RĪGAS TEHNISKĀ UNIVERSITĀTE

Vladislavs Minkevičs

UZ LIELO DATU PARADIGMU BALSTĪTI RISINĀJUMI INFORMĀCIJAS SISTĒMU DROŠĪBAS PĀRVALDĪBAS UZLABOŠANAI

Promocijas darbs

RTU Izdevniecība
Rīga 2026

RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes, informācijas tehnoloģijas un enerģētikas fakultāte
Informācijas tehnoloģijas institūts

Vladislavs Minkevičs

Doktora studiju programmas „Informācijas tehnoloģija" doktorants

UZ LIELO DATU PARADIGMU BALSTĪTI RISINĀJUMI INFORMĀCIJAS SISTĒMU DROŠĪBAS PĀRVALDĪBAS UZLABOŠANAI

Promocijas darbs

Zinātniskais vadītājs

Asociētais profesors *Dr.. sc. ing.*

JĀNIS KAMPARS

Rīga 2026

RĪGAS TEHNISKĀ UNIVERSITĀTE
DATORZINĀTNES UN INFORMĀCIJAS TEHNOLOĢIJAS FAKULTĀTE
Informācijas tehnoloģijas institūts

UZ LIELO DATU PARADIGMU BALSTĪTI RISINĀJUMI INFORMĀCIJAS SISTĒMU DROŠĪBAS PĀRVALDĪBAS UZLABOŠANAI

Vladislavs Minkevičs

ANOTĀCIJA

Darbs veltīts aktuālas mūsdienu problēmas risināšanai, kas saistīta ar informācijas sistēmu drošības risku mazināšanu organizācijā. Darba rezultātā tika izstrādāts adaptīvais drošības pārvaldības modelis, kā arī tehnoloģiskā platforma drošības risku mazināšanai, kas, reaģējot uz incidentiem, veic atbilstošas adaptācijas. Izstrādātā tehnoloģiskā platforma un piedāvātais moduļu komplekts apvieno daudzus informācijas avotus un balstās lielo datu paradigmā, nodrošinot adekvātu informācijas sistēmu drošību, tai skaitā izpildot gan Eiropas, gan Latvijas likumdošanas prasības kiberaizsardzības jomā. Izstrādātais risinājums tika aprobēts Rīgas Tehniskajā universitātē (RTU) un ir apliecinājis savu efektivitāti, uzlabojot spēju reaģēt uz identificētiem incidentiem un samazinot Latvijas informācijas tehnoloģiju drošības incidentu novēršanas institūcijas (*CERT.LV*) paziņojumus par inficētām iekārtām RTU tīklā par 99,98 %, kā arī palielinot spēju nekavējoties reaģēt uz drošības incidentiem. Automatizēta un operatīva potenciāli inficētas ierīces lietotāja iesaiste incidenta risināšanā, izmantojot piedāvāto tehnoloģisko platformu, nodrošināja drošības risku mazināšanu, laikus ieviešot preventīvas vai korektīvas darbības. Darbā piedāvātā tehnoloģiskā platforma ir mērogojama un balstīta atvērtajās tehnoloģijās un lielajos datos. Platformu var lietot jebkurā organizācijā vai iestādē, kas var nodrošināt piekļuvi auditācijas pierakstiem un savai tīkla datu plūsmai, jo platformas darbības pamatā ir datu analīze no dažādiem avotiem, tai skaitā ielaušanās noteikšanas, novēršanas sistēmām, ugunsmūriem, tīkla datiem, kā arī jebkuriem citiem citiem datu avotiem, kas ir spējīgi veidot auditācijas pierakstus. Platforma nodrošina iesūtīto datu agregāciju un analīzi, kā arī pielieto uz mašīnmācīšanās algoritmiem balstītas pieejas iepriekš neidentificētu apdraudējumu detektēšanai, analizējot gan domēnu

pieprasījumus, gan tīkla plūsmas datus. Piedāvātā platforma būtiski atvieglo drošības risku pārvaldību organizācijā, laikus atklājot un dažādos veidos automātiski reaģējot uz identificētajiem drošības riskiem. Darba rezultāti tika aprobēti ne tikai RTU, bet arī Iepirkumu uzraudzības birojā un Centrālajā finanšu un līgumu aģentūrā.

Darba apjoms ir 171 lappuses, darbā ir 49 attēli un 32 tabulas.

RIGA TECHNICAL UNIVERSITY

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

Institute of information technology

SOLUTIONS, BASED ON BIG DATA PARADIGM TO IMPROVE INFORMATION SYSTEM'S SECURITY

Vladislavs Minkevičs

ANNOTATION

The Thesis is devoted to solving a topical contemporary problem related to the mitigation of information systems security risks in an organisation. As a result, an adaptive security management model was developed, as well as a technological platform for mitigating security risks, which performs appropriate adaptations in response to incidents. The developed technological platform and the proposed set of modules combine many information sources and are based on the big data paradigm, ensuring adequate security of information systems, including meeting the requirements of both European and Latvian legislation in the field of cybersecurity. The developed solution has been tested at Riga Technical University (RTU) and has proven its effectiveness by improving the ability to respond to identified incidents and reducing the number of Latvian Information Technology Security Incident Response Team (CERT.LV) notifications about infected devices in the RTU network by 99.98 %, as well as increasing the ability to respond immediately to security incidents. Fast and automated involvement of the user of a potentially infected device using the offered technological platform ensured mitigation of security risks by implementation of preventive or corrective actions in a timely manner. The technological platform proposed in the work is scalable and based on open technologies and big data. The platform can be used in any organisation or institution that can provide access to audit trails and its network traffic, as the platform is based on the analysis of data from various sources, including intrusion detection, prevention systems, firewalls, network data, as well as any other data sources that are capable of creating audit trails. The platform collects and analyses the data received, and applies machine learning-based techniques to detect unknown threats by analysing both domain requests and network data. The platform reduces security risk by detecting and automatically responding to identified security risks in a variety

of ways. In addition to RTU, the project was tested in the Procurement Monitoring Bureau and the Central Finance and Contracting Agency.

The thesis consists of 171 pages, 32 tables and 49 pictures.

SATURS

1 IEVADS

1.1 TĒMAS AKTUALITĀTE UN MOTIVĀCIJA

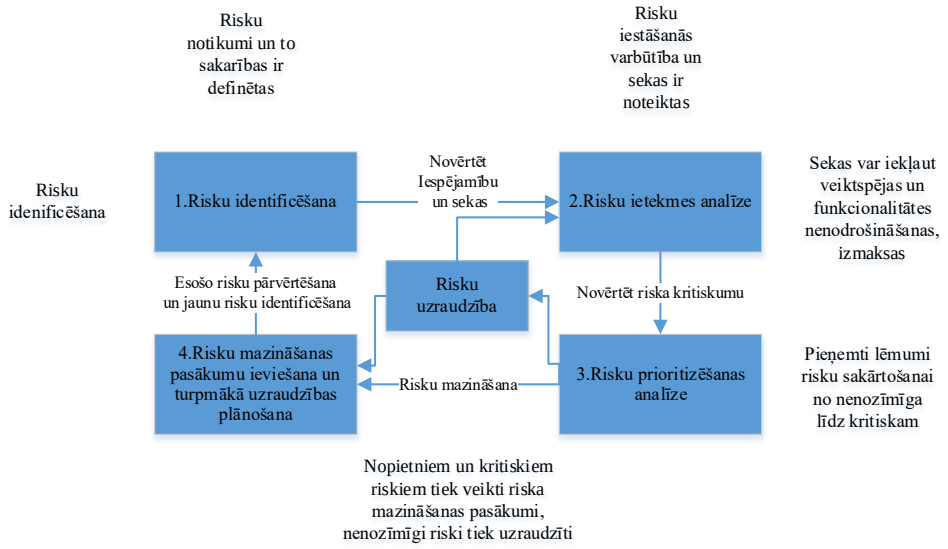
Mūsdienu pasaulē ir grūti iedomāties jomu, kura var pastāvēt bez informācijas tehnoloģiju klātbūtnes. Gan valsts, gan privātais sektors ir atkarīgi no informācijas tehnoloģijām, un, ņemot vērā arvien pieaugošos digitalizācijas procesus pasaulē, šī atkarība arvien pieaug un pieaugs arī nākotnē [1]. Informācijas komunikāciju tehnoloģijas galvenokārt tiek pielietotas, lai atvieglotu cilvēku dzīvi, veicot attālinātas un drošas darbības ar finansēm, saziņai ar valsts institūcijām un citiem mērķiem. Pieaugot atkarībai no informācijas un komunikāciju tehnoloģijām, pieaug arī varbūtība, ka ar to palīdzību ir iespējams nodarīt būtisku kaitējumu publiskās pārvaldes informācijas sistēmām un elektronisko sakaru tīkliem, neitralizēt valsts politisko, ekonomisko, militāro lēmumu pieņemšanas centrus, dezinformēt sabiedrību un izraisīt tehnogēnas avārijas. Tas rada pieaugošu nemilitāru draudu iespējamību ar smagām sekām. Iestādēm, kuru pārziņā ir valsts funkciju nodrošināšanai, būtiski svarīgas valsts informācijas sistēmas drošības jautājumi ir īpaši aktuāli.

Gan valstis atsevišķi, gan Eiropas Savienība kopumā ir pieņēmusi likumus, direktīvas un regulas, kas reglamentē informācijas tehnoloģiju drošību, viena no pēdējām iniciatīvām Eiropas savienībā ir NIS2 direktīvas pieņemšana [2]. Šajos normatīvajos aktos ir definētas minimālās prasības datu aizsardzībai konfidencialitātes, integritātes un pieejamības jomā, kas kopumā uzlabo kiberdrošību. Kiberdrošību var definēt vairākos veidos. Saskaņā ar [3] kiberdrošība ir spēja aizsargāt tīklus, ierīces un datus no neatļautas piekļuves vai noziedzīgas izmantošanas, kā arī spēja nodrošināt informācijas konfidencialitāti, integritāti un pieejamību. Kiberdrošība var tikt iedalīta tīkla drošībā, informācijas drošībā un citās kategorijās.

Saskaņā ar drošības kompāniju MITRE Corporation, kiberdrošības risku mazināšanu nepieciešams plānot, ieviest un uzraudzīt tās progresu (1.1.att.) [4].
Risku mazināšana iekļauj:

- Pieņemšanu – atzīt konkrēta riska esamību un apzināti lemt par tā pieņemšanu, neveicot darbības tā kontrolēšanai. Šajā gadījumā nepieciešama sistēmas īpašnieku piekrišana.
- Samazināšanu – pielāgot prasības vai ierobežojumus, lai novērstu vai samazinātu risku. Šos pielāgojumus veic, mainot finansējumu, izpildes grafikus vai tehniskās prasības.
- Kontroli – īstenot darbības, lai samazinātu riska ietekmi vai tā iestāšanās varbūtību.
- Nodošanu – nodot atbildību un pilnvaras citai ieinteresētajai personai, kura vēlas uzņemties risku par samaksu, piemēram apdrošināšanas kompānijai.

- Uzraudzību – novērot vidi, lai identificētu izmaiņas, kas ietekmē riska iestāšanās varbūtību un/vai ietekmi.



1.1.att. Fundamentālie risku pārvaldības soļi (adaptēts no *[4]*)

Tomēr arvien vairāk ir dzirdēts par dažādiem kiberdrošības incidentiem, kuri, dažkārt skar pat visu pasauli, piemēram, SolarWinds [5], Colonial Pipeline [6] un pieejas atteices uzbrukumi Latvijas valsts iestādēm [7]. Šādu incidentu organizatori parasti ir labi finansētas noziedzīgās organizācijas, dažādi noziedzīgi grupējumi un pat valstis.

Robotu tīkli mūsdienās ir kļuvuši par vienu no lielākajiem kiberdrošības draudiem. Saskaņā ar ENISA pārskatu par 2019-2020.gadu, tika identificēti vairāk kā 17.tūkst. funkcionējoši robotu tīklu serveri [8].

Ņemot vērā augstākminēto, jebkurai iestādei, organizācijai vai privātuzņēmumam nepieciešams pievērst pienācīgu uzmanību informācijas sistēmu drošības pārvaldībai. Valsts un pašvaldību iestāžu darbību nosaka ārējie normatīvie akti, piemēram, Nacionālās kiberdrošības likums [9], MK noteikumi "Kārtība, kādā tiek nodrošināta informācijas un komunikācijas tehnoloģiju sistēmu atbilstība minimālajām drošības prasībām" [10] u.c., uz privātuzņēmumu, ja tas apstrādā personu datus, attiecas Vispārīgā datu aizsardzības regula [11], kuras 5.panta f) apakšpunkts nosaka, ka apstrādājot personu datus, nepieciešams tos aizsargāt, izmantojot atbilstošus tehniskos un organizatoriskos pasākumus. Bieži vien tikai notikušie incidenti liek

privātuzņēmumiem veikt darbības, lai ieviestu atbilstošus drošības pasākumus. Diemžēl presē atrodama informācija par incidentiem, kuri skar personu datus, piemēram, Marriott viesnīcu tīkla datu incidents [12], kura rezultātā viesu vārdi, uzvārdi, lojalitātes informācija un citi personu dati nonākuši neautorizētu personu rīcībā. Jāatzīmē, ka Marriott viesnīcu tīkla incidents ir jau otrs šāda veida incidents divu gadu laikā un viesnīcu tīkls par pārkāpumiem jau ir saņēmis sodu 18.4 milj. britu mārciņu apmērā [13]. Līdzīgs gadījums ir noticis arī Lietuvā, kur "City Bee" lietotāju dati tika nozagti un tirgoti internetā [14]. Latvijā viens no ievērojamākaiem incidentiem, kurš skāris personu datus, bija "Civinity" nekustamo īpašumu pārvaldības uzņēmumu grupas klientu datu zādzība [15]. Šie gadījumi pastāvīgi atgādina par kiberuzbrukumu radīto risku, ko nedrīkst novērtēt par zemu.

