- Įtartino elektroninio pašto adresas;
- Siunčiamų duomenų kontrolinė suma (md5/sha1/sha256);
- Kompiuterio pavadinimas (angl. *hostname*);
- Domeno vardas (angl. *domain*).

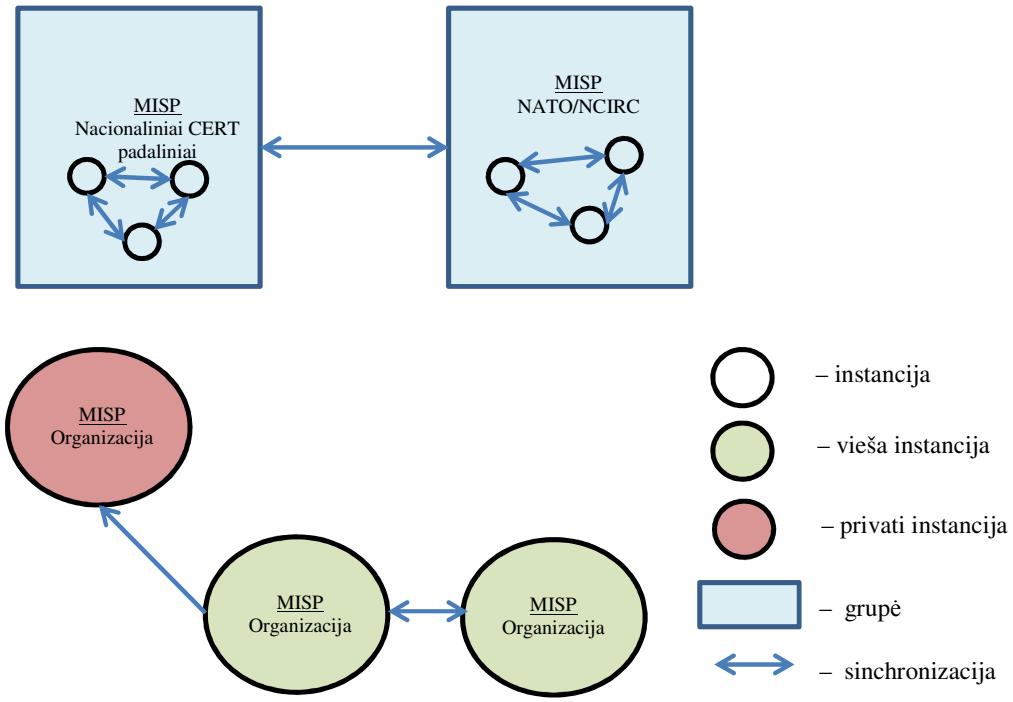
Pagrindinis MISP tikslas yra sukurti platformą, leidžiančią saugiai ir lengvai dalintis informacija tarp IT saugumo ekspertų, tyrėjų, organizacijų, valstybinių institucijų, taip leidžiant efektyviau ir greičiau reaguoti į kylančias grėsmes. Tradicinis pranešimas apie naujas grėsmes, kai tyrėjas, aptikęs ataką ir išanalizavęs požymius, apie tai praneša straipsniu, elektroniniu paštu ar kitomis, tam specialiai nepritaikytomis informacijos perdavimo priemonėmis, yra per lėtas norint efektyviai užkirsti kelią tolimesnei atakai ar kenkėjiškų programų plitimui. MISP automatinis incidentų požymių duomenų bazių sinchronizavimas leidžia informacijai apie grėsmes pasklisti žymiai greičiau, taip suteikiant daugiau laiko apsaugoti potencialiai pažeidžiamas sistemas.[26]

[26] https://www.circl.lu/services/misp-malware-information-sharing-platform/

3.2 MISP architektūra

MISP yra paskirstyta sistema, susidedanti iš atskirų instancijų. Instancijas galima sujungti į grupes (angl. *community*), pavyzdžiui, sujungiant organizacijas nacionaliniu lygmeniu, organizacijos skirtingus padalinius, universiteto tyrimo centrus ir t.t. Kiekviena instancija gali dalintis informacija su kitomis keliais būdais (žr. pav. 3.1):

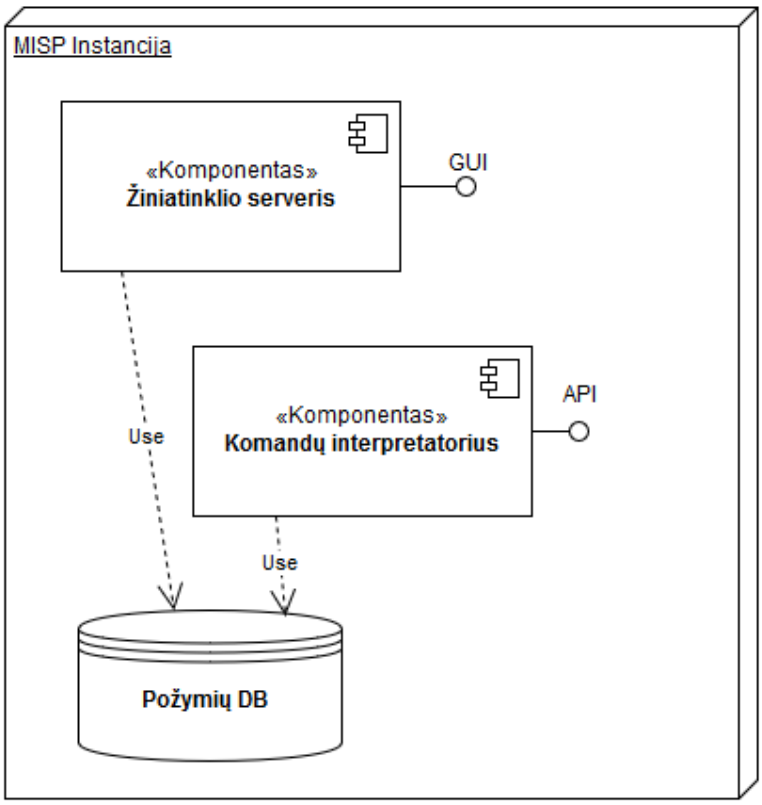
- Privačiai (instancijos viduje, pavyzdžiui, įmonėje);
- Grupėje;
- Sujungtose grupėse (tarpusavio bendradarbiavimo sutartį pasirašiusios grupės, pavyzdžiui, NATO/NCIRC ir nacionaliniai CERT centrai);
- Viešai (bet kas, prisijungęs prie instancijos, gali sinchronizuotis).



3.1 pav. MISP sistemos pavyzdys

Kiekviena instancija susideda iš 3 pagrindinių dalių (žr. pav. 3.2):

- Incidentų ir kenkėjiškų programų požymių duomenų bazės;
- Grafinės vartotojo sąsajos;
- Aplikacijų programavimo sąsajų (angl. *Application Programming Interface – API*);



3.2 pav. MISP instancijos architektūra

Šiame darbe toliau plačiau nenagrinėjamas MISP sistemos veikimas, laikoma, kad atnaujinta požymių duomenų bazė kažkokiu būdu yra prieinama. Svarbiausia dalis MISP integravimui su „Moloch“ yra abiejų sistemų aplikacijos programavimo sąsajos, kurios toliau ir bus plačiau nagrinėjamos.

3.3 Automatiniai incidentų požymių atnaujinimo servisai

„Moloch“ neturi integruotos incidentų požymių duomenų bazės, tačiau tiriamoje „Moloch“ versijoje (0.20.2) palaikomi keletas komercinių automatinių incidentų požymių atnaujinimo servisų[27] (angl. *feed*), prie kurių komercinę prieigą reikia įsigyti atskirai. Taip pat palaikomas nemokamas požymių importavimas per URL, laikmenas, „ElasticSearch“ ir „Redis“[28], tačiau tai tik sąsajos, skirtos požymių rinkiniams importuoti, ne savarankiškai veikiantys požymių atnaujinimo servisai. MISP ir „Moloch“ automatizuotų požymių atnaujinimo servisų palyginimas pateiktas 3.2 lentelėje, iš kurios matyti, kad „Moloch“ turi tik trivialias, mokamas automatinio incidentų požymių atnaujinimo galimybes. MISP yra specialus incidentų požymių atnaujinimui, aprašymui ir dalijimuisi skirtas įrankis, todėl nenuostabu, kad šioje srityje jis lenkia „Moloch“ pagal visus parametrus ir galimybes.