Vēl pagājušā gadsimta astoņdesmitos gados neviens nevarēja iedomājies par ielaušanās noteikšanas sistēmu nepieciešamību datortīklu aizsardzībai. Ielaušanās noteikšanas sistēmu (IDS) pirmsākumi ir meklējami ASV gaisa spēkos, kur James P. Anderson bija izstrādājis datortīkla draudu uzraudzības sistēmu [16], kura bija spējīga nepārtraukti skanēt un salīdzināt tīkla datus ar zināmu draudu sarakstu. Pagājušā gadsimta 90. gados *IDS* tehnoloģijas tika attīstītas, un jau bija spējīgas atklāt tīkla uzbrukumus, kuri arvien pieauga gan skaita, gan sarežģītības ziņā [17]. Attīstoties *IDS* sistēmām, daudzas drošības kompānijas sāka piedāvāt mākoņrisinājumus, kur, lai iegūtu ielaušanās noteikšanas funkcionalitāti, vienīgais nosacījums ir nepieciešamo datu nogādāšana mākonī. Lai arī kā, bet šādi pakalpojumi nav pārāk populāri viena iemesla dēļ – dati, kas tiek sūtīti uz mākoni, var saturēt personu datus un sensitīvu uzņēmuma komercinformāciju. Vēl viena aktuāla problēma mūsdienās ir attiecīgo speciālistu trūkums, kuri varētu interpretēt iegūtos rezultātus un pieņemt lēmumus drošības uzlabošanai [18]. Satraukums par arvien pieaugošiem kiberdraudiem ir vērojams arī dažādos aktuālos pētījumos, piemēram, PricewaterhouseCoopers uzņēmumu vadītāju pētījumā [19] par 2021.gadu, 47% no vadītājiem atzīst, ka viņus satrauc kiberdraudi, salīdzinājumā ar 2020.gadu, tie bija tikai 33%.

Viena no mūsdienu organizāciju un iestāžu kļūdām ir dzīvošanas maldīgā drošības sajūtā, uzskatot, ka, ja par mums neviens neko sliktu neraksta presē, tad visam jābūt kārtībā. Diemžēl šāda pieeja nav tālredzīga. Organizācijām nepieciešams pielietot preventīvu pasākumu kopumu, lai pasargātu gan komercnoslēpumu, gan organizācijas darbinieku privātos datus no to nonākšanas nepilnvarotu personu rīcībā. Kā vienu no pirmajiem pasākumiem autors piedāvā nozīmēt par drošības pārvaldību atbildīgo personu.

Par drošības pārvaldību atbildīgajai personai ir nepieciešams plašs rīku klāsts drošības nodrošināšanai, kā arī atbilstošas zināšanas, kā šos rīkus lietot. Ir nepieciešams veikt gan tīkla, gan galaiekārtu uzraudzību, kā arī saprast ugunsmūra konfigurāciju un novērtēt iespējamos apdraudējumu avotus, tai skaitā noteikt ievainojamības ierīcēs un nepārtraukti sekot līdzi aktuālai informācijai informācijas un komunikāciju tehnoloģiju jomā. Tā kā rīku klāsts ir ļoti plašs un katram rīkam ir savas priekšrocības un savi trūkumi, nepieciešams risinājums, kas apvieno pieejamo rīku priekšrocības, samazina to trūkumus, kā arī apstrādā un reaģē uz incidentiem, izmantojot daudzdimensionālu pieeju.

Arī Yakencheck Jason no securityintelligence.com [20] uzskata, ka mūsdienās drošības pārvaldības īstenošanai ar manuālām darbībām vairs nepietiek. Speciālistam kiberdrošības jomā jābūt spējīgam izstrādāt un ieviest automatizācijas līdzekļus drošības uzraudzībai, kā arī jābūt padziļinātām zināšanām par datortīklu, ierīču arhitektūru, ievainojamībām, kiberaizardzības līdzekļiem un to efektivitāti. Lai arī drošības pārvaldības īstenošanai nepieciešamo tehnisko pusi ir iespējams īstenot, izmantojot maksas un bieži vien ērti izmantojamus, dažādu ar kiberdrošību saistītu uzņēmumu risinājumus [21], plašā klāstā pastāv arī bezmaksas atvērtā koda risinājumi un rīki.

Drošības pārvaldības problēmas risināšanai ir nepieciešams ņemt vērā kontekstu, ko veido ārējie datu avoti, dažādi informācijas sistēmu lokālie mērījumi un uzņēmuma izvirzītie mērķi. Pilnvērtīgai konteksta informācijas apstrādei ir nepieciešams izmantot vairākus rīkus, jo gatavie risinājumi nespēj aptvert visu problēmapgabalu. Izaicinājums ir lielapjoma datu integrācija, lietojumprogrammu integrācija, kas papildināta ar mākslīgā intelekta moduļiem. Spējorientētā izstrādes metodoloģija palīdz veidot sistēmas, kas ir informētas gan par kontekstu, gan uzņēmuma mērķiem, tādēļ šī pieeja ir piemērota IS drošības pārvaldības modeļa izstrādei.

Katrs uzņēmums un iestāde vēlas justies droši mūsdienu digitālajā laikmetā, bet līdzekļi, ko šie uzņēmumi un iestādes atvēl informācijas sistēmu drošībai, joprojām tiek uzskaitīti izdevumu nevis investīciju pozīcijās. Bieži vien par kiberdrošību tiek domāts maz, vai netiek domāts vispār, līdz iestājas drošības incidents, vai arī kiberdrošība tiek noadresēta kādai trešai pusei, kurai nav izpratnes par organizācijas darbības mērķiem, un tiek nodrošināta minimālā aizsardzība, aizmirstot, ka kiberdrošība ir nevis stāvoklis, bet process. Saskaņā ar [22] kiberdrošība ietver 5 fāzes: identificēšana (izpratnes veicināšana par iespējamiem kiberriskiem), aizsardzība (risku mazinošo pasākumu īstenošana kritisko resursu aizsardzībai), draudu noteikšana (līdzekļu pielietošana kiberdrošības incidenta noteikšanai), aktīva rīcība

drauda gadījumā (darbību veikšana, lai mazinātu kiberdrošības incidenta ietekmi), kā arī atkopšanās pēc incidenta (pasākumu kopums, lai nodrošinātu servisu darbību pēc incidenta).

Ņemot vērā autora ilggadējo pieredzi par IS drošību atbildīgās personas amatā, kā arī pieredzi, kas ir gūta piedaloties informācijas sistēmu un personu datu aizsardzības auditos, var secināt, ka galvenais izaicinājums uzņēmumos un iestādēs ir draudu noteikšanas un aktīvas rīcības fāzes (1.1.tabula).

1.1.tabula

Kiberdrošības fāzes un risinājumi to nodrošināšanai

Kiberdrošības fāzes	Risinājumi
Identificēšana	*CERT.LV* [23] (Informēšana)
Aizsardzība	2025.gada 25.jūlija Ministru kabineta noteikumi Nr.397 (Minimālās kiberdrošības prasības [10], NIS2 direktīva [2]
Draudu noteikšana	*SOC* sistēmas [24] [25] [26] [27] [28] [29] (*SOC* ārpakalpojuma sniedzēji)
Aktīva rīcība drauda gadījumā	Par IS drošību atbildīgās personas izpratnes un zināšanu līmenis (Izglītība, sertifikācija, praktiskās iemaņas)
Atkopšanās pēc incidenta	2025.gada 25.jūlija Ministru kabineta noteikumi Nr.397 [10] (Nepārtrauktās darbības plānošana), NIS2 direktīva [2]

Ar līdzīgām problēmām bija saskārusies ikkatra organizācija, kurā autors ir vērtējis drošības pārvaldības atbilstību labajai praksei, it īpaši problēma ir aktuāla augstākās izglītības iestādēs, tādēļ, ka parasti drošības pārvaldības mērķim tiek atvēlēt ļoti nelieli līdzekļi un, īstenojot studiju procesu, vairāk tiek domāts par datortīklu ātruma un skaitļošanas jaudu palielinājumu. Lai īstenotu šīs fāzes, nepieciešama pilnvērtīga un nepārtraukta iekšējā tīkla, galaiekārtu un lietotāju darbību tīklā uzraudzība, kā arī jāsaprot, kā identificējams kiberdrošības incidents iekšējā tīklā.

1.2 PROMOCIJAS DARBA MĒRĶIS UN UZDEVUMI

Promocijas darba mērķis ir izstrādāt no konteksta atkarīgu, adaptīvu drošības pārvaldības modeli un šī modeļa tehnisko realizāciju drošības pārvaldības platformā, kura ietver atbilstošos tehniskos risinājumus kiberdrošības vides uzlabošanai.

Izvirzīto mērķi pamato pieņēmums, ka liela daļa organizāciju un iestāžu nav spējīgas pilnvērtīgi īstenot visas 5 kiberdrošības fāzes saskaņā ar [22].

Promocijas darba mērķa sasniegšanai izvirzīti šādi uzdevumi:

1. Novērtēt esošo situāciju IS drošības pārvaldības jomā, veicot literatūras analīzi un notikušo kiberdrošības incidentu iemeslu izpēti;
2. Apzināt esošos pētījumus IS drošības pārvaldības jomā, kuros tiek izmantoti gan tradicionālie draudu identificēšanas līdzekļi, gan mākslīgais intelekts nezināmo draudu identificēšanai un pamatot tehnoloģiskās platformas būvēšanas nepieciešamību;
3. Sintezēt prasības no konteksta atkarīgam adaptīvam drošības pārvaldības modelim un tā tehniskajiem risinājumiem;
4. Izstrādāt no konteksta atkarīgu adaptīvu drošības pārvaldības modeli;
5. Aprobēt definēto modeli, izstrādājot atbilstoša tehniskā risinājuma (platformas) implementāciju augstākās izglītības iestādē;
6. Novērtēt izstrādātā modeļa un platformas efektivitāti.

1.3 PĒTĪJUMA OBJEKTS UN PRIEKŠMETS

Uzņēmumi un iestādes īsteno drošības pārvaldību ar mērķi nodrošināt atbilstību Eiropas un Latvijas normatīvajiem aktiem, kā arī, lai pasargātu privātus datus un komercnoslēpumu no nesankcionētas piekļuves, nodrošinātu tiem atbilstošu pieejamību un pasargātu datus no integritātes zuduma. Autora praksē bieži tika novērota situācija, ka uzņēmums vai iestāde saprot drošības pārvaldību kā specifiska IS drošības risinājuma iegādi, nevis procesu, kurš ir nepārtraukti jāuzlabo.

Promocijas darba pētījumu objekts ir IS drošības pārvaldība.

Promocijas darba pētījumu priekšmets ir adaptīvas, modulāras drošības pārvaldības modelis daudzdimesionālai drošības analīzei.

1.4 TĒZES UN HIPOTĒZE

Promocijas darbā aizstāvamās tēzes:

1. tēze. Lai nodrošinātu efektīvu daudzdimesionālu datu analīzi drošības apdraudējumu identificēšanai, nepieciešams pielietot lielo datu tehnoloģijas un mašīnmācīšanās metodes.

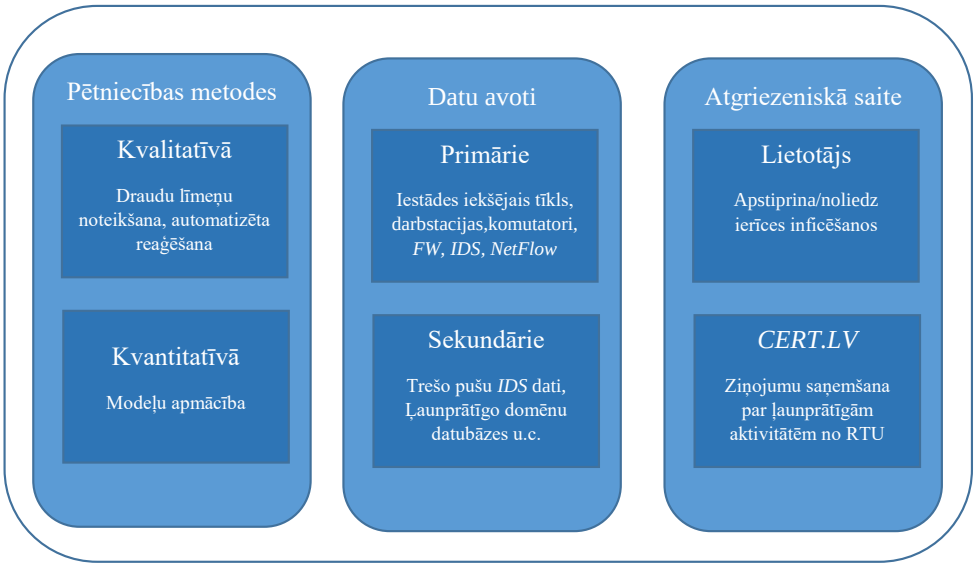
2. tēze. Informācijas sistēmu drošības pārvaldības efektivitāte ir atkarīga no spējas identificēt draudus un reakcijas laika pēc drauda identificēšanas.

3. tēze. Lai nodrošinātu adekvātu drošības pārvaldību, nepieciešams izmantot automatizētas sistēmas, kuras reaģē uz drošības draudiem.

Promocijas darbā aizstāvamā **hipotēze**: Apvienojot vairākus datu avotus, specializētus draudu identificēšanas modeļus un platformas, tiek iegūta pilnvērtīga drošības incidentu identificēšana, salīdzinot ar individuālu šim mērķim paredzētu risinājumu izmantošanu.

1.5 PĒTĪJUMA METODIKA

Darba pamatā ir informācijas sistēmu drošības pārvaldības problēmu identificēšana un šo problēmu risinājumu piedāvāšana. Lai risinātu augstākminētās problēmas, darbā tika pielietotas gan kvalitatīvās, gan kvantitatīvās pētniecības metodes (*1.2.*att.). Draudu līmeņu noteikšana, balstoties uz dažādiem avotiem, un reaģēšana uz tiem, kā arī atgriezeniskās saites ar lietotājiem izmantošana pētījumā ir definētas kā kvalitatīvās pētniecības metodes. Savukārt ļaunprātīgas aktivitātes moduļu un to identificēšanas algoritmu apmācība pētījumā ir definēta kā kvantitatīvā pētniecības metode. Par primārajiem datu avotiem uzskatāmi dati, kas tika ievākti no iestādes iekšējā tīkla informācijas, lietotāju darbstacijām, komunikatoru datiem, ugunsmūra datiem, ielaušanās noteikšanas sistēmas datiem, *NetFlow* datiem, veiksmīgas/neveiksmīgas autentifickācijas kā arī citiem auditācijas pierakstu failiem un citiem datu avotiem. Sekundārie dati [30] [31] tika izmantoti, lai apmācītu mašīnmācīšanās moduļus, klasificējot tos kā leģitīmus vai ļaunprātīgus domēnus. Papildu ļaunprātīgie domēnu vārdi (sekundārie dati) tika iegūti no iestādes izmantojamā ugunsmūra ar ielaušanās novēršanas funkcionalitāti, kā arī veikta izpēte domēna vārdu pieprasījumu datos un aizdomīgu domēna vārdu pieprasījumu manuāla klasifikācija un salīdzināšana ar ļaunprātīgo domēnu datubāzēm.



1.2.att. Pētījuma metodika

Atgriezneniskās saites nodibināšanai ar informācijas sistēmu lietotājiem tika izmantotas *Microsoft Office Forms* izstrādātas formas, kurās potenciāli inficētās ierīces lietotājam tika uzdoti jautājumi par to, vai lietotājs ir noskenējis ierīci ar rekomendētajiem antivīrusiem un vai ierīcē tika atklāts ļaunprātīgais kods. Papildu tika ņemti vērā arī *CERT.LV* paziņojumi par ļaunprātīgām aktivitātēm no iestādes. Šādā veidā (aizpildītas formas un *CERT.LV* ziņojumi) tika panākta lielāka apmācīto mašīnmācīšanās modeļu precizitāte, pārapmācot tos.

Adaptīvas IS drošības pārvaldības pamatā esošais Informācijas drošības pārvaldības spējas konceptuālais modelis (3.3.attēls), tika izstrādāts, izmantojot *CDD* [32] pieeju, jo tā ir piemērota adaptīvu risinājumu specificēšanai un implementēšanai. Darbā izstrādātajam vispārīgajam drošības pārvaldības spēju modelim, kā arī specifiskajam RTU pielāgotajam spēju modelim pamatā ir drošības riskus identificējošie moduļi, kā arī dažādi citi elementi drošības pārvaldības īstenošanai.

Darbā tika pētīti uz mašīnmācīšanos balstīti moduļi, kuri identificē ļaunprātīgus domēnus (*DGA*) un ļaunprātīgu aktivitāti *NetFlow* datos (NFAI). *DGA* tika veikti eksperimenti ar domēnu atlases kritērijiem, kā arī pazīmju kopu veidošanu. *DGA* identificēšanai tika izvēlēti dažādi klasifikatori: atbalsta vektoru mašīna *(Support Vector Machine* (SVC)), Neironu tīkli (*NNC*), Lēmumu koki (*DTC*) un Lēmumu meži (*RFC*). Lai novērtētu klasifikatoru veikumu, katram no tiem tika mērīta ticamība (*Accuracy*), pārklājums (*Recall),* F1-mērs (*F1 Score*),

pielietojot šķērsvalidāciju (*Cross validatio*n). Eksperimetu rezultātā tika salīdzināti apmācīto klasifikatoru sniegtie rezultāti ar RTU izmantotā ugunsmūra ar IPS funkcionalitāti datiem. Tāpat tika veikti eksperimenti ar, uz mašīnmācīšanos balstītu, *NFAI* moduli, kura mērķis ir ļaunprātīgas darbības identificēšana tīkla datos.

DGA identificēšanas moduļa rezultāti tika salīdzināti ar RTU izmantotā ugunsmūra datiem. Rezultāti liecina, ka, pielietojot mašīmācīšanās modeļus *ISMS* efektivitāte uzlabojas, jo tas ļauj identificēt apdraudējumus dažādos līmeņos, agregējot datus. *ISMS* moduļi var iekļaut arī organizācijas mākoņpakalpojumos esošos datus, kuri ugunsmūrim nav pieejami, tadejādi vēl papildus uzlabojot kiberdrošības apdraudējumu identicēšanu.

Pētījuma rezultātā tika definēts no konteksta atkarīgs, adaptīvs drošības pārvaldības modelis un izstrādāta tam atbilstoša uz lielajiem datiem bāzēta, mērogojama drošības pārvaldības sistēmas platforma (ISMS), kurā iespējams integrēt neatkarīgus draudu noteikšanas un novēršanas moduļus atbilstoši organizācijas vajadzībām. *ISMS* platforma šobrīd tiek aktīvi lietota RTU, lai novērstu kiberdrošības apdraudējumus.

1.6 DARBA ZINĀTNISKIE JAUNIEGUVUMI

Darba zinātniskie jaunieguvumi ir šādi:

1. Izstrādāts no konteksta atkarīgs, adaptīvs IS drošības pārvaldības modelis un tā tehniskā realizācija.
2. Sagatavotas apmācību datu kopas ļaunprātīga DNS identificēšanai, kā arī ļaunprātīga koda darbības identificēšanai tīkla datos.
3. Izstrādāta unikāla pazīmju kopa ļaunprārtīga DNS pieprasījuma identificēšanai, ļaunprātīga koda darbības identificēšanai tīkla datos.
4. Izstrādāts multidimensionālu draudu agregācijas algoritms, kurš tika intergēts *ISMS* platformā, nodrošinot reakciju balstoties uz identificētā drauda kritiskumu.
5. Radīta pieeja automatizēti iesaistīt galalietotāju kiberincidentu risināšanā, tai skaitā nodrošinot galalietotājam atgriezenisko saiti.

1.7 DARBA PRAKTISKĀ NOZĪME

Izstrādāts IS drošības pārvaldības modelis un tā tehniskā realizācija, sniedzot atbalstu izpildīt NIST definēto [22] draudu noteikšanas un aktīvas rīcības fāzes ieviešanu. Platformas implementācija ir veikta izmantojot galvenokārt atvērtā koda risinājumus.

Veikta izstrādātās platformas aprobācija RTU. Pielietota uz lielo datu paradigmu balstīta daudzdimensionāla datu analīze un agregācija, nodrošinot platformas mērogojamamību.

Platforma ir paplašināma ar apakšmoduļiem atbilstoši organizācijas vajadzībām. Uz mašīnmācīšanos balstīts, robottīkla domēna identifikācijas modulis (*DGA*) ir aprobēts gan RTU, gan arī citās iestādēs, pierādot savu efektivitāti.

Novērtēta platformas efektivitāte informācijas sistēmu drošības nodrošināšanā.

1.8 DARBA APROBĀCIJA

Pētījumos iegūtie rezultāti tika prezentēti divpadsmit konferencēs:

1) Rīgas Tehniskās universitātes 45. zinātniskā konference, Rīga (Latvija), 2004. gada 14.–16. oktobrī. Referāts "Efektīva risku menedžmenta meklējumi".
2) 19. Eiropas konference modelēšanā un simulācijā, Rīga (Latvija), 2005. gada 1.–4. jūnijā. Referāts "Riska menedžmenta modelēšana unificētām draudu apstrādes sistēmām".
3) Rīgas Tehniskās universitātes 46. zinātniskā konference, Rīga (Latvija), 2005. gada 13.–15. oktobrī. Referātu "Riska menedžmenta modelēšana, izmantojot neironu tīklus".
4) Rīgas Tehniskās universitātes 47. zinātniskā konference, Rīga (Latvija), 2006. gada 12.–14. oktobrī. Referāts "Reāla laika riska menedžmenta izmantošana organizācijā".
5) 6. *Eurosim* kongress "*Eurosim 2007*", Ļubļana (Slovēnija), 2007. gada 9.–13. septembrī. Referāts "Reāla laika riska menedžmenta sistēmas modelēšana".
6) Rīgas Tehniskās universitātes 48. zinātniskā konference, Rīga (Latvija), 2007. gada 11.–13. oktobrī. Referāts "Reāla laika riska menedžmenta izmantošana organizācijā".
7) Rīgas Tehniskās universitātes 48. zinātniskā konference, Rīga (Latvija), 2008. gada 13.–15. oktobrī. Referāts "Reāla laika riska menedžmenta modelis".
8) *Modelling IT Security Risk Management in Academic Environment. IEEE Workshop on advances in information, electronic and electrical engineering* (*AIEEE'2017*), Rīga, 2017. gada 24. novembrī, Rīgā.
9) *IS Security Governance Capability Design for Higher Education Organization. 59th International Scientific Conference on Information Technology and Management Science of Riga-Technical-University* (*ITMS*), Rīga, 2018. gada 12.–14. novembrī.
10) *ICEIS 2020 – 22nd International Conference on Enterprise Information Systems*. Referāts "*Methods, models and techniques to improve information system's security in large organizations*", Prāga, Čehija (attālināti), 2020. gada 5.–7. maijā.
11) *Artificial intelligence and big data driven IS security management solution with applications in higher education organizations. 17th International Conference on Network and Service Management*, Izmira, Turcija, 2021. gada 25.–29. oktobrī.

12) *Managing Information System Security in Higher Education Organizations, IEEE Workshop on Advances in Information, Electronic and Electrical Engineering* (*AIEEE*), Viļņa, Lietuva, 2023. gada 27.–29. aprīlī.
13) Praktiskā pieredze *SOC* izveidē, izmantojot atvērtā koda risinājumus, *CERT.LV*, 2023. gada 12. decembrī, tiešsaistē: https://cert.lv/lv/2023/11/it-drosibas-seminars-esi-dross-decembri.

Promocijas darbā veikto pētījumu rezultāti ir atspoguļoti trīspadsmit publikācijās:

1) Minkevics V., Slihte J., Vulfs G. "Search for effective risk management". RTU zinātnisko rakstu krājums "Datorzinātne. Datorvadības tehnoloģijas", 5. sēr., 20. sēj., Rīga, RTU, 2004, 174.–180. lpp. (ISSN 1407-7493).
2) Minkevics V., Slihte J., Vulfs G. "Modelling risk management for unified threat management systems" 19th European Conference on Modelling and Simulation Riga 2005, 144.–150. lpp. (ISBN 1-84233-112-4).
3) Minkevics V., Slihte J., Vulfs G. "Modelling risk management system using neural networks". RTU zinātnisko rakstu krājums "Datorzinātne. Datorvadības tehnoloģijas", 5. sēr., 23. sēj., Rīga, RTU, 2005, 66.–72. lpp. (ISSN 1407-7493).
4) Minkevics V., Slihte J., Vulfs G. "Use of real – time risk management in organisation". RTU zinātnisko rakstu krājums "Datorzinātne. Datorvadības tehnoloģijas", 5. sēr., 28. sēj., Rīga, RTU, 2006, 23.–29. lpp. (ISSN 1407-7493).
5) Minkevics V., Slihte J., Vulfs G. "Modelling real – time risk management system". Proceedings of the 6th EUROSIM Congress on Modelling and Simulation, vol. 1. (ISBN-13:978-3-901608-32-2), 414. lpp.
6) Minkevics V., Slihte J., Vulfs G. "Modelling real – time risk management system using associative approach". RTU zinātnisko rakstu krājums "Datorzinātne. Datorvadības tehnoloģijas", 5. sēr., 31. sēj., Rīga, RTU, 2007, 34.–40. lpp. (ISSN 1407-7493).
7) Minkevics V., Vulfs G. "Real-time risk management model". RTU zinātnisko rakstu krājums "Datorzinātne. Datorvadības tehnoloģijas", 36. sēj., Rīga, RTU, 2008, 49.–55. lpp. (ISSN 1407-7493).
8) Minkevičs, V., Šlihte, J. Modelling IT Security Risk Management in Academic Environment. No: 2017 5th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE 2017): Proceedings, Latvija, Riga, 24.–25. novembris, 2017. Piscataway: IEEE, 2017, 5.–8. lpp. ISBN 978-1-5386-4138-5. e-ISBN 978-1-5386-4137-8. Pieejams: doi:10.1109/AIEEE.2017.8270562.

9) Minkevičs V., Kampars J. IS Security Governance Capability Design for Higher Education Organization. No: 2018 59th International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS 2018): Proceedings, Latvija, Rīga, 29.–29. novembris, 2018. Piscataway: IEEE, 2018, 66.–70. lpp. ISBN 978-1-7281-0099-9. e-ISBN 978-1-7281-0098-2. Pieejams: doi:10.1109/ITMS.2018.8552975.
10) Minkevičs V., Kampars J. Methods, models and techniques to improve information system's security in large organizations: included in registration In Proceedings of the 22nd International Conference on Enterprise Information Systems – vol. 1, 2020: ICEIS, 632–639, 2020, ISBN: 978-989-758-423-7.
11) Minkevičs V., Kampars J. Artificial intelligence and big data driven IS security management solution with applications in higher education organizations, 17th International Conference on Network and Service Management, 2021, Izmir, Turkey, doi:10.23919/CNSM52442.2021.9615575,
12) Minkevičs V., Kampars J., Grabis J. Managing Information System Security in Higher Education Organizations, IEEE Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE). 2023. gada 27.–29. aprīlis, Viļņa, Lietuva, doi:10.1109/AIEEE58915.2023.10134911.
13) Minkevičs V., Grabis J. A capability-driven automated cybersecurity monitoring and response system, Frontiers in Computer Science Journal, vol. 7, 2025, doi:10.3389/fcomp.2025.1692263.

Dalība ar procijas darbu saistītajos RTU projektos:

1) Perspektīvās tehnoloģijas noturīgiem un drošiem servisiem (VPP-ARTSS: ARTSS, īstenots no 2020. gada 1. jūlija līdz 2020. gada 31. decembrim https://artss.rtu.lv/);
2) Lielo datu vadīta informācijas un komunikācijas tehnoloģiju drošības pārvaldības risinājuma izstrāde. Projekta realizācijas laiks no 2021. gada 1. janvāra līdz 2023. gada 30. jūnijam (BICTSEMS, http://iti.rtu.lv/vitk/lv/petnieciba/projekti/lielo-datu-vadita-informacijas-un-komunikacijas-tehnologiju-drosibas).

1.9 DARBA STRUKTŪRA

Darbs sastāv no 5 nodaļām, rezultātiem un secinājumiem, literatūras avotu saraksta un 2 pielikumiem.

Ievadā tika sniegts vispārīgs darba raksturojums, pamatota risināmā problēma, definēts darba mērķis, uzdevumi un pierādāmās tēzes, kā arī izklāstīta promocijas darba pētījumu metodika, galvenie rezultāti un darba struktūra.

Pirmajā nodaļā iekļauti promocijas darbā izmantotie pamatjēdzieni, literatūras apskats pamatojoties uz Kofod-Petersen dizaina zinātnes metodoloģiju [33], esošās situācijas apraksts, kā arī iespējamie problēmas risinājumi.

Otrajā nodaļā izstrādāts problēmas risinājums, definējot pamatprasības izstrādājamajai platformai, tai skaitā sniegta tehniskā risinājuma augsta līmeņa arhitektūra. Šajā nodaļā izstrādājamā platforma ir prezentēta, izmantojot spējās izstrādes metodoloģiju [32].

Trešajā nodaļā ir prezentēta informācijas sistēmu drošības pārvaldības platformas ieviešana RTU, aprakstītas platformas sastāvdaļas, kā arī sniegts ieskats platformas darbības specifikā, tai skaitā piedāvāta arhitektūras detalizācija RTU lietošanas gadījumiem.

Ceturtajā nodaļā ir novērtēti dažādi platformā iekļautie moduļi, tādi kā *DGA* modulis, kurš nosaka vai izsauktās mājaslapas nosaukums (domēns) ir tipisks vai mākslīgi ģenerēts. Mākslīgi ģenerētie domēni tiek izmantoti robotu tīklu dalībnieku savstarpējai saziņai. Tāpat ir novērtēts draudu agregācijas modulis, kura pamatā ir dažādu moduļu rezultātu apvienošana ar mērķi identificēt inficētu ierīci un samazināt viltus pozitīvo ziņojumu skaitu. Darbā novērtēts *NFAI NetFlow* tīkla datu analīzes modulis, kurš izmanto apmācītu mašīnmācīšanās algoritmu un, balstoties uz apmācības datiem, nosaka, vai noteikta komunikācija var tikt uzskatīta par ļaunprātīgu vai nē.