[27] https://github.com/aol/moloch/wiki/WISE#commercial-sources
[28] https://github.com/aol/moloch/wiki/WISE#free-sources

3.2 lentelė. Automatinių incidentų požymių atnaujinimo servisų palyginimas

Parametras	„Moloch“	MISP
Integruotų požymių atnaujinimo servisų skaičius (iš jų nemokamų)	6 (0)	45 (45)
Naujų servisų integravimo galimybė	Kiekvienas servisas programiškai aprašomas atskirai; reikia perkrauti servisus	Užtenka nurodyti palaikomo serviso URL ar laikmeną ir pasirinkti parametrus
Palaikomi formatai	Naudojamas savitas formatas kiekvienam servisui atskirai	MISP, CSV, laisvo teksto
Incidentų požymių atnaujinimo galimybė	Konfigūruojama automatinė	Konfigūruojama automatinė, rankinė
Sukauptų incidentų požymių peržiūra per grafinę vartotojo sąsają	Negalima	Galima
Sukauptų incidentų požymių modifikavimas per grafinę vartotojo sąsają	Negalimas	Galimas
Naujų incidentų požymių aprašymas per grafinę vartotojo sąsają	Negalimas	Galimas
Incidentų požymių dalijimosi galimybė per grafinę vartotojo sąsają	Negalima	Galima tiek privačiai, tiek viešai

3.4 MISP integracija su „Moloch“

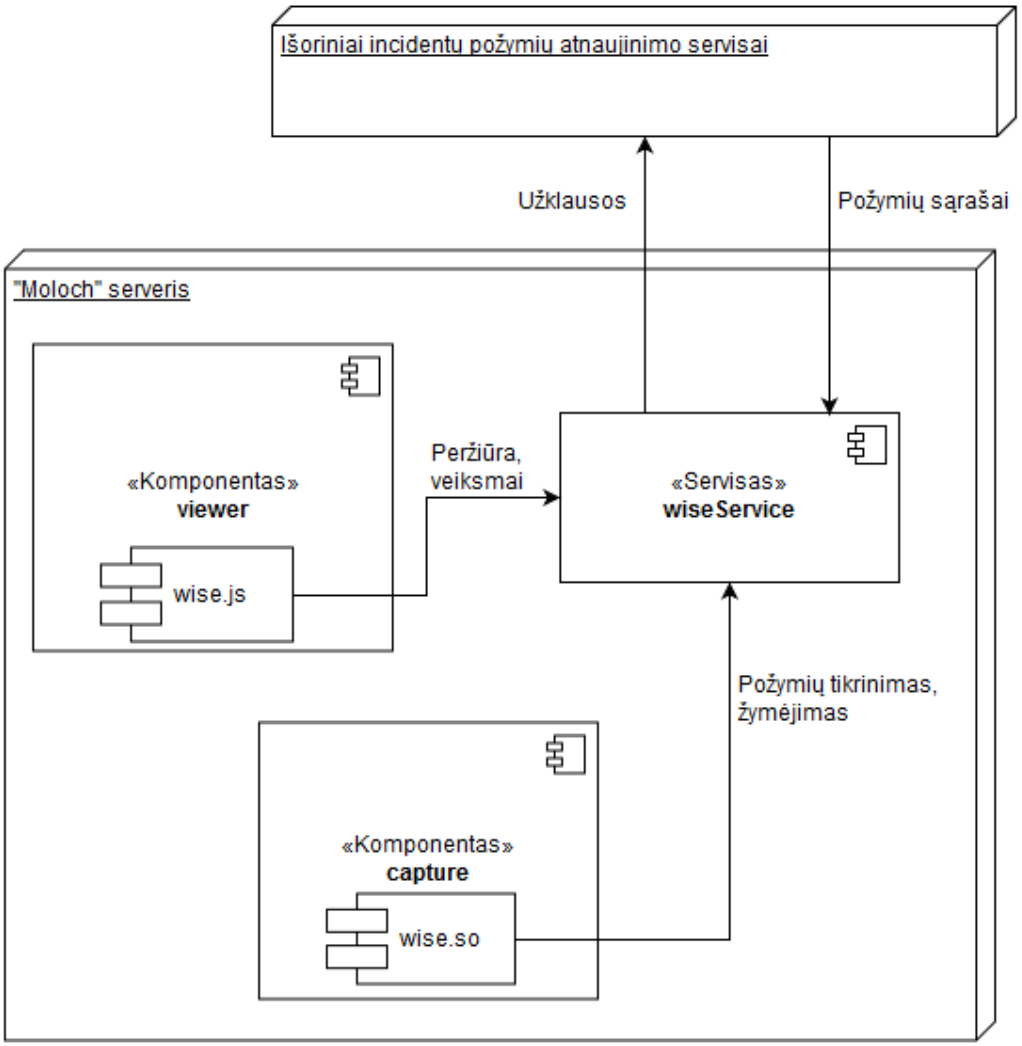
„Moloch“ WISE modulis yra „JavaScript“ programavimo kalba, „Node.js“ aplinkoje parašytas įrankis, skirtas automatiniam incidentų požymių atnaujinimui, srauto žymėjimui (angl. *tag*) ir pažymėto srauto atvaizdavimui skirtas įrankis. Pažymėti paketai grafinėje vartotojo sąsajoje gali būti išskirti (vizualus paryškinimas, tekstinio pranešimo spausdinimas operatoriui ir t.t.), taip atkreipiant dėmesį į potencialius incidentus. Papildomai galima suprogramuoti veiksmus, kurie atsiranda meniu spragtelėjus pele ant pažymėto lauko. Šiuo metu palaikomas specialios nuorodos parodymas, tačiau „Moloch“ kūrėjai žada, kad ateityje galima tikėtis daugiau veiksmų[29].

WISE susideda iš 3 pagrindinių dalių (žr. pav. 3.3):

- **wiseService**: savarankiškas „Node.js“ servisas, kuris atsakingas už automatinį incidentų požymių atnaujinimą iš nurodytų ir sukonfigūruotų šaltinių, jų įrašymą į atmintį ar laikmenas bei konkrečių duomenų atitikimą su požymiais tikrinimą (incidentų aptikimas). Kaip ir daugelis „Moloch“ sistemos dalių, **wiseService** taip pat yra lengvai plečiamas ir gali būti iškeliamas į atskirą fizinį ar virtualų serverį;
- **wise.so**: srauto įrašinėjo serviso (**capture**) įskiepis (angl. *plugin*) – „C“ programavimo kalba parašyta statinė biblioteka, kuri nuolat tikrina sukaupto srauto duomenų elementus (paketo antraštę, adresus, siunčiamus duomenis ir t.t.) per **wiseService** suteiktą sąsają. Aptikus incidentą, duomenų elementui priskiriama atitinkama žyma;
- **wise.js**: grafinės vartotojo sąsajos (**viewer**) įskiepis, skirtas pažymėtų srauto elementų atvaizdavimui ir papildomų veiksmų funkcionalumui suteikti.

[29] https://github.com/aol/moloch/wiki/Settings#rightclick

Dėl potencialiai didelių užklausų skaičiaus ir duomenų srautų, **wiseService** ir **wise.so** turi integruotus, kelių sluoksnių podėlius (anl. *cache*). Esant itin dideliems srautams ir nepakankant integruotų podėlių galimybių, juos galima konfigūruoti ar pakeisti galingesniais, specialiai tam pritaikytais įrankiais (pvz., „Redis“[30]).



3.3 pav. WISE modulių tarpusavio ryšių diagrama

Naujo automatinio incidentų požymių atnaujinimo serviso integravimui į „Moloch“, WISE modulyje reikia atlikti tokius veiksmus:

1. Sukurti naują laikmeną pavadinimu ***source.<ServisoPavadinimas>.js*** ir patalpinti ***moloch/wiseService*** aplanke. Sukurtoje laikmenoje „JavaScript“ kalba aprašyti serviso klasę, kuri suteikia programinę prieigą **wise.js** ir **wise.so** moduliams, parsiunčia ir išsaugo incidentų požymių sąrašus ar tikrina juos išoriniame incidentų požymių atnaujinimo servise;
2. Konfigūracijos laikmenoje (**wise.ini**) parametrui **wiseHost** priskirti nuorodą (URL), kuria pasiekiamas **wiseService**;
3. Specialiai **wiseService** skirtoje (**wise.ini**) konfigūracinėje laikmenoje (žr. pav. 3.5) aprašyti naują servisą, jo adresą ir naudojamus parametrus (pvz., prisijungimo raktą);

[30] https://redis.io/

4. Bendroje konfigūracinėje laikmenoje (**config.ini**) nurodyti WISE serviso parametrus (žr. pav. 3.4);
5. ***capture*** serviso konfigūracinėje laikmenoje (bendroje **config.ini** ar individualioje, jei naudojama specifinė konfigūracija) nurodyti **wise.so** modulį kaip naudojamą įskiepį (**plugins=wise.so**);
6. Kad naujai įdiegtas servisas pradėtų veikti, reikia perkrauti „Moloch“.

```
## Start wiseService configuration
# Host to connect to for wiseService
wiseHost=127.0.0.1

# Number of seconds to cache results before asking wiseService again
wiseCacheSecs=600

# Max number of items to store in the wise cache that is local to each moloch-capture node
wiseMaxCache=100000

# Number of connections to wiseService, this is also the number of concurrent wise queries.
wiseMaxConns=10

# Number of oustanding requests to the wiseService
wiseMaxRequests=100
## End wiseService configuration

# Semicolon ';' seperated list of plugins to load and the order to load in
# plugins=tagger.so; netflow.so
plugins=wise.so

# Plugins to load as root, usually just readers
#rootPlugins=reader-pfring; reader-daq.so

# Semicolon ';' seperated list of viewer plugins to load and the order to load in
viewerPlugins=wise.js
```

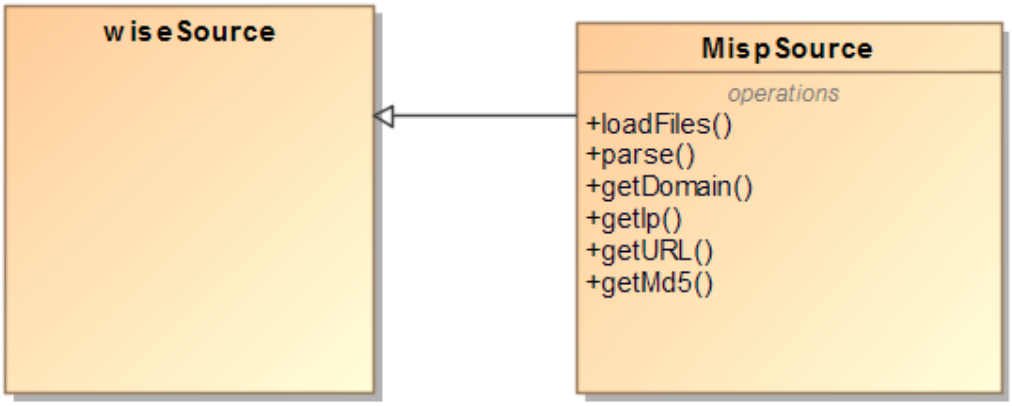
3.4 pav. Bendros konfigūracinės laikmenos (**config.ini**) ištrauka

```
# WISE Service Config - https://github.com/aol/moloch/wiki/WISE
[wiseService]
port = 8081

[misp]
key=1LsqdHC0fmMrG6wokpFgocdU6Dvlk08koxUa1KoL
host=192.168.1.17
```