Piektajā nodaļā vairākos posmos un izmantojot dažādas pieejas, ir novērtēta *ISMS* platforma, piemēram, gan salīdzinot platformu ar tirgū pieejamiem risinājumiem, gan veicot lietotāju reaģēšanas ātruma mērījumus, kā arī novērtējot atsevišķus platformas moduļus.

Rezultātu un secinājumu nodaļā ir sniegts darba rezultātu, iegūto secinājumu un turpmāko pētījumu izklāsts. Promocijas darbam ir 2 pielikumi. Darba 1.pielikumā ievietots svarīgāko darbā lietoto terminu un saīsinājumu skaidrojums. Darba 2.pielikumā ietverts piedāvātās sistēmas moduļu avota kods.

2 LITERATŪRAS APSKATS UN IESPĒJAMIE PROBLĒMAS RISINĀJUMI

Sistemātisks literatūras apskats tika veikts saskaņā ar [33] definētajiem strukturēta literatūras apskata veikšanas principiem un ir sadalīts trīs posmos – literatūras apskata plānošanā, veikšanā un analīzē. Sistemātiska literatūras apskata priekšrocības ir eksistējošo risinājumu apzināšana, izvairīšanās no kļūdainu pieņēmumu izdarīšanas, zināšanu plaisu (no angļu val. *knowledge gap*) apzināšana un eksistējošo pētījumu izaicinājumu formulēšana. Literatūras apskats izpēta un apkopo informāciju par esošo situāciju IS drošības jomā, ko tālāk paredzēts izmantot spējorientēta IS drošības pārvaldības modeļa un tajā ietverto tehnoloģisko risinājumu izstrādei. Literatūras apskatā ietvertie izpētes jautājumi ir apkopoti 2.1.tabulā.

2.1.tabula

Literatūrā apskatīto jautājumu pārskats

ID	Izpētes jautājums	Jautājuma mērķis	Sagaidāmais rezultāts
RQ1	Kādi ir tipiskie informācijas sistēmu drošības pārvaldībā īstenojamie procesi?	Identificēt procesus, kas veido IS drošības pārvaldību	IS drošības pārvaldības procesu raksturojums
RQ2	Kādi datu avoti mūsdienās tiek lietoti informācijas sistēmu (IS) drošības nodrošināšanai?	Identificēt datu avotus, kas tiek lietoti drošības analīzei, kā arī identificēt metodes, kas tiek lietotas šo datu apstrādei	Datu avotu, kas ir piemēroti sistēmas drošības analīzei, saraksts, kā arī datu avotu apstrādes metožu pārskats
RQ3	Kādas automatizētas metodes un rīki tiek lietoti IS drošības pārvaldības nodrošināšanai?	Identificēt, kādas automatizēšanas metodes un rīki mūsdienās tiek lietoti IS drošības pārvaldības nodrošināšanai	Automatizācijas metožu un to lietojumu pārskats
RQ4	Kādas mašīnmācīšanās metodes tiek lietotas IS drošības pārvaldības nodrošināšanai?	Identificēt, kādas mašīnmācīšanās metodes un rīki tiek lietoti, lai identificētu vēl nezināmus draudus	Dažādu mašīnmācīšanās metožu un rīku, kas tiek lietoti nezināmu draudu identificēšanai, pārskats
RQ5	Kādi ir iespējamie risinājumi, lai nodrošinātu automatizētu ļaunprātīgas aktivitātes apturēšanu tīklā?	Identificēt iespējamos risinājumus, lai nodrošinātu automatizētu ļaunprātīgas aktivitātes apturēšanu tīklā	Risinājumu, kurus izmantojot, ir iespējams īsā laikā apturēt ļaunprātīgu aktivitāti tīklā, apraksts

2.1 RQ1: IS DROŠĪBAS PĀRVALDĪBAS PROCESI

Informācijas sistēmu drošības jomā ir izstrādāti vairāki ietvari [22] [34] [35] [36] [37] [38], kas rekomendē informācijas sistēmu drošības pārvaldībā īstenojamos procesus. Minētie ietvari

papildina viens otru, un galvenokārt tiek orientēti uz maksimālu sakārtotības pakāpi drošības pārvaldības jomā. Promocijas darbā tiek izmantoti brīvi pieejamajā NIST standartā esošie pamatelementi: Identifikācija, Aizsardzība, Atklāšana, Reakcija, Atjaunošanās. Promocijas darba ietvaros galvenā uzmanība tika pievērsta Atklāšanas un Reakcijas funkcijām izstrādājot *ISMS* platformu, kas saskan ar Latvijas *CERT* [39], definējot, ka ka ir ārkārtīgi svarīgi nodrošināt tīkla inventarizāciju un redzamību, operētājsistēmu un izmantotās programmatūras savlaicīgus atjauninājumus, sistēmas drošības notikumu apkopošanu un uzraudzību, kā arī reaģēšanu uz incidentiem.

2.1.1 Atklāšanas process

Atklāšana ir atbilstošu pasākumu īstenošana, lai identificētu kiberdrošības notikumu. Funkcija ļauj savlaicīgi atklāt kiberdrošības notikumus. Šīs funkcijas rezultātu piemēri ir: anomāliju un notikumu identificēšana; nepārtraukta drošības uzraudzība, kā arī kiberdraudu identificēšanas process.

Atklāšanas process iedalās trīs kategorijās:

1) Anomāliju un notikumu identificēšana – tiek atklāta anomāla aktivitāte un tiek izprasta šīs aktivitātes iespējamā ietekme. Anomāliju identifikāciju saprot ar lietotāju un ierīču normālas uzvedības definēšanu un novirzi no šīs normālās uzvedības. Atklātie notikumi tiek analizēti, lai izprastu uzbrukuma mērķus un to metodes. Notikumu dati tiek apkopoti un korelēti no vairākiem avotiem un sensoriem. Tiek noteikta identificēto notikumu ietekme. Tiek noteikti incidenta reaģēšanas līmeņi.
2) Nepārtraukta drošības uzraudzība – tiek uzraudzīti informācijas sistēmu resursi, galaiekārtas un tīkls, lai identificētu kiberdrošības incidentus un pārbaudītu aizsardzības pasākumu efektivitāti. Tiek uzraudzīta gan fiziskā vide, gan personāls, lai atklātu iespējamos kiberdrošības riskus. Tā ir iespēja konstatēt ļaunprātīgā koda darbību. Arī ārējo pakalpojumu sniedzēju darbība tiek uzraudzīta, lai atklātu iespējamos kiberdrošības riskus. Tiek veikta neautorizēta personāla, savienojumu, ierīču un programmatūras uzraudzība. Tiek veiktas regulāras sistēmu ievainojamību pārbaudes.
3) Kiberdraudu identificēšanas process – process un procedūras, kuras tiek uzturētas, lai nodrošinātu izpratni par anomāliem notikumiem. Lai nodrošinātu atbildību, lomas un pienākumi draudu atklāšanai ir precīzi definēti. Atklāšanas metodes ir atbilstošas prasībām (ārējiem un iekšējiem normatīvajiem aktiem). Atklāšanas process ir pārbaudīts un strādā. Notiek atbilstoša apziņošana par atklātajiem drošības incidentiem. Atklāšanas process tiek nepārtraukti uzlabots.

2.1.2 Reakcijas process

Reakcija ir spēja, izstrādājot un ieviešot atbilstošus pasākumus, nekavējoties rīkoties atklāta kiberdrošības incidenta gadījumā. Funkcijai jāspēj identificēt potenciālā kiberdrošības incidenta ietekmi. Šīs funkcijas rezultātu kategoriju piemēri: reaģēšanas plānošana; komunikācija; analīze; incidenta mazināšana un uzlabojumu ieviešana.

Reakcijas process iedalās piecās kategorijās:

1) Reaģēšanas plānošana – procesi un procedūras tiek izpildītas un atbilstoši uzturētas, lai nodrošinātu nekavējošu reaģēšanu uz atklātajiem kiberdrošības incidentiem. Reaģēšanas plāns tiek pildīts incidenta laikā vai pēc tam, kad incidents ir beidzies.
2) Komunikācija – reaģējot uz incidentu komunikācija notiek ar iekšējām un ārējām ieinteresētām personām. Personāls ir informēts par savu lomu un darbību kārtību gadījumos, kad nepieciešama reaģēšana uz incidentu. Par incidentiem tiek ziņots atbilstoši noteiktajai kārtībai. Informācija personālam tiek sniegta atbilstoši pieņemtajam reaģēšanas plānam. Koordinācija ar augstāko vadību notiek saskaņā ar reaģēšanas plānu. Lai panāktu plašāku izpratni par kiberdrošību, notiek brīvprātīga informācijas apmaiņa ar ārējām ieinteresētajām personām.
3) Analīze – efektīvas reakcijas nodrošināšana un, ja nepieciešams, atjaunošanas darbību uzsākšana. Tiek izmeklēti ziņojumi no draudu atklāšanas sistēmām. Drošības incidenta ietekme ir novērtēta un saprotama. Tiek veikta drošības incidenta izmeklēšana. Incidenti tiek iedalīti kategorijās atbilstoši reaģēšanas plānam. Izstrādāti procesi, lai saņemtu, analizētu un reaģētu uz ievainojamībām, kuras ir atklātas no iekšējiem vai ārējiem avotiem (piemēram, iekšējās testēšanas, drošības pētnieki).
4) Drošības incidenta mazināšana – darbības, kuras tiek veiktas, lai novērstu incidenta paplašināšanos, mazinātu tā sekas, kā arī novērstu pašu incidentu. Drošības incidentu ietekme ir ierobežota un samazināta. Jaunatklātās ievainojamības tiek mazinātas vai arī risks tiek pieņemts.
5) Uzlabojumu ieviešana – reaģēšanas spējas tiek uzlabotas, iekļaujot mācības, kas gūtas no pašreizējiem un iepriekš notikušiem drošības incidentiem un darbībām to novēršanai. Uzlabošanas plāni ietver gūto pieredzi. Reaģēšanas stratēģijas tiek regulāri pārskatītas. [22].

2.1.3 Secinājumi

Veicot dažādu darbā apskatīto ietvaru analīzi, autors ir izvēlējies definīcijas, kuras autoraprāt vislabāk raksturo drošības pārvaldību, tās ir: identifikācija, aizsardzība, atklāšana, reakcija un atjaunošanās kuras visprecīzāk ir aprakstītas NIST ietvarā. Tā kā precīzi sakārtota reakcija organizācijās visbiežāk izpaliek, darbā autors tai pievērsis vislielāko uzmanību.

2.2 RQ2: DATU AVOTU IDENTIFICĒŠANA DROŠĪBAS ANALĪZEI

Datu avoti informācijas sistēmu drošības kontekstā tiek saprasti ar informācijas avotu nodrošināšanu drošības analīzei. Lai gūtu maksimāli pilnīgu ainu informācijas sistēmu drošības jomā ir svarīgi iegūt datus no plaša datu avotu skaita. Katrs datu avots ir noderīgs un ļoti svarīgs kopējās bildes radīšanā. Bieži vien ļaunprātīga koda identificēšanai nepietiek ar antivīrusa programmatūru un ir nepieciešami arī citi datu avoti, tādi kā *IDS*. Saskaņā ar Li et al [40] datu avoti var būt dažādi, ieskaitot dažādus aģentu datus, tīkla datus, SMTP, Syslog un citus. Papildu datiem, kuru avots ir uzņēmuma vai iestādes infrastruktūra, iespējams izmantot dažādus apkopotus melnos sarakstus ar IP adresēm vai domēna vārdiem [41] [42] [43], kuros tiek izvietots ļaunprātīgs saturs, piemēram pikšķerēšanas (angl.*phishing*) lapas. Šāda salīdzināšana ar zināmām ļaunprātīgām adresēm atvieglo analītiķa darbu, kā arī iespēja iegūt viltus pozitīvo ziņojumu nav liela. Diemžēl gan ļaunprātīgās IP adreses, gan domēna vārdi nonāk šādos sarakstos novēloti, tādēļ nepieciešami arī citi risinājumi drošības risku mazināšanai. Turpmāk tiks aplūkoti dažādi datu avoti, kuri var būt noderīgi analītiķa darbā, tādi kā tīkla dati, auditācijas pieraksti un citi.

2.2.1 Tīkla dati

Lai saprastu tīkla datus, nepieciešams izprast tīkla paketes sastāvdaļas (2.1.attēls). IP paketes galvene ir svarīgākā paketes sastāvdaļa, tādēļ, ka norāda uz dažādiem tīkla paketes būtiskākajiem atribūtiem, tādiem kā avota un mērķa IP adreses, avota un mērķa porti, paketes dzīves laiks un citiem.

	Oktets	0								1								2								3							
Oktets	Biti	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Versija				Galvenes garums				Pakalpojuma veids						ECN		Kopējais garums															
4	32	Identifikācija																IP karodziņi			Fragmenta nobīde											
8	64	Dzīvlaiks								Protokols								Galvenes kontrolsumma															
12	96	Avota adrese																															
16	128	Galamērķa adrese																															
20	160	IP opcijas																															
⋮	⋮																																
60	480																																

2.1.att. IP paketes galvene

IP galvenē ir iekļauta būtiska informācija, tostarp versijas numurs, kas šajā kontekstā ir 4. Cita papildinformācija ir šāda:

• versija - versijas nr. izmantotā interneta protokola (piem., IPv4);

• interneta galvenes garums; Visas IP galvenes garums;

• pakalpojuma veids;

• ECN (*Explicit Congestion Notification*) – paziņojums, kas informē par maršrutā identificēto sastrēgumu;

• kopējais garums – visas IP paketes garums (ieskaitot IP galveni un IP nosūtāmos datus);

• identifikācija – ja pārraides laikā IP pakete ir fragmentēta, visiem fragmentiem ir viens un tas pats identifikācijas numurs, lai varētu identificēt oriģinālo IP paketi, kurai šie fragmenti pieder;

• karodziņi – lielas IP datu paketes gadījumā, "karodziņi" norāda arī uz to, ka tā ir sadalīta;

• fragmenta nobīde – šī nobīde norāda precīzu fragmenta atrašanās vietu sākotnējā IP paketē;

• dzīvošanas laiks – lai izvairītos no mūžīgas paketes ceļošanas, katra pakete tiek nosūtīta ar noteiktu dzīvošanas laika (TTL (t*ime to live*)) vērtību, kura norāda, cik maršrutētājus šī pakete var šķērsot. Katrā maršrutētajā šķērsošanas reizē TTL vērtība tiek samazināta par vienu, un, kad vērtība sasniedz nulli, pakete tiek iznīcināta;

• protokols – norāda tīkla slānim galamērķa resursdatorā, kuram protokolam pieder šī pakete, t.i., nākamā līmeņa protokols. Piemēram, ICMP (*Internet Control Message Protocol*) protokola numurs ir 1, *TCP* (*Transmission Control Protocol*) ir 6 un *UDP* (*User Datagram Protocol*) ir 17;

• galvenes kontrolsumma - šis lauks tiek izmantots, lai saglabātu visas galvenes kontrolsummas vērtību, kuru pēc tam izmanto, lai pārbaudītu, vai pakete ir saņemta bez kļūdām;

• avota adrese – paketes sūtītāja (vai avota) 32 bitu adrese;

• galamērķa adrese – paketes saņēmēja (vai adresāta) 32 bitu adrese;

• opcijas – tas ir neobligāts lauks, kurš var ietvert tādu opciju vērtības kā drošība, ierakstu maršruts, laika zīmogs utt.

Pamatojoties uz [44], iespējams iegūt labu datu kopu (2.2.tabula) turpmākai mākslīgā intelekta modeļu apmācībai izmantojot IP paketes galvenes informāciju un metadatu informāciju par datu pakešu plūsmu.

2.2.tabula

Pazīmju kopa mākslīgā intelekta modeļu apmācībai (adaptēts no *[44]*)

Plūsmu skaits, ienākošo plūsmu skaits, izejošo plūsmu skaits
Ienākošo un izejošo plūsmu procentuālā daļa no kopējās
% ar simetrisko un asimetrisko ienākošo plūsmu pār kopējo
IP pakešu summa, maksimālā, minimālā, vidējā un dispersija uz ienākošajām, izejošajām un kopējām plūsmām
Baitu summa, maksimālā, minimālā, vidējā un to dispersija uz ienākošajām, izejošajām un kopējām plūsmām
Avota baitu summa, maksimālā, minimālā, vidējā un to dispersija uz ienākošajām, izejošajām un kopējām plūsmām
Dažādu avotu IP skaits ienākošajām plūsmām un galamērķa IP izejošajām plūsmām
Ienākošo un izejošo plūsmu dažādu avotu un galamērķa portu skaits
Ienākošo plūsmu avotu IP un izejošo plūsmu galamērķa IP entropija
Ienākošo un izejošo plūsmu avota un galamērķa entropija
% no avota un galamērķa portiem> 1024 ienākošajām un izejošajām plūsmām
% no avota un galamērķa portiem ≤ 1024 ienākošajām un izejošajām plūsmām

2.2.2 Ielaušanās noteikšanas sistēmas (IDS) dati

Ielaušanās noteikšanas/novēršanas sistēma ir viena no mūsdienu automatizētām drošības pārvaldības sistēmām, kura bieži tiek pielietota organizāciju drošības pārvaldības nodrošināšanai. Ielaušanās noteikšanas sistēma no ielaušanās novēršanas sistēmas atšķiras tikai ar iespēju reaģēt uz incidentu bloķējot datu plūsmu, tādēļ turpmāk tiks apskatīta tikai ielaušanās noteikšanas sistēma (IDS).

IDS var tikt iedalītas šādās kategorijās [45]:

1) tīkla bāzētas *IDS* (*NIDS*), kuras monitorē datortīklu un reaģē uz tajā atklātajām problēmām;
2) resursdatora bāzētās *IDS* (HIDS), kuras analizē auditācijas datus un dažādus citus notikumus un tiek instalētas katrā auditējamajā darbstacijā;

3) uz protokolu balstīta *IDS* (PIDS) ir ielaušanās noteikšanas sistēma, kas parasti tiek instalēta tīmekļa serverī un tiek izmantota skaitļošanas sistēmas izmantotā protokola uzraudzībā un analīzē. PIDS uzrauga protokola uzvedību un stāvokli, un to veido sistēma vai aģents, kas parasti atrodas uz servera, uzrauga un analizē saziņu starp pievienoto ierīci un sistēmu, kura tiek aizsargāta.
4) Uz OSI lietojumslāni balstīta *IDS* (APIDS) ir ielaušanās noteikšanas sistēma, kuras uzraudzība un analīze ir vērsta uz konkrētu lietojumprogrammu protokolu vai protokoliem, ko izmanto informācijas sistēma. APIDS uzrauga protokola dinamisko uzvedību un stāvokli, un tā parasti sastāv no sistēmas vai aģenta, kas atrodas starp serveru grupu, pārraugot un analizējot lietojumprogrammu protokolu starp divām savienotām ierīcēm. Tipiska vieta APIDS būtu starp tīmekļa serveri un datu bāzes serveri, pārraugot SQL protokolu, kad tas mijiedarbojas ar datu bāzi.
5) hibrīdais *IDS*, kas ir dažādu augstākminēto *IDS* apvienojums.

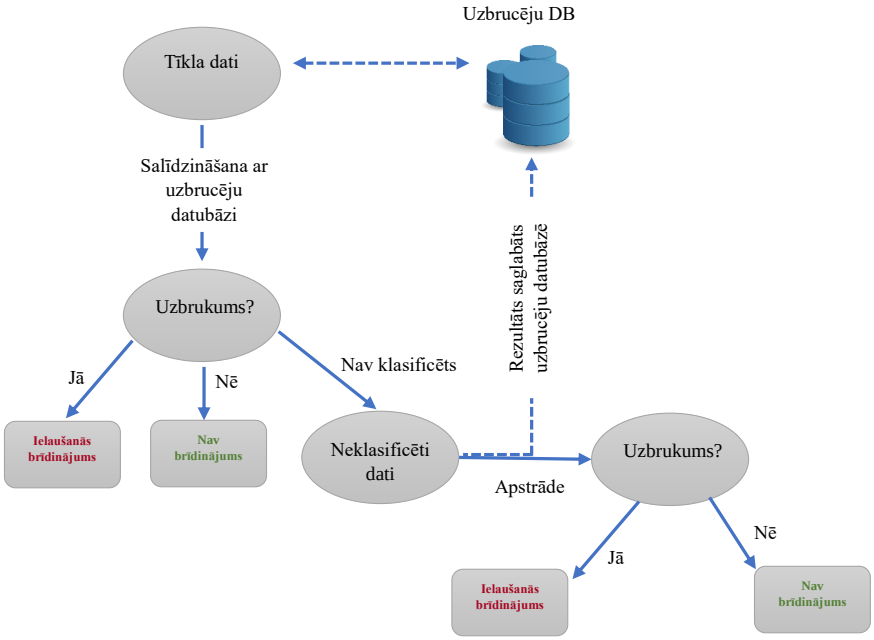
IDS izmanto divas dažādas ļaunprātīgas darbības noteikšanas metodes: uz parakstu balstīta metode, kur ļaunprātīgas darbības noteikšana notiek pamatojoties uz noteiktiem likumiem (signatūrām), kuras izstrādājot tiek izmantota informācija par jau iepriekš atklātu ļaunprātīgu programmatūru; uz anomālijām balstīta metode, kas izmanto mašīnmācīšanos, lai atklātu iepriekš nezināmu, iespējams, ļaunprātīgu darbību.

Pētnieki [46] piedāvā izkliedētu ielaušanās noteikšanas sistēmas modeli (DIDS). Piedāvātais modelis ir balstīts uz autonomu sadarbības aģentu kopumu, kuri spēj atklāt anomālas darbības un draudus, kuru īstenošanās var ietekmēt informācijas konfidencialitāti, integritāti un/vai pieejamību. Pētnieku piedāvātais modelis sastāv no četriem aģentiem, kuri, tīkla uzbrukumu identificēšanai, sazinās viens ar otru. Katrs no šiem aģentiem veic savu uzdevumu autonomā veidā, vienlaikus saglabājot mijiedarbību ar pārējiem aģentiem, kuri iesaistīti DIDS:

1) Uztveršanas aģents: Šī aģenta uzdevumus ir uztvert tīklā cirkulējošos tīkla datus, klausoties tīkla interfeisu.
2) Filtrēšanas un slodzes līdzsvarošanas aģents: šim aģentam ir divas galvenās lomas, no vienas puses, tas piemēro filtrēšanas darbības apstrādājamajiem datiem, pārbaudot iegūto pakešu un ielaušanās datu bāzes atbilstību, nodrošinot neklasificētas datu plūsmas apstrādi (normāli vai neparasti tīkla dati) un ļauj apstrādāt un uzglabāt datus dažādos HDFS (Hadoop Distributed File System) klasteru mezglos.

3) Lēmumu pieņēmēja aģents: Šis aģents ir atbildīgs par lēmuma pieņemšanu pēc Hadoop HDFS saglabātās nekategorizētās tīkla datu analīzes un apstrādes. Tas var klasificēt paketi kā ielaušanos vai kā nekaitīgu darbību, bet, neatkarīgi no rezultāta, tā tiks saglabāta datu bāzē.
4) Aģentu pārvaldnieks: tas ļauj tīkla un drošības administratoriem veikt apziņošanas uzdevumus par visiem konstatētajiem ielaušanās gadījumiem (piemēram: top 10), mijiedarbojoties ar iebrucēju datu bāzi.

Pētnieki tīkla datu uzglabāšanai izmantoja augstas veiktspējas apstrādes un uzglabāšanas sistēmu. Šim nolūkam tika izmantota Hadoop failu sistēma, izmantojot HDFS, un datu apstrādei tiek izmantots MapReduce. Piedāvātā modeļa attēlojums redzams 2.2.attēlā.

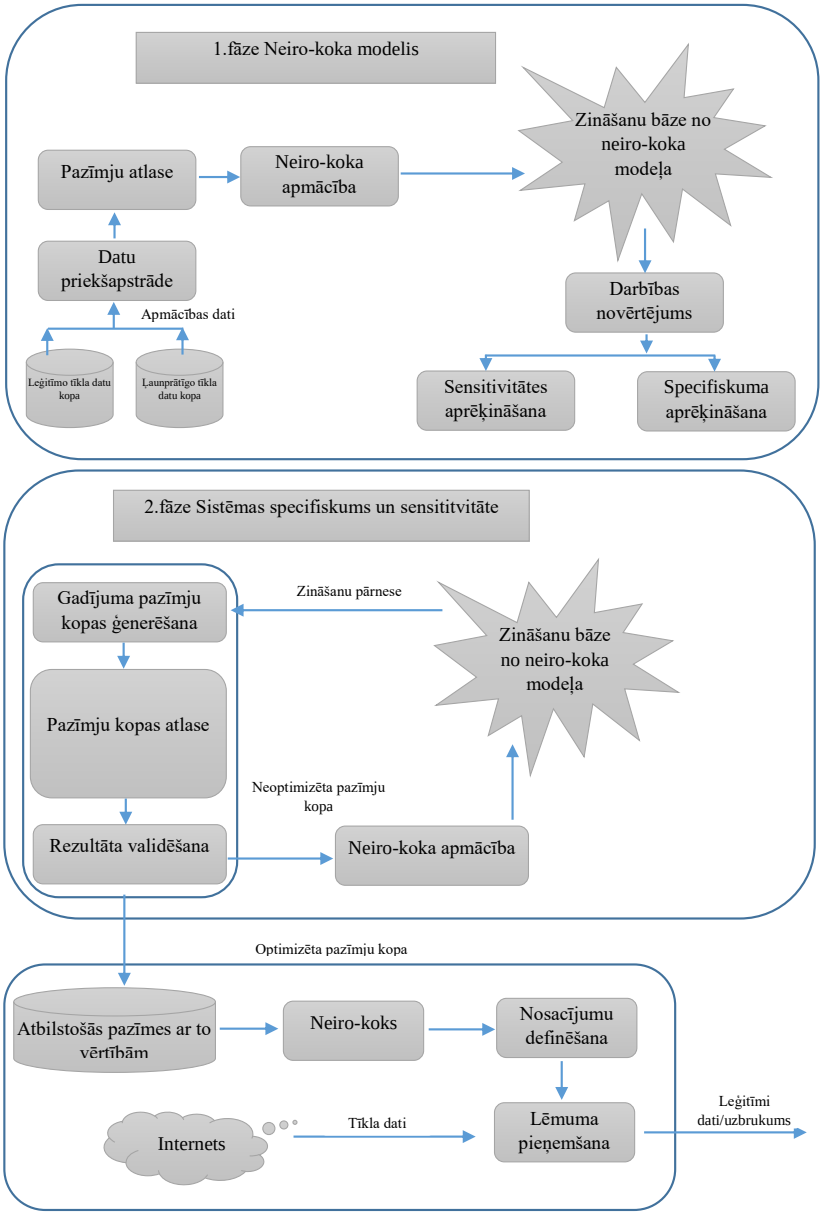


2.2. att. Piedāvātā modeļa attēlojums (adaptēts no *[46]*)

Izmantojot piedāvāto metodi (2.2.attēls), pētnieki apgalvoja, ka ir spējīgi identificēt nulles dienas uzbrukumus (tos uzbrukumus, kuri nav aprakstīti *IDS* signatūru datubāzēs).

Arī Shah et al. [47] piedāvāja uz anomāliju bāzētu *IDS* sistēmas modeli, kurš izmantoja neironu tīklus. Metodes pamatā bija apmācīts algoritms apdraudējumu identificēšanai. Pētnieki apgalvoja, ka, izmantojot viņu piedāvāto metodi *KNN* modeļa precizitāte bija vidēji līdz

91,14%, kur 91% precizitāte identificēt leģitīmus tīkla datus un 92% precizitāte paredzēt tiešus uzbrukumus un 91% precizitāte prognozēt slēptus uzbrukumus. *RFC* modelis sasniedza vidējo precizitāti 92,41%, precīzi prognozējot 100% leģitīmu datu plūsmu, 92% paredzot tiešus uzbrukumus un 100% paredzot slēptus uzbrukumus. *SVM* sasniedza vidēji 19,23% precizitāti, un 0% precizitāti identificējot leģitīmus tīkla datus, ar 100% precizitāti identificējot tiešu uzbrukumu un 82% slēptu uzbrukumus.



2.3.att. Neiro-koka apmācības modelis (adaptēts no *[48]*)

Arī citi pētnieki [48] piedāvāja uzlabot *IDS* sniegumu, izmantojot mašīnmācīšanās algoritmus tādus kā lēmumu koki, neironu tīkli un ģenētiskie algoritmi (2.3.attēls). Pētnieki piedāvāja ļaunatūras identificēšanu sadalīt divās daļās, kā arī optimizēt pazīmju kopu ansamblī. Izmantojot šādu ansambli, autoriem izdevās sasniegt 98.38% algoritma precizitāti.