3.5 pav. WISE serviso konfigūracinės laikmenos (wise.ini) ištrauka

MISP integravimas į „Moloch“ atliekamas pagal aukščiau pateiktą scenarijų. Pirmiausia suprogramuojamas naujas integracijos WISE modulis (***source.misp.js***), kuris komunikuodamas su MISP per REST (angl. *Representational state transfer*) sąsają parsiunčia požymių sąrašus CSV (angl. Comma-Separated Values) ir suteikia metodus požymių tikrinimui. Pilnas integracijos programos tekstas, parašytas šiam darbui, pateiktas A priede. Naujajame modulyje aprašoma nauja klasė **MispSource**, kurios prototipas yra **wiseSource** klasė (žr. pav. 3.6).

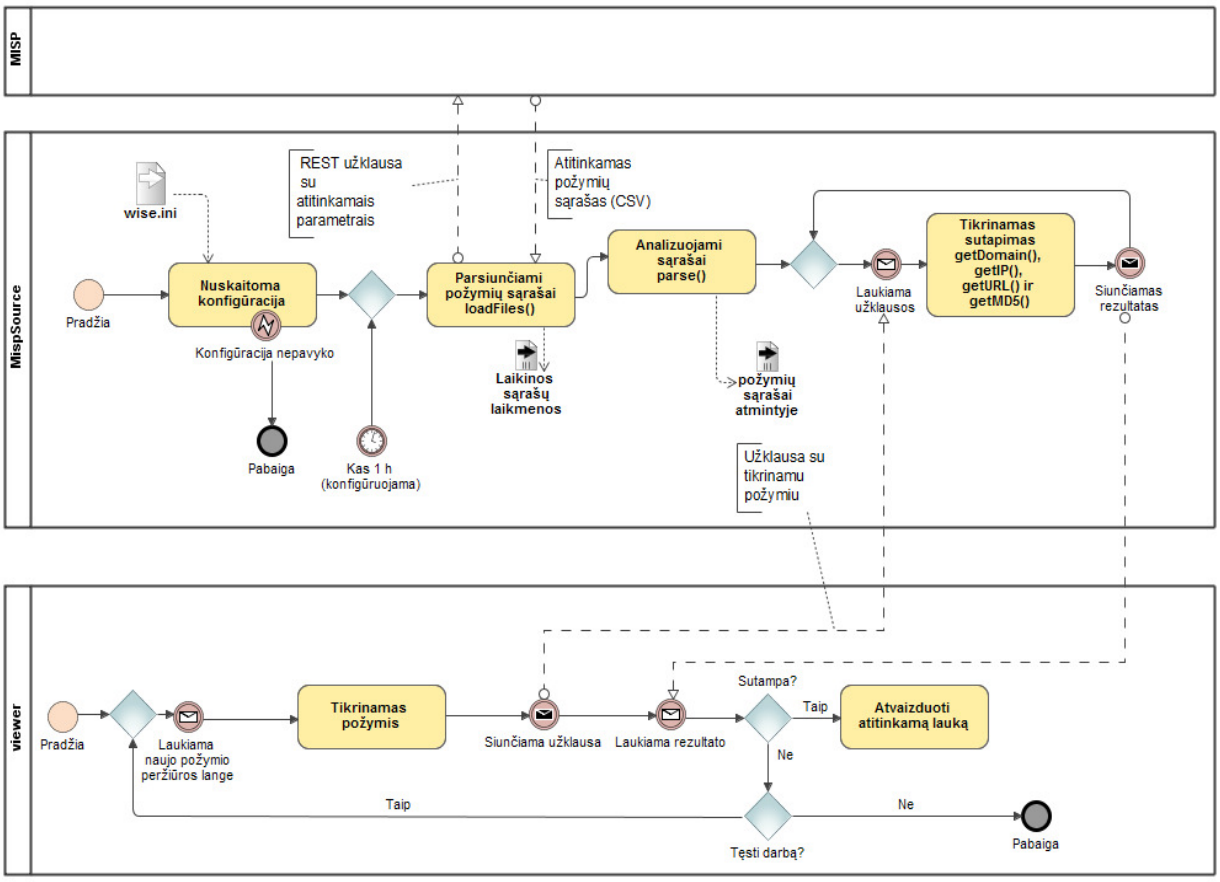


3.6 pav. Integracijos modulyje aprašyta klasė ir pagrindiniai jos metodai

MispSource klasėje aprašomi tokie pagrindiniai metodai:

- **loadFiles** metodas, skirtas požymių sąrašų (CSV formatu) parsiuntimui iš MISP per REST sąsają ir jų išsaugojimui diske. Šis metodas iškviečiamas tam tikrais, sukonfigūruotais laiko intervalais, kad iš naujo parsiųstų požymių sąrašus;
- **parse** metodas analizuoja parsiųstus CSV sąrašus, sukelia požymius į atmintyje saugomus sąrašus (angl. *Hash Table*) pagal tipą ir kiekvienam sąrašui priskiria atitinkamus WISE laukus;
- **getDomain, getIP, getURL, getMD5** metodai atitinkamai skirti domeno vardo, IP adreso, nuorodos ir maišos funkcijos reikšmių tikrinimui. Šiuos metodus kviečia **viewer** modulis, kiekvieną kartą, kai į peržiūros langą patenka naujas požymis iš išvardintų. Jei tikrinamas požymis sutampa su esančiu atitinkame sąraše, metodas grąžina atitikimo reikšmę, ir požymio laukas pažymimas (angl. *tag*) peržiūros lange (žr. pav. 3.10). Sutampantys požymiai galimai indikuoja incidentą.

Integracijos modulio veikimas ir jo sąveika su MISP bei **viewer** elementais pavaizduotas 3.7 paveiksle.



3.7 pav. Integracijos modulio procesų diagrama

Nors MISP REST sąsaja aprašo daug komandų, tačiau šiame darbe naudojamos tik keletas jų, skirtų incidentų požymių sąrašų parsisiuntimui. MISP taip pat palaiko daugiau kategorijų ir incidentų požymių tipų, todėl bus naudojami tik tie, kurie turi atitikmenį „Moloch“ sistemoje. Incidentų požymių tipai „Moloch“ ir MISP sistemose, atitinkamos REST komandos jiems gauti ir integracijos modulio metodai požymių tikrinimui pateikti 3.3 lentelėje.

3.3 lentelė. Incidenų požymių tipai „Moloch“ ir MISP sistemose

Tipas	**Požymis „Moloch“ sistemoje**	**Atitikmuo MISP sistemoje**	**REST GET komandos struktūra šio tipo požymių sąrašui parsisiųsti**	**Požymio atitikmens tikrinimo metodo pavadinimas *source.misp.js* modulyje**

IP adresas	ip	ip-src/ip-dst	`GET { header: "Authorization: <aut_kodas>", url: „http://<misp_adr>/events/csv/download/false/false/false/ip-src“ }`
			
			`GET { header: "Authorization:`
				`getIp`

			<aut_kodas>", url: „http://<misp_adr>/events/csv/download/false/false/false/false/ip-dst" }	

E. pašto adresas	email	email-src, email-dst	GET {
			header: "Authorization: <aut_kodas>",
			url:
			„http://<misp_adr>/events/csv/download/false/false/false/false/email-src"
			}
			
			GET {
			header: "Authorization: <aut_kodas>",
			url:
			„http://<misp_adr>/events/csv/download/false/false/false/false/email-dst"
			}	