IDS sistēmas palīdz organizācijām identificēt aizdomīgu darbību tīklā, kura var liecināt par uzbrukumu, kā arī novērst tā īstenošanu. Vienas no visizplatītākajām ir *IDS* ir tīkla bāzētas *IDS* (*NIDS*), kuras ir uz signatūrām balstītas *IDS* (SIDS). Tās salīdzina tīkla datos pakešu parametrus ar ļaunprātīgiem tīkla datu pakešu parakstiem no datu bāzes un ģenerē brīdinājumus, ja tie sakrīt [47]. *IDS* brīdinājumu piemērs parādīts 2.3.tabulā.

2.3.tabula

IDS brīdinājumu piemērs (adaptēts no RTU izmantotā *IDS*)

06/16/2021-18:26:16.569694 [**] [1:2016870:12] ET POLICY Unsupported/Fake Internet Explorer Version MSIE 5. [**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {*TCP*} ??.??.??.??:52465 -> 202.213.202.97:80
06/17/2021-08:05:52.704426 [**] [1:2027412:1] ET POLICY Inbound RDP Connection with TLS Security Protocol Requested [**] [Classification: Potentially Bad Traffic] [Priority: 2] {*TCP*} 71.6.135.131:43472 -> ??.??.??.??:3389
06/17/2021-08:39:33.830214 [**] [1:2500134:5832] ET COMPROMISED Known Compromised or Hostile Host Traffic *TCP* group 68 [**] [Classification: Misc Attack] [Priority: 2] {*TCP*} 23.95.191.212:46675 -> ??.??.??.??:443

Saskaņā ar Sharma et al. [49], *IDS* aktivizē brīdinājumus šādos gadījumos:

1) Kad *IDS* dinamiskā protokola noteikšanas mehānisms identificē aizdomīgu IRC savienojumu (piemēram, IRC datus nestandarta portā).
2) Identificējot skenēšanu (iekšējā, ārējā skenēšana). Kaut arī uzbrukumi sākas ar fāzi, kur skenēšana ir neatņemama sastāvdaļa, lielākā daļa skenēšanas brīdinājumu ir viltus pozitīvi vai labākajā gadījumā nepārliecinoši, un tāpēc lielākā daļa no tiem tiek ignorēti.
3) Identificējot HTTP vai FTP protokolu izmantošanu, kad tiek lejupielādēta zināma ievainojamība no publiski pieejamiem resursiem, kura ir *IDS* sistēmas signatūru datubāzē. Šādu brīdinājumu efektivitāte ir atkarīga no pastāvīgas signatūru atjaunināšanas.

4) Identificējot ļaunprātīgas programmatūras lejupielādi, ņemot vērā pieejamo informāciju par zināmo identificēto ļaunprātīgo programmu izpildāmā koda jaucējfunkcijas interpretāciju, piemēram MD5 un SHA1. Bieži uzbrucēji pārdēvē failu tipus (piemēram, Linux binārie faili tiek lejupielādēti kā JPG faili).

Lai arī vīrusu signatūru izmantošana ir diezgan efektīvs veids kā identificēt ļaunprātīgu kodu, tomēr polimorfie vīrusi ir spējīgi apiet šo tehnoloģiju, mainot savu izpildāmo kodu, tādēļ ir svarīgi izmantot arī anomāliju identificēšanas metodes, kuras identificē, piemēram tādu populāru vīrusu/tārpu kā *blaster* un *nimbda* aktivitāti.

2.2.3 Auditācijas pieraksti

Auditācijas pieraksti ir svarīgs datu avots IS drošības analīzes nodrošināšanai. Auditācijas pierakstus vāc no dažādiem avotiem ar dažādu pieraksta formu, piemēram tie var būt *JSON* vai *syslog* formātā (2.4.tabula). Saskaņā ar MK noteikumiem Nr.442 [10] auditācijas pierakstiem jābūt saglabātiem atsevišķi no sistēmas, kura tos ģenerē, tas ir, nepieciešams tos pārsūtīt uz citu serveri apstrādei un glabāšanai.

2.4.tabula

Dažādas auditācijas pierakstu formas (adaptētas no RTU izmantojamām sistēmām)

2021-06-17 10:25:39"#011Logout|module_instance|Application#011id=rasmus2,ou=agent,dc=openam,dc=forgerock,dc=org#0118ed04095d0b75#011??.??.??.??#011INFO#011dc=openam,dc=forgerock,dc=org#011"cn=dsameuser,ou=DSAME Users,dc=openam,dc=forgerock,dc=org"#011AUTHENTICATION-305#011Application#011"Not Available"#011??.??.??.??#011
{"CreationTime": "2021-06-17T07:00:19","Id":"????75fa-13fa-4cd5-a970","Operation":"UserLoggedIn","OrganizationId":"????3d45-a972-4474-9d53","RecordType":15,"ResultStatus":"Success","UserKey":"dece2a17-281d-4626-bcc6-435d228c????","UserType":0,"Version":1,"Workload":"AzureActiveDirectory","ClientIP":"??.??.??.??","ObjectId":"00000002-0000-0ff1-ce00-000000000000","UserId":"????@rtu.lv","AzureActiveDirectoryEventType":1,"ExtendedProperties":[{"Name":"ResultStatusDetail","Value":"Success"},{"Name":"KeepMeSignedIn","Value":"true"},{"Name":"UserAgent","Value":"Mozilla\/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit\/537.36 (KHTML, like Gecko) Chrome\/91.0.4472.106 Safari\/537.36"},{"Name":"UserAuthenticationMethod","Value":"16"},{"Name":"RequestType","Value":"Login:login"}],"ModifiedProperties":[],"Actor":[{"ID":"dece2a17-281d-4626-bcc6","Type":0},{"ID":"???@rtu.lv","Type":5}],"ActorContextId":"2a543d45-a972-4474-9d53","ActorIpAddress":"??.??.??.??","InterSystemsId":"0371eef2-3fcf-437e9e2b","IntraSystemId":"414375fa-13fa-4cd5-

a970","SupportTicketId":"","Target":[{"ID":"00000002-0000-000000000000","Type":0}],"TargetContextId":"4474-9d53dcf1a97e1623","ApplicationId":"00000002-0000-ce00-000000000000","DeviceProperties":[{"Name":"OS","Value":"Windows 10"},{"Name":"BrowserType","Value":"Chrome"},{"Name":"IsCompliantAndManaged","Value":"False"},{"Name":"SessionId","Value":"4296d0d5-1dac-453b-9c7e"}],"ErrorNumber":"50140"}

Apstrādājot auditācijas pierakstus, saskaņā ar Rose et al. [50] pieejai jābūt šādai:

1) Prioritāte kritisko resursu aizsardzībai – nepieciešams identificēt kritiskos resursus un sensitīvos datus, kuriem nepieciešama paaugstināta aizsardzība.
2) Identificēt interesējošos scenārijus. Uzskaitīt "scenāriju" kolekciju vai iespējamos ļaundaru mērķus (piemēram, sensitīvu datu noplūde). Nepieciešams noteikt darbības scenāriju elementus, tos identificējot auditācijas žurnāla failos, definējot noteiktus pārliecības kritērijus.
3) Sekot līdzi interesējošām darbībām. Izpildīt uz scenārijiem balstītas darbības automatizētā sistēmā.
4) Apkopot un korelēt rezultātus, centralizējot analīzes rezultātus, attēlojot tos identificētā scenārija kontekstā un aprēķinot scenāriju izpildes kopējo varbūtību, lai prioritizētu notikumus un identificētu kritiskus vai nenovēršamus incidentus.

Saskaņā ar [51] sistēmas auditācijas pierakstos (Syslog) iekļauj dažādus datus (2.4.tabula) un tie iedalās dažādos tipos (2.5.tabula):

2.5.tabula

Sistēmu auditācijas pierakstos iekļaujamā informācija (saskaņā ar *RFC*5424) [51]

Numerācijas kods	Apraksts (Angl)	Apraksts
0	*kernel messages*	kodola ziņas
1	*user-level messages*	lietotāja līmeņa ziņojumus
2	*mail system*	pasta sistēma
3	*system daemons*	sistēmas dēmoni
4	*security/authorization messages*	drošības/autorizācijas ziņojumus
5	*messages generated internally by syslogd*	syslogd iekšēji ģenerēti ziņojumi
6	*line printer subsystem*	līniju printera apakšsistēma
7	*network news subsystem*	tīkla ziņu apakšsistēma
8	*UUCP subsystem*	UUCP apakšsistēma
9	*clock daemon*	laika sistēma (dēmons)
10	*security/authorization messages*	drošības/autorizācijas ziņojumi
11	*FTP daemon*	FTP dēmons
12	*NTP subsystem*	NTP apakšsistēma

13	*log audit*	žurnālfaili – auditācijas pieraksti
14	*log alert*	žurnālfaili – brīdinājumi
15	*clock daemon (note 2)*	pulksteņa dēmons (2. piezīme)

Auditācijas pierakstos iekļaujamās informācijas kritiskums ir attēlots 2.6.tabulā.

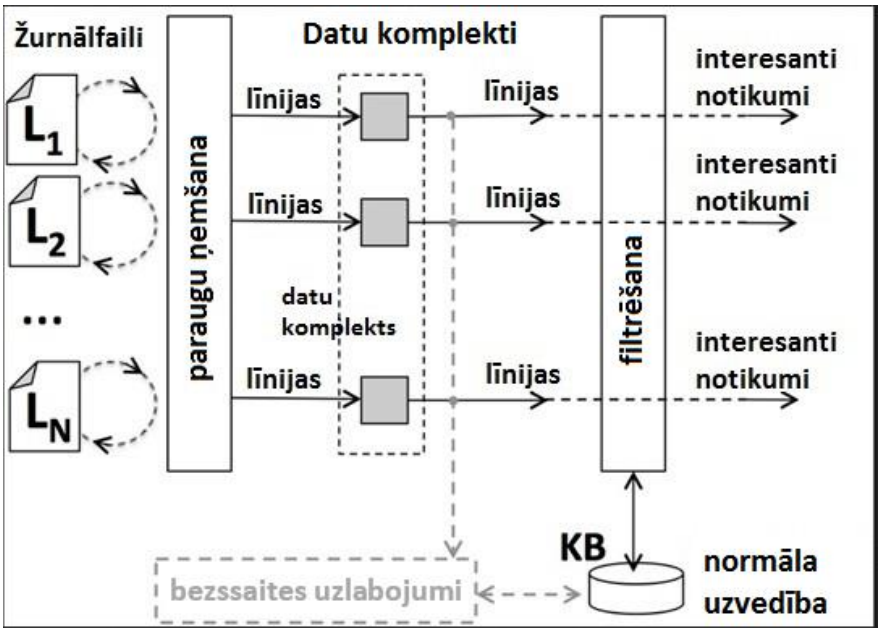
2.6.tabula

Ziņojumu kritiskuma līmenis [51]

Numerācijas kods	**Kritiskuma līmenis**	**Apraksts**
0	*Emergency*	Sistēma ir nestabila
1	*Alert*	Nepieciešama tūlītēja rīcība
2	*Critical*	Kritiska situācija
3	*Error*	Kļūdas situācija
4	*Warning*	Brīdinājums
5	*Notice*	Normālas situācijas ziņojums
6	*Informational*	Informēšanas ziņojumi
7	*Debug*	Atkļūdošanas līmeņa ziņojumi

Sistēmas notikumi ir Syslog notikumu apakšgrupa, kas uzkrāj informāciju, piemēram par aplikācijas kļūdām. Ieraksts satur informāciju par notikuma datumu, laiku un lietotāju, kurš ir veicis noteiktu darbību (piemēram pieteicies sistēmā), ierīces IP adresi, notikuma identifikatoru un citus papildu datus.

Mūsdienās SIEM sistēmas, tādas kā AlienVault [52], IBM QRadar [53], LogRhythm [54] un Splunk Enterprise Security [55] korelē un apstrādā notikumu pierakstus. Viena no notikumu žurnāla analīzes metodēm ir notikumu žurnāla filtrēšana, kas ļauj tikt galā ar lieliem datu apjomiem, identificējot tajos ļaunprātīgus notikumus. Parasti filtrēšanu ietekmē konteksta elementi, kas aptver notikumus noteiktā laika posmā, un, lai noteiktu notikuma prioritāti, tiek apvienoti ar prioritāšu noteikšanas paņēmieniem. Pētnieki [56] piedāvāja filtrēšanas metodi, lai identificētu un noteiktu prioritāti notikumam tekstuālos un neviendabīgos lietojumprogrammu žurnālfailos. Pētnieku piedāvātā metode (2.4.attēls) regulāri reģistrē incidentus, aprēķina kvantitatīvo rādītāju katram notikumam un pamatojoties uz definētajiem kritērijiem, saglabā notikumus, kuriem ir jāpievērš uzmanība. Informācija tiek apstrādāta izmantojot logaritmisko entropiju pa daļām (laika sprīžiem) un tai tiek noteikts kvantitatīvais vērtējums. Lēmums par to vai saglabāt notikumu, ņemto vērā tā kvantitatīvo vērtējumu tiek pieņemts balstoties uz uzkrāto zināšanu bāzi.



2.4.att. Pētnieku piedāvātās metodes apskats (adaptēts no *[56]*)

2.2.4 Tīkla metadati

Tīkla metadati (*NetFlow*) ir tīkla datu statistikas protokols, ko izstrādājusi Cisco. Tā darbības princips ir: izmantojot standarta apmaiņas modeli, *NetFlow* apstrādā datu plūsmas pirmo IP datu paketi veidojot *NetFlow* buferi, vēlāk visi dati tiek apstrādāti vienotā datu plūsmā, izmantojot šo buferi. *NetFlow* buferis satur datu plūsmas statistikas informāciju. Plūsma ir vienvirziena datu pakete, kurai ir tāds pats avota IP un ports, kā arī galamērķa IP un ports. Ņemot vērā atšķirīgās *NetFlow* versijas, pastāv dažādi datu uzkrāšanas veidi. Diezgan plaši izmantotās versijas ir V5 un V8.

Lai arī *NetFlow* dati nesatur pašus datus, tie viennozīmīgi ir vērtīgs avots, lai izprastu situāciju tīklā. Arī mūsdienu šifrētu datu īpatsvars tam nav šķērslis. No *NetFlow* datiem ir iespējams iegūt pazīmju kopu, kuru vēlāk var apstrādāt izmantojot dažādus mašīnmācīšanās algoritmus (2.2.tabula).