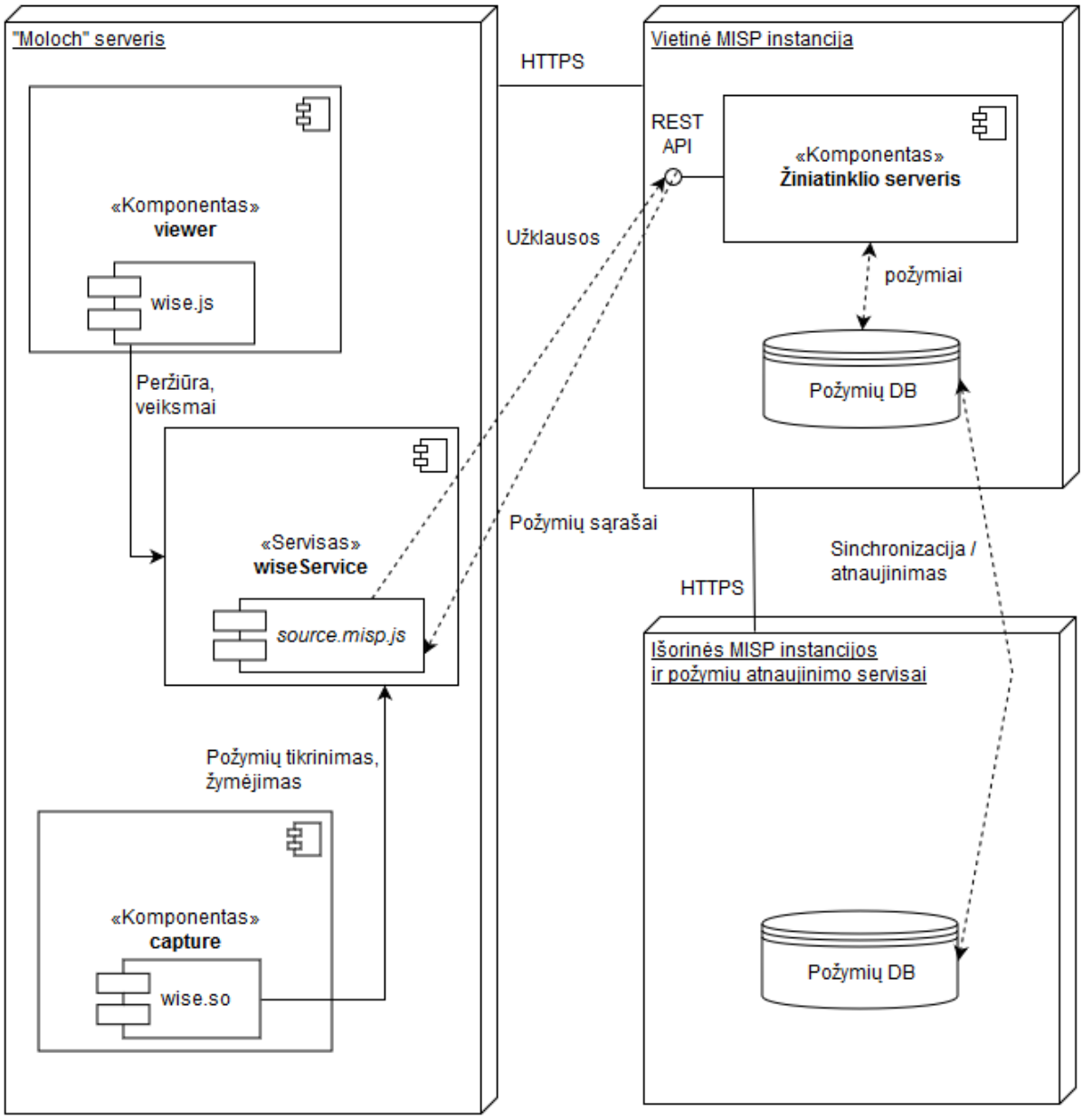
Siunčiamų duomenų (angl. *payload*) maišos funkcija	md5	md5	GET {
			header: "Authorization: <aut_kodas>",
			url:
			„http://<misp_adr>/events/csv/download/false/false/false/false/md5"
			}
				getMd5

Domeno pavadimimas	domain	domain	GET {
			header: "Authorization: <aut_kodas>",
			url:
			„http://<misp_adr>/events/csv/download/false/false/false/false/domain"
			}
				getDomain

Universalusis adresas	url	url	GET {
			header: "Authorization: <aut_kodas>",
			url:
			„http://<misp_adr>/events/csv/download/false/false/false/false/url"
			}
				getURL

Čia **<aut_kodas>** yra unikalus raidžių ir skaičių rinkinys, atliekantis rakto funkciją autorizuotam prisijungimui prie MISP RESP sąsajos. Nepateikus teisingo autorizacijos kodo, prieiga blokuojama. **<misp_adr>** yra adresas (IP ar domenas), kuriuo pasiekiamas MISP serveris. Pagal nutylėjimą MISP į siunčiamos incidentų požymių sąrašus įtraukia tik tuos įrašus, kurie yra suprantami mašinoms, todėl papildomai apdoroti jų nereikia.

Integravus „Moloch“ ir MISP sistemas, jos veikia lygiagrečiai viena kitai. Tokio derinio architektūra lanksti, abi sistemos gali veikti tiek viename serveryje, tiek nutolusiuose serveriuose komunikuojant tinklu, per saugią HTTPS sąsają. Supaprastinta integruotos „Moloch“ ir MISP sistemos struktūrinė diagrama pavaizduota 3.8 paveiksle.



3.8 pav. „Moloch“ ir MISP integracijos struktūrinė diagrama

„Moloch“ neturi priemonių pranešti apie aptiką incidentą, todėl tokiam funkcionalumui įdiegti pasitelkiama nemokama, atviro kodo programa „Tailon“[31], kuri veikia kaip žiniatinklio serveris norimam įvykių žurnalui atvaizduoti grafiškai nutolusiam operatoriui.

```
#!/bin/bash

grep MATCH ../logs/wise.log >> ~/alert.log &
tailon -b 0.0.0.0:8080 -f ~/alert.log
```

3.9 pav. Įspėjimo ir atvaizdavimo apie aptiktus incidentų požymius rašmena

[31] https://tailon.readthedocs.io/en/latest/

Sukurtas integracijos modulis, aptikęs požymį, į **wise.log** įvykių žurnalą parašo raktažodį „MATCH!", po kurio toje pačioje eilutėje išvedama trumpa aptikto požymio informacija. Pranešimo išvedimui į žiniatinklį, sukuriama „Bash" rašmena (angl. *script*), kuri nuolat tikrina įvykių žurnalą ir laukia, kol bus aptiktas raktažodis (žr. pav. 3.9).

3.5 „Moloch" ir MISP integracijos tyrimas ir testavimas

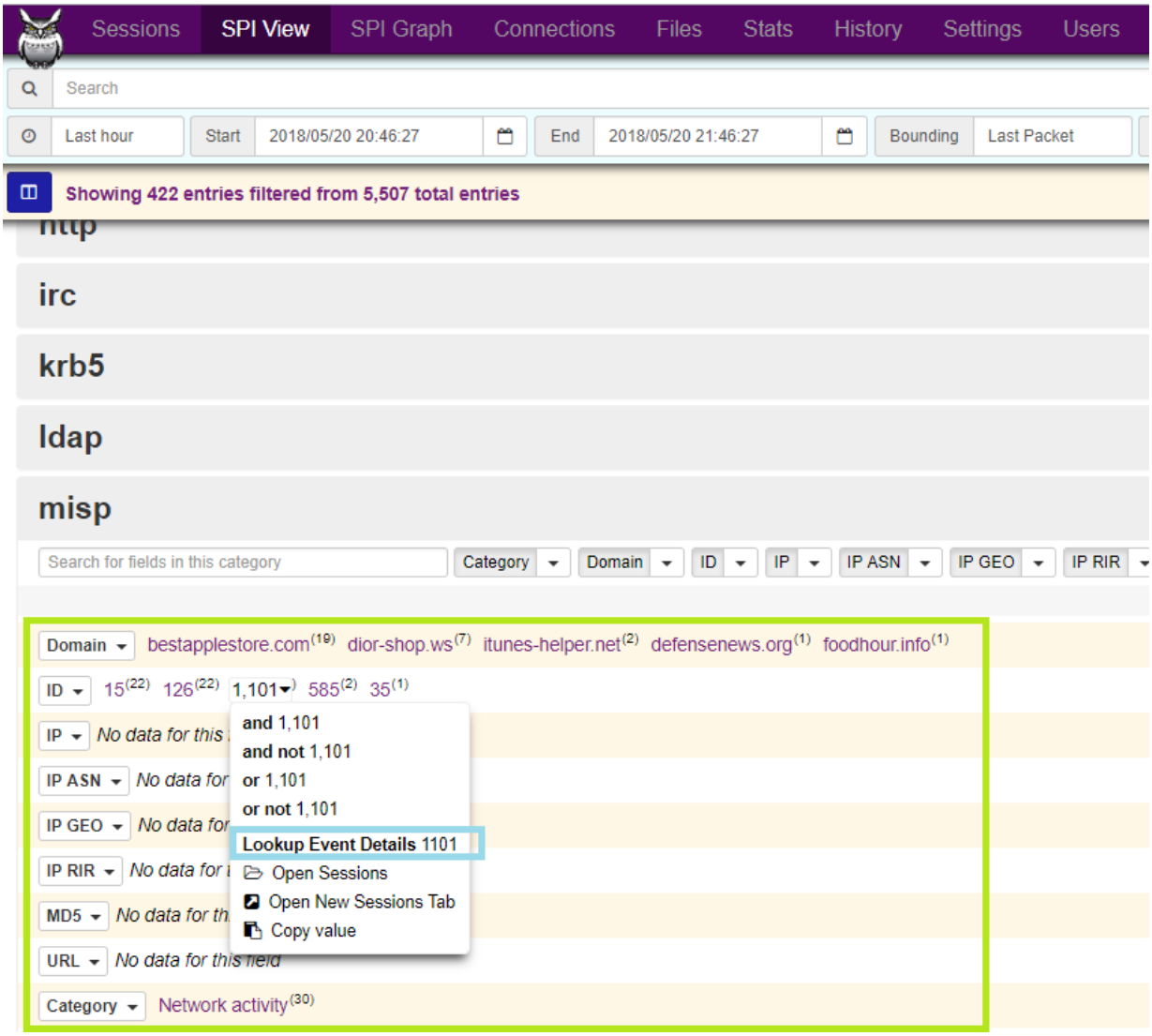
Atliktos integracijos rezultatas yra nauja sistema, kuri paveldi galingas „Moloch" analizės priemones ir gauna naują automatinį incidentų aptikimo, pagal atsinaujinančius incidentų požymius, funkcionalumą. Palyginus su paprastu „Moloch", sukurta integracija padengia IS incidentų valdymo automatinio aptikimo etapą bei žymiai supaprastina analizės galimybes, kadangi incidentai pažymimi automatiškai, ir nereikia jų kiekvieno ieškoti atskirai. Integruotosios sistemos galimybės pateiktos 3.4 lentelėje.