Pētnieki [57] piedāvā izmantot *NetFlow*, lai veiktu anomāliju identificēšanu tīklā, izmantojot tīkla *IDS* (*NIDS*):

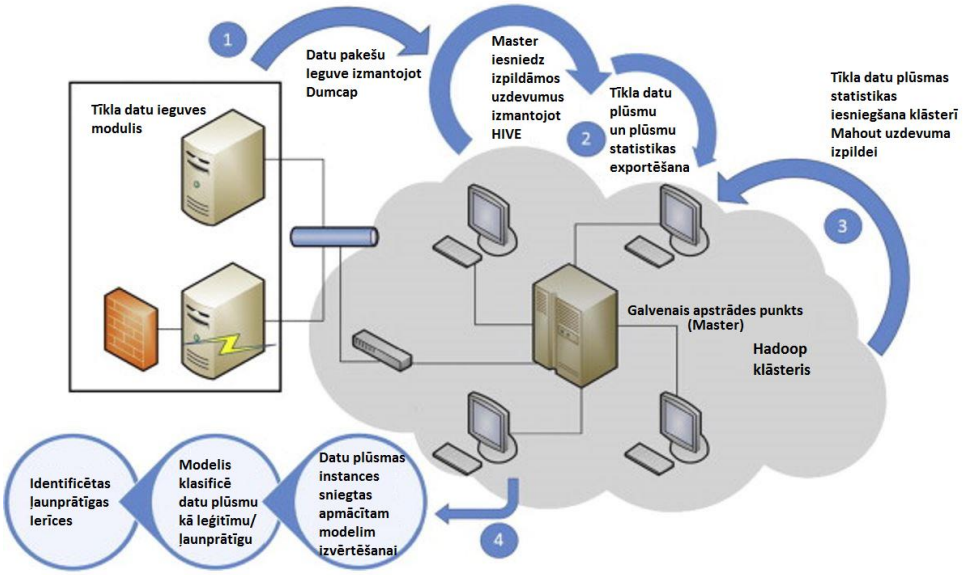
1) *NetFlow* apkopo informācijas plūsmu maršrutētājā. Kamēr IP plūsma tiek nosūtīta izmantojot šo maršrutētāju, tā var tikt analizēta un pārsūtīta. *NetFlow* ir viegli uzstādīt gan pamata maršrutētājā iekšējā tīklā, gan uz ārējo tīklu vērstajā maršrutētājā. Tomēr,

ja ar *NIDS* vēlas analizēt anomālijas visā tīklā, tad ir jāanalizē katrs tīkla segments un šāda veida datu apkopošana ir īpaši sarežģīta.

2) *NIDS* analizē tīkla paketi, lai saprastu datu plūsmu līdz lietojumprogrammas līmenim, tomēr ja lieto *NetFlow*, tad tas nav nepieciešams. Lai gan šādas pieejas rezultātā tiek zaudēta daļa informācijas, tomēr tas ievērojami samazina datu apstrādei nepieciešamo laiku. Augstas caurlaides tīklos *NetFlow* izmantošanas priekšrocības ir acīmredzamas.
3) *NetFlow* izmantošanas gadījumā anomāliju analīzei nav nepieciešama milzīga un pastāvīgi augoša signatūru datubāze, jo tiek izmantota sniegtā plūsmas informācija, nosakot normālas uzvedības atskaites punktu un, pēc tam identificējot, vai analizētā uzvedība novirzās no noteiktā bāzes punkta. Šādi uzbūvēts *IDS* ir spējīgs atklāt vēl nezināmus draudus.

Lai iegūtu datus var tikt izmantots *fprobe* [58] rīks, kurš ievāc tīkla datus no tīkla kartes un izveido *NetFlow* datu struktūru ar *nfcapd* rīku, kurš ir nfdump rīka sastāvdaļa [59].

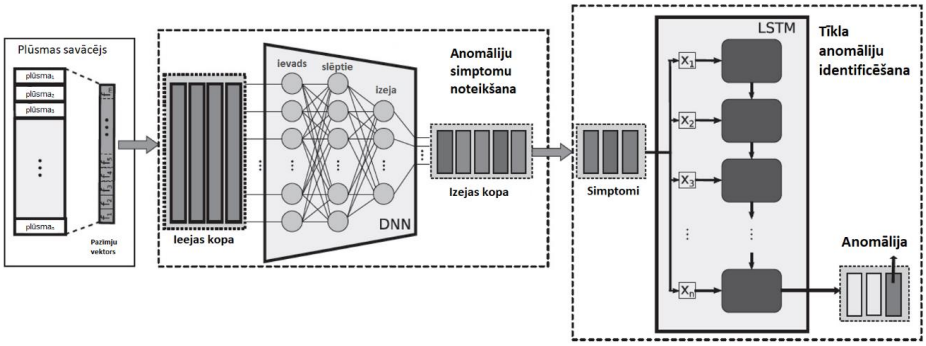
Pētnieki [60] piedāvāja robotu tīklu identificēšanai izmantot tīkla metadatus, tos iegūstot no tīkla interfeisa ar Dumpcap [61] un Tshark [62]. Iegūtajiem datiem tika eksportētas pazīmes izmantojot *Perl* [63] valodas skriptus. Autoru piedāvātā robotu tīklu identificēšanas arhitektūra ir parādīta 2.5.attēlā.



2.5.att. Robotu tīklu identificēšanās arhitektūra (adaptēts no *[60]*)

Autori [60] pielietoja dažādus mašīnmācīšanās algoritmus, tādus kā lēmumu koki, k-tuvākie kaimiņi, neironu tīkli un *SVM*. Autori atzīmēja, ka lieldimensiju apmācības datu kopa padara neefektīvus neironu tīklu un k-tuvāko kaimiņu mašīnmācīšanās algoritmus. Pētnieki piedāvāja izmantot lēmumu kokus, kuri apvienoti ansambļos, jeb *Random Forest* klasifikatoru (*RFC*). Izmantojot šo pieeju, autoriem bija izdevies iegūt 99.8% precizitāti robotu tīkla identificēšanā. Pētījumā tika piedāvāta analīze, kas ir tuvu reālam laikam 5-30 sekunžu intervālos. Viena no izteiktajām Autoru atziņām ir, ka diemžēl robotu tīkli, kuri tīklā neuzvedās "skaļi" bieži vien netiek pamanīti.

Arī citi pētnieki [64] piedāvāja izmantot *NetFlow*, lai identificētu anomālas ļaunprātīgas darbības tīklā. Šim mērķim tika piedāvāts izmantot neironu tīklus (2.3.attēls).



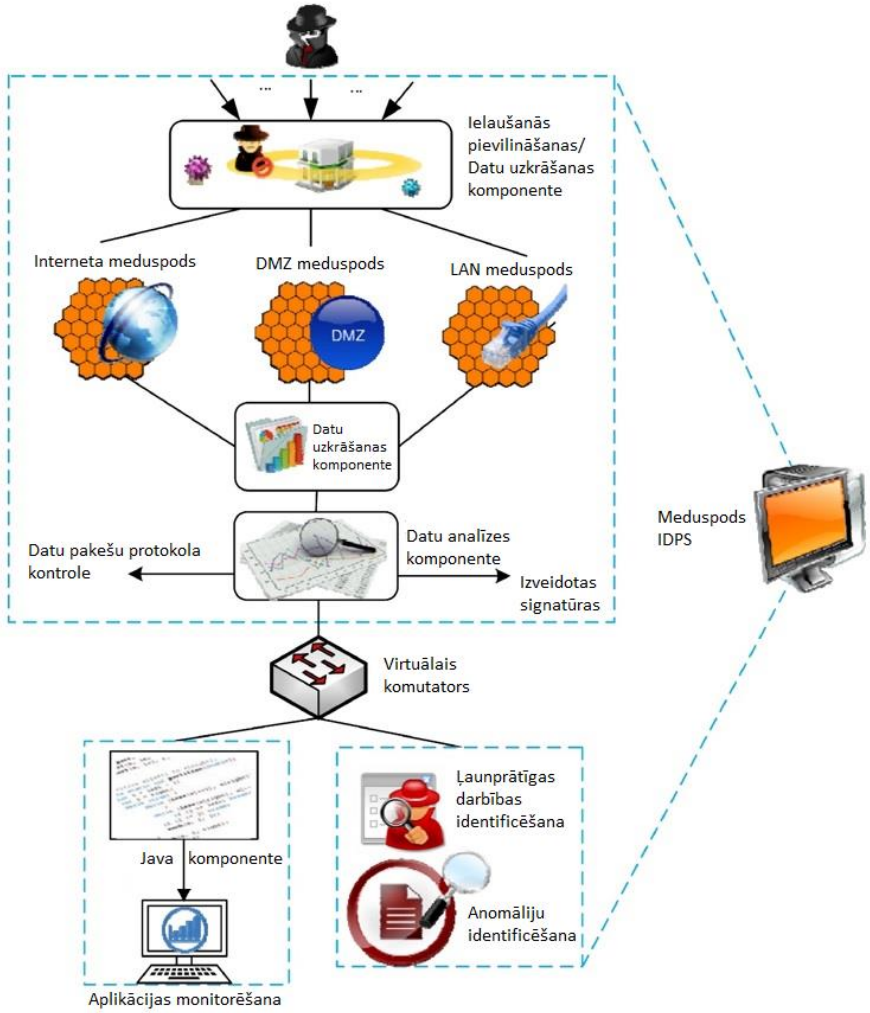
2.6.att. Anomāliju identificēšana tīklā (adaptēts no *[64]*)

Izmantojot pētnieku piedāvāto pieeju, kā arī pētniekiem eksperimentējot ar slēptajiem neironu tīkla slāņiem tika sasniegta precizitāte: 95.37%, pārklājums: 99.54%, un F1-mērs: 97.47%.

2.2.5 "Meduspoda" funkcionalitāte

"Meduspods" (*honeypot*) ir programmatūras kopums, kura uzdevums ir pievilināt uzbrucējus, atstājot pieejamus dažādus portus, kā arī imitējot ievainojamu sistēmu atbildes. Izmantojot "meduspodu" ir iespējams noteikt uzbrukuma metodes un identificēt vēl nezināmus uzbrukumu vektorus. Pats "meduspods" nav drošības risinājums, bet tas var būt drošības risinājuma sastāvdaļa. Medus podus var klasificēt pēc to interaktivitātes līmeņa. Zemas un vidējas mijiedarbības "meduspodi" piesaista iebrucējus, imitējot pakalpojumus, kuriem ir drošības ievainojamība (to var īstenot, izmantojot virtualizācijas tehnoloģijas). Augstas mijiedarbības "meduspodi" piedāvā dažādus papildu pakalpojumus, tai skaitā iespēju uzbrukt uzbrucējam. Tas ļauj iegūt vairāk informācijas par uzbrucēju, bet arī paša "meduspoda"

apdraudēšanas risks ir lielāks kā zemas interaktivitātes "meduspodiem". Pētnieki [65] piedāvā izmantot *IDS* modeli ar "meduspoda" funkcionalitāti, lai identificētu uzbrukumus tīklā (2.7.attēls), kurš kas sastāv no trim "meduspoda" servera moduļiem, kuri mijiedarbojas starp *IDS* un uzraudzības moduli.



2.7.att. *IDS* modelis ar "meduspoda" funkcionalitāti (adaptēts no *[65]*)

Visi moduļi ir atdalīti un darbojas fiziskā mašīnā, lai samazinātu risku, ka iebrucējs apdraud "meduspodu". "Meduspoda" servera lietojumprogramma sastāv no šādām komponentēm: uzbrukuma piesaistes komponente (zemas mijiedarbības "meduspods", kurš piesaista iebrucējus), konfigurācijas komponente (konfigurē *IDS* un modus poda lietojumprogrammu) un *IDS* sakaru komponente (nodrošina iespēju sazināties ar "meduspoda" servera moduli un *IDS* moduli). Zemas mijiedarbības "meduspodi" tīklā var veidot viltotus virtuālos datoru

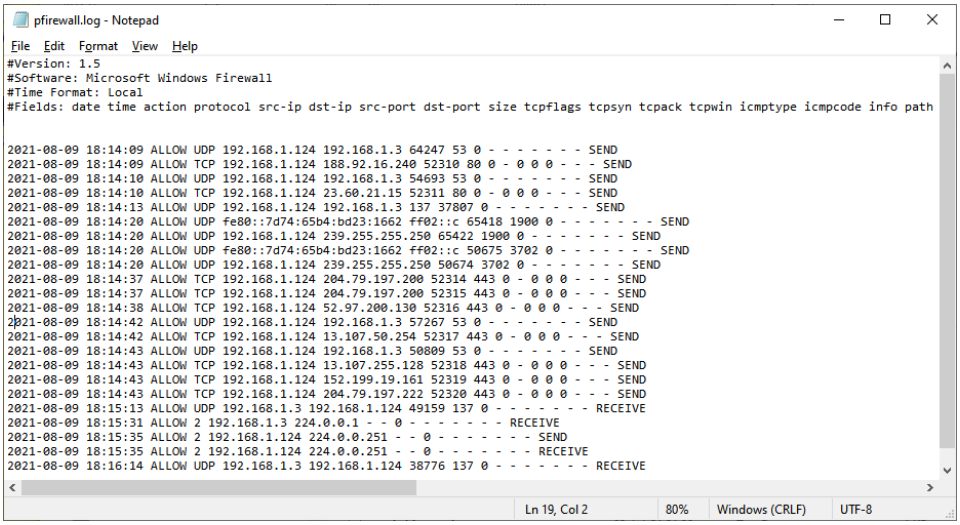
profilus, kuri atbild tikai uz ARP pieprasījumiem. Iebrucējs var uzskatīt, ka "meduspods" ir īsts dators un mēģināt izmantot sistēmas ievainojamības, lai ielauztos sistēmā un pārņemtu kontroli pār to. Lai uzlabotu uzņēmuma tīkla drošību tiek izmantoti žurnāla faili, kuros dati par uzbrucēju var norādīt uz jaunām uzbrukuma metodēm vai nulles dienas uzbrukumiem. Autoru [65] piedāvātajā modelī, *IDS* lietojumprogramma (*Snort IDS*) uzraudzīja "meduspodu" darbību un ļāva konfigurēt "meduspoda" pakalpojumus, izmantojot saskarni (izveidot/atjaunināt/dzēst "meduspodu"). Vienlaikus tā varēja izveidot un izmantot vairākus "meduspodus", kā arī veikt uz protokolu balstītu datu analīzi. Ielaušanās darbības varēja uzraudzīt reāllaikā, izmantojot monitoringa lietojumprogrammu, kas ļāva novērst iespējamās sistēmas kļūdas, pielāgojot konfigurāciju un laicīgi reaģējot uz ļaunprātīgām darbībām. *IDS* parakstu datubāze tika pastāvīgi atjaunināta, izmantojot informāciju, kas ir savākta no izmantotajiem "meduspodiem".