3.4 lentelė. Integruotos „Moloch" ir MISP sistemos galimybių tyrimas

<table>
<tr><th>Etapas / Tipas</th><th>Prevencija</th><th>Aptikimas</th><th>Blokavimas</th><th>Analizė</th></tr>
<tr><td>Kenkėjiškos programos</td><td rowspan="4">Nėra.</td><td rowspan="4">Yra. Integracija su MISP leidžia aptikti incidentus pagal automatiškai atnaujinamus požymių sąrašus.</td><td rowspan="4">Nėra. Aptikus incidentą, galima generuoti įvykį, todėl yra galimybė atlikti integraciją su SIEM, NSM ar IAS sistemomis incidentų blokavimui ir eskalavimui.</td><td>Yra. Kenkėjiškas programas galima nustatyti pagal siunčiamą turinį (angl. payload), taip pat pagal įtartinus sujungimus.</td></tr>
<tr><td>Paskirstytos sutrigdymo atakos</td><td>Yra. Tokio tipo atakos nesunkiai nustatomos pagal stipriai išaugusį tinklo srautą į vieną ar kelis prisijungimo tašką (-us).</td></tr>
<tr><td>Sukčiavimas</td><td>Yra. Laiškų ir siunčiamų laikmenų analizė gali tiksliai atkurti laiško ar laikmenos turinį</td></tr>
<tr><td>Brukalų siuntimas</td><td>Yra. Laiškų ir siunčiamų laikmenų analizė gali tiksliai atkurti laiško ar laikmenos turinį. Taip pat galima nustatyti tokius laiškus siunčiančius serverius ar e.</td></tr>
</table>

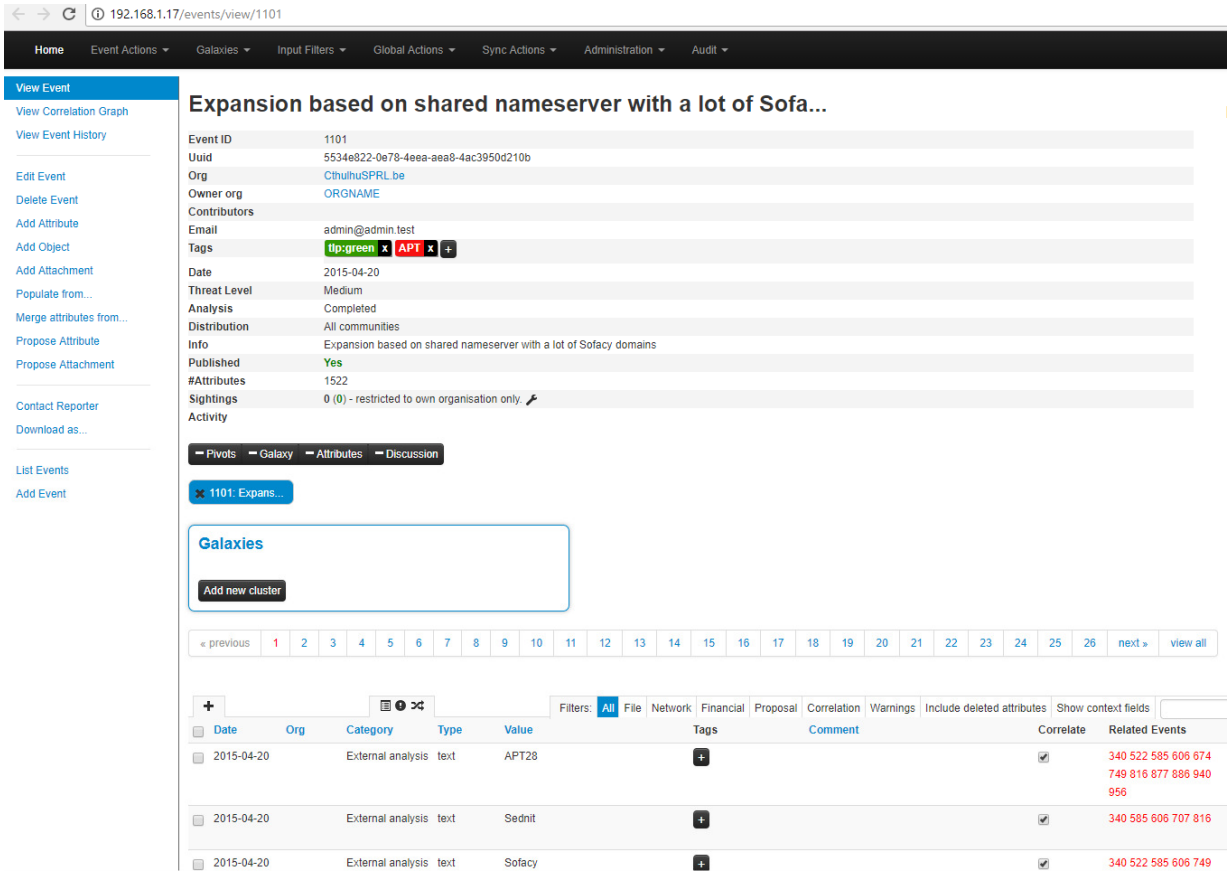
				pašto dėžutes.
Infekuotų kompiuterių tinklai				**Yra.** Pagal įtartinus ir siunčiamus pranešimus tarp įrenginių sujungimus galima nustatyti valdantįjį serverį.
Programų pažeidžiamumų išnaudojimas				**Yra.** Galima pilnai atkurti siunčiamų pranešimų turinį jį tirti, nustatant bandymus išnaudoti programų pažeidžiamumus.
Vidinių sistemos vartotojų keliama grėsmė		**Ribota.** Galima sukurti integraciją su konfidencialią informaciją atitinkančiais požymiais, tačiau tokia galimybė šitame darbe nenagrinėta.		**Yra.** Galima atkurti siunčiamų laiškų ir laikmenų turinį ar atlikti jų paiešką pagal maišos funkcijos reikšmę, taip nustatant, kokia informacija iš kur ir į kur buvo siunčiama.

Iš praktinio taikymo pusės, integracija leidžia lengvai aptikti įtartinus įvykius, kadangi jie automatiškai pažymimi „Moloch“ grafinės sąsajos lange „SPI View“ (žr. pav. 3.10, požymiai apvesti žaliu stačiakampiu). Aptikti įtartini požymiai grupuojami pagal atitinkamą incidento identifikacinį numerį MISP sistemoje (ID). Paspaudus ant atitinkamo incidento numerio, iššoka meniu su naujai sukurta komanda (žr. pav. 3.10, meniu punktas pažymėtas mėlynu stačiakampiu) ant kurios paspaudus, atidaromas naujas naršyklės langas, kuriame nukreipiama į detalų incidento aprašą MISP serveryje (žr. pav. 3.11).



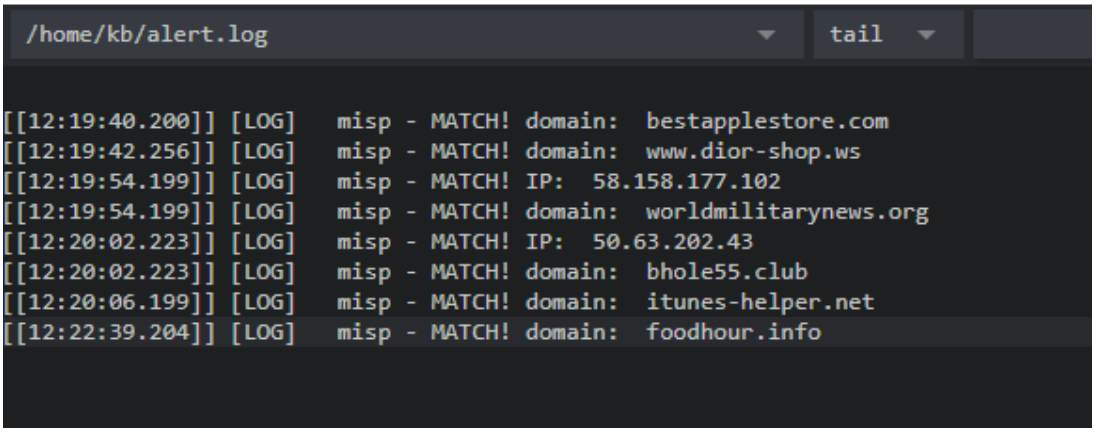
3.10 pav. Matomi aptikti incidentų požymiai „Moloch“ sistemoje po integracijos su MISP

Tie patys požymių laukeliai veikia kaip filtrai, kuriuos paspaudus, atrenkamas tik tą požymį atitinkantis srautas. Pasirinkus kelis požymius, galima atlikti detalų tyrimą, pavyzdžiui, pasirinktas tam tikras išeigos (angl. *source*) IP adresas ir maišos funkcijos reikšmė (MD5) parodys, kur iš šio IP adreso buvo siunčiamas konkretus turinys (kenksminga programa, konfidencialus dokumentas ir t.t). Toks funkcionalumas supaprastina visų tipų incidentų tyrimą, kadangi nereikia atrinkinėti ir ranka suvedinėti užklausų, taip pat sumažėja klaidos tikimybė, kadangi visi požymiai aptinkami automatiškai, ir nelieka žmogiškosios klaidos faktoriaus.



3.11 pav. Detalus incidento aprašymas MISP serveryje

Aptikus incidento požymį, perspėjimo rašmena nukopijuoja atitinkamą įvykio žurnalo eilutę į „Tailon“ programos stebimą laikmeną, kuri automatiškai atvaizduojama žiniatinklyje (žr. pav.3.12). Tai primityvi įspėjimo apie incidentus realizacija, sukurta siekiant parodyti, kad naujojo modulio galimybes nesunkiai galima išplėsti ir jį integruoti į SIEM, NSM ar kitas incidentų stebėjimo bei perspėjimo sistemas.



3.12 pav. Incidentų požymių aptikimo įspėjimo pranešimai žiniatinklyje

3.6 „Moloch“ ir MISP integracijos taikymo ir plėtimo galimybės

Naujai sukurta integruota „Moloch“ ir MISP sistema padengia IS incidentų valdymo aptikimo ir analizės etapus, taip pat turi galimybę generuoti incidentų aptikimo pranešimus. Atsižvelgus į aprašytas galimybes, galimi tokie integruotos sistemos taikymo scenarijai:

1) Integruotos sistemos naudojimas **lygiagrečiai kitoms tinklo saugos priemonėms**. MISP dalis leidžia dalintis incidentų požymiais tarp organizacijų realiu laiku, todėl potencialiai sukurta integracija leidžia aptikti tokius incidentus, kuriuos IAS dar nesugeba aptikti, pavyzdžiui, nauja greitai plintant kenkėjiška programa. Gavus informaciją per MISP apie tokią naują grėsmę, ir aptikus jos požymius tinkle, galima ją sekti, izoliuoti ir efektyviai blokuoti realiu laiku, atitinkamai sukonfigūravus ugniasienes ar pasitelkus kitas tinklo saugos priemones. Toks scenarijus reikalautų nemažai resursų, kadangi reikėtų įdiegti „Moloch“ sensorius tinklo mazguose, tačiau tai gali pasiteisinti didelės svarbos, kritinės infrastruktūros tinkluose;
2) Integruotos sistemos naudojimas **vietoj aktyvių incidentų aptikimo priemonių**. Integruota sistema tam tikrais atvejais galėtų iš dalies pakeisti IAS. Nors toks scenarijus neleistų automatiškai blokuoti incidentų, tai gali būti naudinga, kai nėra labai svarbu sustabdyti kiekvieną individualų incidentą, tačiau siekiama neleisti jiems nekontroliuojamai plisti. Tokio scenarijaus taikymo pavyzdys galėtų būti vidinis universiteto tinklas studentų bendrabučiuose, kuris nėra kritiškai svarbus, jame nesaugoma konfidenciali informacija, tačiau vis tiek norima turėti galimybę laiku sustabdyti kenkėjiškų programų plitimą (pavyzdžiui, gavus pranešimą apie incidento aptikimą, atitinkamai sukonfigūruojant ugniasienę ar atjungiant konkrečius įrenginius nuo tinklo), perspėti infekuotų įrenginių savininkus;
3) Integruotos sistemos naudojimas tik kaip **analizės įrankio**. Tokiu atveju nediegiama pilna „Moloch“ infrastruktūra ir sensoriai, o srauto informacija gaunama importuojant PCAP laikmenas, išsaugotas kitų tinklo įrenginių. Tai nebūtų incidentų aptikimas ir tyrimas realiu laiku, tačiau gali būti naudinga atliekant išsamų incidentų tyrimą post factum, siekiant atkurti įvykių eigą, nustatyti nutenkintos informacijos mastus, atliekant kriminalinį tyrimą ir pan. Iš visų pateiktų scenarijų šis išsiskiria savo mažais resursų reikalavimais, kadangi atlikti gigabaitų eilės srauto duomenų tyrimą užtektų vienos virtualios mašinos su minimaliais integruotos sistemos reikalavimais ir pakankamai vietos diske.

Sukurta „Moloch“ ir MISP integracija nėra galutinis produktas, tai labiau demonstracinė versija (angl. *Proof of Concept*), skirta pademonstruoti abiejų sistemų sinergiją ir plačias taikymo galimybes. Norint gauti galutinį produktą, reikėtų atlikti bent šiuos patobulinimus:

- **optimizuoti**, nes dabar nustatytu laiko tarpu visi požymiai siunčiami iš naujo, net jei pasikeitimų jų sąrašuose nebuvo;
- **suprogramuoti algoritmą**, kuris tikrintų, ar yra naujų požymių MISP sistemoje, ir juos aptikęs, inicijuotų sąrašų atnaujinimą;
- **patobulinti perspėjimo modulį**, kadangi dabar jis tiesiog išveda įspėjimą į žiniatinklio puslapį.

3.7 „Moloch“ ir MISP integracijos išvados

„Moloch“ yra galinga paketų gaudymo ir įrašymo sistema, suteikianti ribotas srauto analizavimo galimybes. Labiausiai analizę apsunkina automatizuotų įrankių nebuvimas, kas ypač pasijaučia analizuojant didesnius duomenų srautus. Šie „Moloch“ trūkumai kompensuojami integruojant MISP incidentų požymių valdymo platformą, skirtą incidentų ir kenkėjiškų programų

požymių rinkimui, saugojimui, katalogavimui ir dalijimuisi tarp skirtingų organizacijų bei nepriklausomų tyrėjų.

Sukurtoje integruotoje sistemoje MISP atlieka incidentų požymių rinkimo ir valdymo funkciją, o „Moloch“, naudojant WISE modulį, parsisiunčia paruoštus incidentų sąrašus ir ieško atitikmenų renkamame sraute. Tokia sistema leidžia automatizuoti incidentų aptikimą, juos paryškinant grafinėje vartotojo sąsajoje. Paspaudus dešinį pelės klavišą ant pažymėto laukelio, gaunama nuoroda į išsamesnį incidento aprašymą MISP sistemoje. Požymių laukai grafinėje sąsajoje veikia kaip filtrai, todėl pažymėjus keletą jų, galima sukurti įvairias kombinacijas išsamiam incidentų tyrimui.

„Moloch“ incidentų aptikimo automatizavimas padengia IS incidentų valdymo aptikimo etapą, kas šį įrankį daro patrauklų įtraukti į incidentų valdymo procesą greta prevencijos ir blokavimo priemonių. Automatinis incidentų aptikimas ir tyrimas leidžia žymiai sumažinti apkrovą operatoriui ir tyrėjui, kadangi nereikia kiekvieno incidento požymių ieškoti ranka įvedant užklausas (filtrus), kas taip pat sumažina klaidos tikimybę.

MISP sistemoje galima keistis incidentų požymiais tarp organizacijų realiu laiku, kas teoriškai leidžia aptikti ir neutralizuoti naujo tipo kenkėjiškas programas, vos tik joms pasirodžius. Toks kenkėjiškų programų plitimo stabdymas pradiniame etape leistų žymiai sumažinti jų padaromą žalą, kadangi paprastai kenkėjiškos programos plitimas vyksta geometrinės progresijos principu, atitinkamai augant ir daromai žalai.

IŠVADOS

1. Šiame darbe, apžvelgus šiuolaikines incidentų valdymo metodikas ir tinklo saugos priemones, nustatyta, kad dauguma priemonių orientuotos į prevencijos, aptikimo ir neutralizavimo etapus, tačiau ne mažiau svarbus analizės ir tyrimo etapas lieka apleistas bei jaučiamas tam skirtų priemonių trūkumas.
2. Nustatyta, kad pilnų paketų gaudymo, įrašymo, indeksavimo, saugojimo ir srauto analizės sistema „Moloch“ turi geriausias incidentų tyrimo galimybes tarp lygintų įvairių klasių tinklo saugos priemonių. „Moloch“ taip pat išsiskiria pilnųjų paketų gaudyklių rinkoje kaip vienas moderniausių ir lanksčiausių sprendimų.
3. Nepaisant to, kad su „Moloch“ galima atlikti išsamią incidentų analizę, sistemoje nėra integruotų automatinių incidentų aptikimo ar kitų automatizuotų priemonių, todėl toks darbas reikalauja gerų tyrėjo žinių. Net tyrimą atliekant aukščiausios klasės specialistui, vis tiek išlieka žmogiškosios klaidos galimybė, taip pat ribojamas tiriamo srauto kiekis, kadangi kiekvieno galimo incidento požymių tenka ieškoti atskirai.
4. Nustačius automatizacijos trūkumo „Moloch“ sistemoje faktą, atliekama jos integracija su incidentų požymių ir tyrimo informacijos dalijimosi platforma MISP, šiam trūkumui pašalinti. Integracija realizuojama suprogramavus „Moloch“ posistemės WISE modulį (pilnas programos kodas pateiktas priede A), kuris automatiškai atsisiunčia ir atnaujina incidentų požymių sąrašus iš MISP bei ieško požymių atitikimo sraute. Aptikus požymio atitikimą srauto elementui, šis pažymimas ir atvaizduojamas grafinėje vartotojo sąsajoje, atskirame laukelyje, kuris tuo pačiu atlieka ir filtro funkciją. Toks automatinis žymėjimas ir atvaizdavimas palengvina tyrėjo darbą, kadangi nereikia atitikimų ieškoti rankiniu būdu. Ištyrus integruotą sistemą, stebimas ženklus incidentų aptikimo tikslumo ir tyrimo produktyvumo padidėjimas.
5. Integruota sistema gali būti lengvai plečiama, papildant ją naujomis galimybėmis. Kaip to pavyzdys sukurta rašmena, kuri laukia, kol bus aptiktas požymis tarp srauto elementų, ir jo sulaukus, išveda pranešimą su aptikto požymio informacija žiniatinklyje. Taip atliekama įspėjimo apie incidentus funkcija, kuri gali pranešti administratoriui apie incidentą pati ar būti integruota su kitomis centralizuotomis tinklo ar IS stebėjimo sistemomis.
6. Integruota „Moloch“ ir MISP sistema gali būti taikoma trims pagrindiniams scenarijams: naudojimui kaip IAS papildant priemonė didelės svarbos tinkluose, naudojimui vietoj IAS, kaip mažiau galinga, bet tuo paprastesnė ir pigesnė alternatyva; ir naudojimui ne kaip tinklo saugos priemonė, o tik kaip analizės priemonė atliekant post factum incidentų tyrimą.

LITERATŪRA

[1] NIST, „SP 800-61: Computer Security: Incident Handling Guide,“ 2012. [Tinkle]. Prieiga per Interntą: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf. [Kreiptasi 15 03 2017].

[2] ISO/IEC, 27000: Information technology — Security techniques — Information security management systems — Overview and vocabulary, 2018.

[3] NIST, „Guide to Integrated Forensic: Techniques into Incident Response,“ 2006. [Tinkle]. Prieiga per Interntą: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-86.pdf#page=6&zoom=auto,-265,743. [Kreiptasi 15 03 2017].

[4] ISO/IEC, 27001: Information technology Security techniques -- Information security management systems Requirements, 2013.

[5] NIST, SP 800-53: Security and Privacy Controls for Federal Information Systems and Organizations - Revision 4, 2013.

[6] CIS, Security and Privacy Controls for Federal Information Systems and Organizations - Version 6.1, 2016.

[7] „What is Network Security,“ [Tinkle]. Prieiga per Interntą: https://www.paloaltonetworks.com/documentation/glossary/what-is-network-security. [Kreiptasi 02 Sausio 2017].

[8] „2016: Current State of Cybercrime,“ 2016. [Tinkle]. Prieiga per Interntą: https://www.rsa.com/content/dam/rsa/PDF/2016/05/2016-current-state-of-cybercrime.pdf. [Kreiptasi 02 Sausio 2017].

[9] „The Security Wheel,“ [Tinkle]. Prieiga per Interntą: http://www.cisco.com/web/learning/netacad/demos/FNSDemo1_1/ch1/1_3_1/index.html. [Kreiptasi 02 Sausio 2017].

[10] T. Kaur, V. Malhotra ir D. D. Singh, „Comparison of network security tools - Firewall, Intrusion Detection System and Honeypot,“ *International Journal of Enhanced Research in Science Technology & Engineering,* t. 3, nr. 2, pp. 200-204, 2014.

[11] M. E. Whitman, Principles of Information Security, Boston, MA: Course Technology Press, 2011.

[12] C. S. a. J. Smith, Applied Network Security Monitoring, Waltham, MA: Syngress, 2014.

[13] N. Boudriga, Security of mobile communications, Boca Raton, FL: CRC Press, 2010.

[14] R. Oppliger, „Internet Security: FIREWALLS and BEYOND,“ *Communications of the ACM,* t. 5, nr. 94, p. 40, 1997.

[15] K. Scarfone ir P. Hoffman, „Guidelines on Firewalls and Firewall Policy,“ 2009. [Tinkle]. Prieiga per Interntą: http://csrc.nist.gov/publications/nistpubs/800-41-Rev1/sp800-41-rev1.pdf. [Kreiptasi 13 Sausio 2017].

[16] „ISO/IEC 7498-1,“ 1996. [Tinkle]. Prieiga per Interntą: http://www.ecma-international.org/activities/Communications/TG11/s020269e.pdf. [Kreiptasi 02 Sausio 2017].

[17] The Center for Internet Security, „Critical Security Controls for Effective Cyber Defense,“ 2016. [Tinkle]. Prieiga per Interntą: https://www.cisecurity.org/critical-controls/documents/CSC-MASTER-VER61-FINAL.pdf. [Kreiptasi 14 Sausio 2018].

[18] A. Sperotto, Flow-Based Intrusion Detection, Zutphen, Netherlands: Wöhrmann Print Service, 2010.

[19] SANS Institute, „Understanding Intrusion Detection Systems,“ 2001. [Tinkle]. Prieiga per Interntą: https://www.sans.org/reading-room/whitepapers/detection/understanding-intrusion-

detection-systems-337. [Kreiptasi 03 Sausio 2017].

[20] L. R. Even, „What is a Honeypot?," 2000. [Tinkle]. Prieiga per Interntą: https://www.sans.org/security-resources/idfaq/what-is-a-honeypot/1/9. [Kreiptasi 03 Sausio 2017].

[21] K. Scarfone ir P. Mell, „Guide to Intrusion Detection and Prevention Systems (IDPS)," 2007. [Tinkle]. Prieiga per Interntą: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf. [Kreiptasi 10 Sausio 2017].

[22] D. C. Huy Nguyen, „Network Anomaly Detection: Flow-based or Packet-based Approach?," 2010. [Tinkle]. Prieiga per Interntą: https://arxiv.org/abs/1007.1266. [Kreiptasi 03 Sausio 2017].

[23] P. S. Kumar, „Establishing a valuable method of packet capture and packet analyzer tools in firewall," *International Journal of Research Studies in Computing,* t. 1, nr. 1, pp. 11-20, 2012.

[24] M. Koch, „Implementing Full Packet Capture," 2016. [Tinkle]. Prieiga per Interntą: https://www.sans.org/reading-room/whitepapers/forensics/implementing-full-packet-capture-37392. [Kreiptasi 10 Sausio 2017].

[25] K. Gennuso, „Shedding Light on Security Incidents Using Network Flows," 2012. [Tinkle]. Prieiga per Interntą: https://www.sans.org/reading-room/whitepapers/networkdevs/shedding-light-security-incidents-network-flows-33935. [Kreiptasi 10 Sausio 2017].

[26] C. Fuchs, „Implications of Deep Packet Inspection (DPI) Internet Surveillance for Society," 2012. [Tinkle]. Prieiga per Interntą: http://www.projectpact.eu/privacy-security-research-paper-series/%231_Privacy_and_Security_Research_Paper_Series.pdf. [Kreiptasi 02 Sausio 2017].

[27] „Lietuvos Respublikos asmens duomenų teisinės apsaugos įstatymas," [Tinkle]. Prieiga per Interntą: https://www.e-tar.lt/portal/lt/legalAct/TAR.5368B592234C/lGOrBAvuZc. [Kreiptasi 02 Sausio 2017].

[28] „Lietuvos Respublikos elektroninių ryšių įstatymas," [Tinkle]. Prieiga per Interntą: https://www.e-tar.lt/portal/lt/legalAct/TAR.82D8168D3049/BWaAwPRnRd. [Kreiptasi 02 Sausio 2017].

[29] „Lietuvos Respublikos kibernetinio saugumo įstatymas," [Tinkle]. Prieiga per Interntą: https://www.e-tar.lt/portal/lt/legalAct/5468a25089ef11e4a98a9f2247652cf4. [Kreiptasi 02 Sausio 2017].

[30] G. Tamašauskaitė, „Informacinių technologijų poveikis darbuotojo teisei į privatų gyvenimą," 2013. [Tinkle]. Prieiga per Interntą: http://www.zurnalai.vu.lt/teise/article/viewFile/1605/987. [Kreiptasi 03 Sausio 2017].

[31] „Europos parlamento ir tarybos direktyva (95/46/EB)," 24 Spalio 1995. [Tinkle]. Prieiga per Interntą: http://eur-lex.europa.eu/legal-content/LT/TXT/PDF/?uri=CELEX:31995L0046&from=lt. [Kreiptasi 03 Sausio 2017].

[32] „Europos parlamento ir tarybos reglamentas (ES) 2016/679," 27 Balandžio 2016. [Tinkle]. Prieiga per Interntą: http://eur-lex.europa.eu/legal-content/LT/TXT/PDF/?uri=CELEX:32016R0679&from=LT. [Kreiptasi 10 Sausio 2017].

[33] A. Cormack, „Incident Response: Protecting Individual Rights Under the General Data Protection Regulation," 2016. [Tinkle]. Prieiga per Interntą: https://script-ed.org/article/incident-response-protecting-individual-rights-under-the-general-data-protection-regulation/. [Kreiptasi 13 Sausio 2017].

[34] ENISA, Threat Landscape Report 2017: 15 Top Cyber-Threats and Trends, 2018.

[35] P. Haag, „Watch your Flows with NfSen and NFDUMP," 2005. [Tinkle]. Prieiga per Interntą: http://meetings.ripe.net/ripe-50/presentations/ripe50-plenary-tue-nfsen-nfdump.pdf. [Kreiptasi 10 Sausio 2017].

[36] „Incident Definition," [Tinkle]. Prieiga per Interntą: https://www.us-cert.gov/government-users/compliance-and-reporting/incident-definition. [Kreiptasi 15 03 2017].

[37] ISO/IEC, 27035: Information technology -- Security techniques -- Information security incident management, 2011.

PRIEDAS A. INTEGRACIJOS MODULIO PROGRAMINIS KODAS

```
/*****************************************************************************/
/*
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this Software except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
'use strict';

var fs             = require('fs')
  , csv            = require('csv')
  , wiseSource     = require('./wiseSource.js')
  , util           = require('util')
  , HashTable      = require('hashtable')
  ;

////////////////////////////////////////////////////////////////////////////
/
function MispSource (api, section) {
  MispSource.super_.call(this, api, section);
  this.key          = api.getConfig("misp", "key");
  this.host         = api.getConfig("misp", "host");

  if (this.key === undefined) {
    console.log(this.section, "- No key defined");
    return;
  }

  console.log(this.section, "- key: ", this.key);
  console.log(this.section, "- host: ", this.host);

  this.domain      = new HashTable();
  this.ipSrc       = new HashTable();
  this.ipDst       = new HashTable();
  this.url         = new HashTable();
  this.md5         = new HashTable();
  this.emailSrc    = new HashTable();
  this.emailDst    = new HashTable();
  this.cacheTimeout = -1;

  this.api.addSource("misp", this);

  this.idField                                                    =
this.api.addField("field:misp.id;db:misp.id;kind:integer;friendly:ID;help:Misp
Event ID;count:false");
  this.categoryField  =  this.api.addField("field:misp.category;db:misp.category-
term;kind:termfield;friendly:Category;help:Misp Category;count:false");
  this.domainField     =     this.api.addField("field:misp.domain;db:misp.domain-
term;kind:termfield;friendly:Domain;help:Misp Domain;count:false");
  this.ipField                                                    =
this.api.addField("field:misp.ip;db:misp.ip;kind:ip;friendly:IP;help:Misp
IP;count:false");
```

```
  this.urlField          =          this.api.addField("field:misp.url;db:misp.url-
term;kind:termfield;friendly:URL;help:Misp URL;count:false");
  this.md5Field          =          this.api.addField("field:misp.md5;db:misp.md5-
term;kind:termfield;friendly:MD5;help:Misp MD5;count:false");
  this.emailField       =       this.api.addField("field:misp.email;db:misp.email-
term;kind:termfield;friendly:Email;help:Misp Email;count:false");

  this.api.addView("misp",
  "if (session.misp)\n" +
  "  div.sessionDetailMeta.bold Misp\n" +
  "  dl.sessionDetailMeta\n" +
  "    +arrayList(session.misp, 'id', 'ID', 'misp.id')\n" +
  "    +arrayList(session.misp, 'category', 'Category', 'misp.category')\n" +
  "    +arrayList(session.misp, 'domain-term', 'Domain', 'misp.domain')\n" +
  "    +arrayList(session.misp, 'ip', 'IP', 'misp.ip')\n" +
  "    +arrayList(session.misp, 'url-term', 'URL', 'misp.url')\n" +
  "    +arrayList(session.misp, 'md5-term', 'MD5', 'misp.md5')\n" +
  "    +arrayList(session.misp, 'email-term', 'Email', 'misp.email')\n"
  );

  this.api.addRightClick("mispid",       {name:"Lookup       Event       Details",
url:`http://${this.host}/events/view/%TEXT%`, fields:"misp.id"});

  setImmediate(this.loadFiles.bind(this));
  setInterval(this.loadFiles.bind(this), 60*60*1000); // Reload files every hour
}
util.inherits(MispSource, wiseSource);
////////////////////////////////////////////////////////////////////////////////
/
MispSource.prototype.parse = function (fn, hash, field)
{
  var parser = csv.parse({skip_empty_lines:true}, (err, data) => {
    if (err) {
      console.log(this.section, "- Couldn't parse", fn, "csv", err);
      return;
    }

    console.log(this.section, "total: ", data.length);

    for (var i = 1; i < data.length; i++) {
      var value = hash.get(data[i][4]);
      if (value) {
        var     encoded     =     wiseSource.encode(this.idField,     data[i][1],
this.categoryField, data[i][2]);
        value.num += 2;
        value.buffer = Buffer.concat([value.buffer, encoded]);
      } else {
        var encoded = wiseSource.encode(this.idField, data[i][1],
              this.categoryField, data[i][2], field, data[i][4]);
        hash.put(data[i][4], {num: 3, buffer: encoded});
      }
    }

    console.log(this.section, "- Done Loading", fn);
  });
  fs.createReadStream(fn).pipe(parser);
};

////////////////////////////////////////////////////////////////////////////////
MispSource.prototype.loadFiles = function ()
{
  var headers = {};

  headers['Authorization'] = this.key;
```

```
  console.log(this.section, "- Downloading Files");

  wiseSource.request('http://'             +             this.host             +
'/events/csv/download/false/false/false/false/domain',        '/tmp/domains.txt',
(statusCode) => {
    if (statusCode === 200 || !this.domainLoaded) {
      this.domainLoaded = true;
      this.domain.clear();
      console.log(this.section, "- Domains downloaded");
      this.parse("/tmp/domains.txt", this.domain, this.domainField);
    }
    else
    {
      console.log(this.section, "- Domains downloading failed");
    }
  }, headers);

  wiseSource.request('http://'             +             this.host             +
'/events/csv/download/false/false/false/false/ip-src',         '/tmp/ip_src.txt',
(statusCode) => {
    if (statusCode === 200 || !this.ipSrcLoaded) {
      this.ipSrcLoaded = true;
      this.ipSrc.clear();
      console.log(this.section, "- Source IPs downloaded");
      this.parse("/tmp/ip_src.txt", this.ipSrc, this.ipField);
    }
    else
    {
      console.log(this.section, "- Source IPs downloading failed");
    }
  }, headers);

  wiseSource.request('http://'             +             this.host             +
'/events/csv/download/false/false/false/false/ip-dst',         '/tmp/ip_dst.txt',
(statusCode) => {
    if (statusCode === 200 || !this.ipDstLoaded) {
      this.ipDstLoaded = true;
      this.ipDst.clear();
      console.log(this.section, "- Destination IPs downloaded");
      this.parse("/tmp/ip_dst.txt", this.ipDst, this.ipField);
    }
    else
    {
      console.log(this.section, "- Destination IPs downloading failed");
    }
  }, headers);

  wiseSource.request('http://'             +             this.host             +
'/events/csv/download/false/false/false/false/url',  '/tmp/url.txt',  (statusCode)
=> {
    if (statusCode === 200 || !this.urlLoaded) {
      this.urlLoaded = true;
      this.url.clear();
      console.log(this.section, "- URL downloaded");
      this.parse("/tmp/url.txt", this.url);
    }
    else
    {
      console.log(this.section, "- URL downloading failed");
    }
  }, headers);
```

```
  wiseSource.request('http://'              +              this.host              +
'/events/csv/download/false/false/false/false/md5',  '/tmp/md5.txt',  (statusCode)
=> {
    if (statusCode === 200 || !this.md5Loaded) {
      this.md5Loaded = true;
      this.md5.clear();
      console.log(this.section, "- Md5 downloaded");
      this.parse("/tmp/md5.txt", this.md5);
    }
    else
    {
      console.log(this.section, "- Md5 downloading failed");
    }
  }, headers);

  wiseSource.request('http://'              +              this.host              +
'/events/csv/download/false/false/false/false/email-src',    '/tmp/email_src.txt',
(statusCode) => {
    if (statusCode === 200 || !this.emailSrcLoaded) {
      this.emailSrcLoaded = true;
      this.emailSrc.clear();
      console.log(this.section, "- Source IPs downloaded");
      this.parse("/tmp/email_src.txt", this.emailSrc, this.emailField);
    }
    else
    {
      console.log(this.section, "- Source IPs downloading failed");
    }
  }, headers);

  wiseSource.request('http://'              +              this.host              +
'/events/csv/download/false/false/false/false/email-dst',    '/tmp/email_dst.txt',
(statusCode) => {
    if (statusCode === 200 || !this.emailDstLoaded) {
      this.emailDstLoaded = true;
      this.emailDst.clear();
      console.log(this.section, "- Destination IPs downloaded");
      this.parse("/tmp/email_dst.txt", this.emailDst, this.emailField);
    }
    else
    {
      console.log(this.section, "- Destination IPs downloading failed");
    }
  }, headers);
};
////////////////////////////////////////////////////////////////////////////////
/
MispSource.prototype.getDomain = function(domain, cb) {
  console.log(this.section, "- getDomain", domain);

  if                          (this.domain.get(domain)                          ||
this.domain.get(domain.substring(domain.indexOf(".")+1)))
  {
    console.log(this.section, "- MATCH!");
  }

  cb(null,                        this.domain.get(domain)                         ||
this.domain.get(domain.substring(domain.indexOf(".")+1)));
};
////////////////////////////////////////////////////////////////////////////////
/

MispSource.prototype.getIp = function(ip, cb) {
  console.log(this.section, "- getIp", ip);
```

```
  var match = this.ipSrc.get(ip) || this.ipDst.get(ip);

  if (match)
  {
    console.log(this.section, "- MATCH!");
    cb(null, match);
  }
  else
    cb(null, null);
};
////////////////////////////////////////////////////////////////////////////////
/
MispSource.prototype.getURL = function(url, cb) {
  console.log(this.section, "- getURL", url);

  if (this.url.get(url))
  {
    console.log(this.section, "- MATCH!");
  }

  cb(null, this.url.get(url));
};
////////////////////////////////////////////////////////////////////////////////
/
MispSource.prototype.getMd5 = function(md5, cb) {
  console.log(this.section, "- getMd5", md5);

  if (this.md5.get(md5))
  {
    console.log(this.section, "- MATCH!");
  }

  cb(null, this.md5.get(md5));
};
//
////////////////////////////////////////////////////////////////////////////////
/
MispSource.prototype.dump = function(res) {
  ["ips", "domains"].forEach((ckey) => {
    res.write(`${ckey}:\n`);
    this[ckey].forEach((key, value) => {
      var str = `{key: "${key}", ops:\n` +
        wiseSource.result2Str(wiseSource.combineResults([value])) + "},\n";
      res.write(str);
    });
  });
  res.end();
};
////////////////////////////////////////////////////////////////////////////////
/
exports.initSource = function(api) {
  var source = new MispSource(api, "misp");
};
```