2.2.6 Ugunsmūra dati

Ugunsmūris balstoties uz iebūvētajiem drošības nosacījumiem, aizsargā iekšējo tīklu no ārējā, monitorējot un bloķējot tīkla paketes [66]. Ugunsmūra ģenerētie dati (auditācijas pieraksti) uzrāda kā iebūvētie drošības nosacījumi darbojas un, vai tie darbojas kā paredzēts, dodot iespēju saprast vai ugunsmūris tiek galā ar tīkla datiem, kā arī vai nav identificēta ļaunprātīga darbība un neparedzēti uz āru vērsti savienojumi.

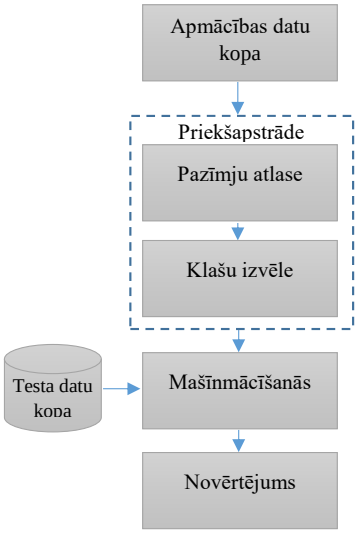
Windows iebūvētā ugunsmūra (2.8.attēls) auditācijas pierakstos ir iekļauta informācija par:

1) savienojuma laiku un datumu;
2) to, kas notika ar savienojumu – vai ugunsmūris ir atļāvis savienojumu vai to liedzis.
3) savienojuma veidu – *TCP*, *UDP* vai cits.
4) savienojuma avota un galamērķa IP adresi, kā arī izmantotajiem portiem savienojumu izveidošanai. Šo informāciju var izmantot, lai identificētu tos portus, kuri ir jāatver, lai specifiskā programmatūra darbotos.
5) to vai darbstacija saņēma datu paketi vai nosūtīja to.



2.8.att. *Windows* ugunsmūra auditācijas pieraksti

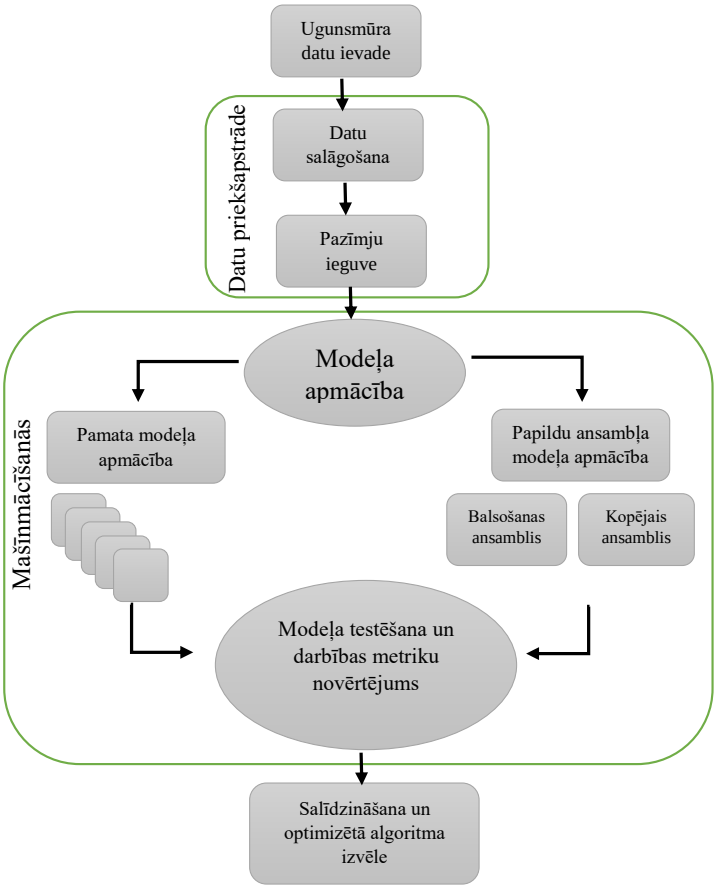
Ugunsmūra auditācijas pierakstu klasificēšana ir viens no veidiem kā tos ir iespējams apstrādāt. Saskaņā ar [67], ugunsmūra un virtuālā privātā tīkla auditācijas pierakstu dati var tikt izmantoti, lai identificētu uzbrukumus datortīklam un informācijas sistēmām. Lai apmācītu izvēlētos (*Naïve Bayes (NiB), KNN, One R un J48*) klasifikācijas algoritmus, pētnieki no auditācijas pierakstiem eksportēja tikai 6 pazīmes: ugunsmūra darbība (atļaut, aizliegt), avota IP adrese, avota ports, galamērķa IP adrese, galamērķa ports un paketes protokols (*TCP*/*UDP*). Piedāvātais modelis paredzēja iegūt apmācību datu kopu, to apstrādājot, apmācot algoritmu un novērtējot algoritma precizitāti (2.9.attēls).



2.9. att. Piedāvātā modeļa blokshēma (adaptēts no *[67]*)

Starp pētnieku pielietotajiem klasifikatoriem, *KNN* klasifikācijas algoritms ir bijis visprecīzākais.

Cits pētījums [68] piedāvāja izmantot heterogēnu balsošanas ansambļa tehniku, izmantojot piecus klasifikatorus (kNN, regresijas metodi, *SVM*, lēmumu kokus un gadījuma mežus, lai apkopotu balsošanas rezultātus un klasificētu ugunsmūra auditācijas pierakstus. Modelis sastāvēja no diviem galvenajiem soļiem (2.10.att.): datu priekšapstrādi un mašīnmācīšanos. Datu priekšapstrādes rezultātā ugunsmūra darbības (akceptēt, liegt, atteikt un nosūtīt *TCP reset*) tika pārveidotas skaitliskā izteiksmē un papildinātas ar tādiem parametriem kā *NAT* avota un galamērķa ports, nosūtīto un saņemto datu apjoms, patērētais laiks, nosūtīto un saņemto datu pakešu skaits. Saskaņā ar pētnieku datiem, kopējais ansamblis sasniedza 85% F1-mēru, 91% precizitāti un 82% pārklājumu.



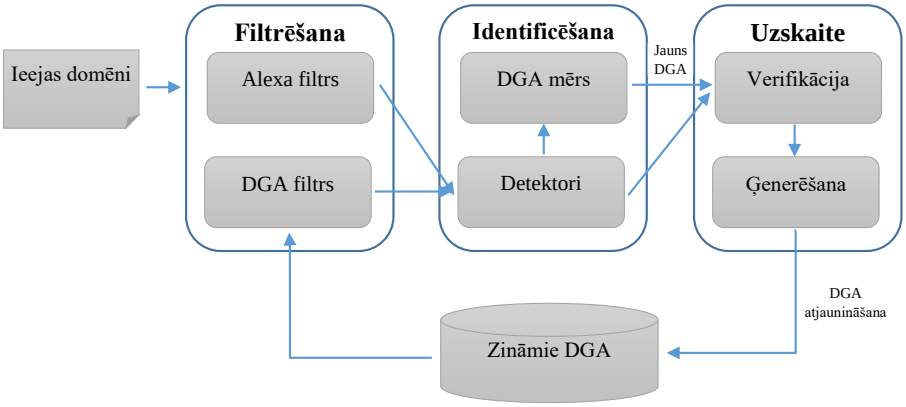
2.10.att. Ansambļa modelis ugunsmūra auditācijas pierakstu klasifikācijai (adaptēts no *[68]*)

2.2.7 DNS informācija

DNS auditācijas pieraksti satur informāciju par avota un galamērķa IP adresēm, to portiem, kā arī izsaukto DNS vārdu. Šie dati var tikt izmantoti, lai ar lielu varbūtību varētu identificēt ierīci, kura ir robotu tīkla sastāvdaļa, identificējot netipisku DNS pieprasījumu, kuru nebūtu ievadījis cilvēks. Mūsdienās, lai robotu tīkla īpašnieks varētu komunicēt ar inficētajiem botiem, tiek izmantoti domēnu ģenerēšanas algoritmi (*DGA*). *DGA* ir identificēti dažādās ļaunprātīgu programmu grupās un tiek izmantoti, lai periodiski ģenerētu lielu skaitu domēnu vārdu, kurus var izmantot kā piekļuves punktus ar viņu komandu un vadības serveriem. Lielais potenciālo piekļuves punktu skaits apgrūtina tiesībaizsardzības iestāžu efektīvu darbu cīņā ar robottīkliem, jo inficētie datori katru dienu mēģina sazināties izmantojot unikālus domēna vārdus, ar mērķi saņemt atjauninājumus vai komandas. Publiskas atslēgas kriptogrāfijas izmantošana ļaunprātīgas programmatūras kodā padara neiespējamu tiesībsargājošajām iestādēm identificēt

nosūtīto komandu, kuru robotu tīklā esošie datori saņem no komandcetriem. *DGA* domēnu ģenerēšanai var izmantot arī vārdus no vārdnīcas. Šīs vārdnīcas var būt iekodētas ļaunprātīgā programmatūrā vai ņemtas no publiski pieejama avota [69]. *DGA*, kuros tiek izmantotas vārdnīcas parasti ir grūtāk noteikt, jo tie ir līdzīgi bieži izmantotajiem domēniem. Ir dažādi veidi, kā var noteikt, vai domēns ir algoritmiski ģenerēts vai cilvēka izveidots. Viens no vienkāršākajiem veidiem ir izmantot melnos sarakstus, kurus piedāvā dažādas organizācijas, piemēram [70].

Daudzi darbā apskatītie pētnieki [71] [69] [72] [73] bija pievērsušies *DGA* identificēšanai un piedāvāja dažādus veidus, kā identificēt algoritmiski veidotus domēnus, piemēram 2.11.att..



2.11.att. Ļaunprātīgi izmantojamo domēnu identificēšana (adaptēts no *[69]*)

Viena no autoru visbiežāk pieminētām metodēm ir n-garms izmantošana, kur tiek novērtēta sakarība starp domēna nosaukumā esošām zīmēm, piemēram sakarības starp 2 zīmēm vai starp 3 zīmēm. Šīm sakarībām tiek noteikta vērtība un ar noteiktajām vērtībām tiek apmācīts mašīnmācīšanās algoritms. Apskatītajā literatūrā pētnieki pārsvarā izmantoja ekspertu sistēmas, kur mašīnmācīšanās algoritms tika apmācīts izmantojot "labo" un "ļauno" domēnu paraugus.

Mēģinot identificēt *DGA* DNS, pētnieki [74] [73] [75] savos darbos ir definējuši svarīgākās pazīmes, kuras var tikt izmantotas algoritmu apmācīšanai, kā arī to, ka visu iespējamo pazīmju izmantošana var padarīt rezultātu sliktāku – t.i. mašīnmācīšanās algoritmi biežāk pieļauj kļūdas. Mowbray et al. [76] piedāvāja pētīt otrā līmeņa domēna nosaukumu, kur otrā līmeņa virkņu garumu sadalījums domēna nosaukumā bija, viņuprāt, labs rādītājs tam, ka tas ir identificējams kā *DGA* ģenerēts domēna vārds. Piedāvātais algoritms identificēja domēnus ar neparastu garumu sadalījumu, tomēr tas varēja atklāt tikai *DGA*, kuros tika izmantoti otrā līmeņa domēni.

Gadījumos, kad ir konstatēti vēl nezināmi *DGA* domēni autori tos izmantoja, lai veiktu klasifikatoru pārapmācīšanu.

Truong et al. [72] ierosināja izmantot lēmumu kokus, lai apmācītu *DGA* klasifikatoru. Pētnieku izmantotais algoritms uzrādīja 92,3% precizitāti. Pētnieki ierosināja izmantot domēna garumu kā vienu no parametriem *DGA* noteikšanai. Autori ir secinājuši, ka *DGA* noteikšanu ar mašīnmācīšanās algoritmiem nevajadzētu izmantot kā vienīgo risinājumu robottīklu noteikšanai, jo jaunās paaudzes robottīkli mēdz mainīt domēnus un izmantot nosaukumus, kas ir līdzīgi bieži lietojamiem domēna vārdiem.

Pētījumā, ko izstrādājuši Ahluwalia et al. [71] garuma loma *DGA* domēna noteikšanai bija ļoti nozīmīga un ietekmēja noteikšanas precizitāti. Pētnieki izmantoja lēmumu kokus, un *DGA* domēna vārda noteikšanas modeļa precizitāte sešzīmju domēna vārdiem bija 85,15%, bet 10 zīmju gariem domēna vārdiem tā sasniedza 94%. Autori izmantoja "n-gram", lai izveidotu pazīmes un eksperimentu rezultāti parādīja, ka "*random forest*" klasifikatoram ir nedaudz labāki rezultāti *DGA* ģenerētā domēna vārda noteikšanā salīdzinājumā ar citiem klasifikatoriem.

Selvi et al. [73] veica 32.tūkst. ļaunprātīgu domēnu analīzi un piedāvāja izmantot leksikas pazīmes, kas bastītas uz "n-gram". Pētnieki identificēja 18 pazīmes un uzskatīja, ka leksikas un statisku pazīmju kombinācija sniedz vislabākos rezultātus. Veiktie pētnieku eksperimenti uzskatāmi parādīja, ka svarīgākās pazīmes ir "unigram", "bigram" un "trigram", to standartnovirze, kā arī patskaņu un līdzskaņu sakarība ar domēna vārda garumu. Tika izpētīti trīs klasifikatori, k-tuvāko kaimiņu, lēmumu koki, lēmumu meži, kur pēdējais klasifikators sniedza vislabākos rezultātus.

J. Peck et al. [77] uzskatīja, ka mašīnmācīšanās algoritmi, lai noteiktu *DGA* domēnu būtu jākombinē ar citiem identificēšanas rīkiem, jo mūsdienās ir pieejams liels DNS maskēšanas veidu skaits, piemēram (*DNS over TLS)* vai (*DNS over HTTPS)* [78] [79].

Lielākā daļa no darbā apskatītajiem pētījumiem balstās uz novecojušiem publiski pieejamiem gan "labiem", gan "sliktiem" domēna vārdiem, tādēļ autors uzskata, ka reālā situācijā tie varētu uzrādīt sliktus rezultātus. Autoraprāt nepieciešams uzkrāt datubāzi ar "sliktajiem" domēniem izmantojot reālus DNS pieprasījumus, piemēram apstrādājot esošus pieprasījumus un salīdzinot ar dažādām publiski pieejamām datubāzēm, kā Quad9 [80] un VirusTotal [42]. Šim mērķim var tikt izstrādāti specifiski mikroservisi.

2.2.8 Secinājumi

Daudzi apskatītie pētnieki uzsvēra, ka kvalitatīvas drošības analīzes veikšanai ir svarīga dažādu datu avotu (auditācijas pierakstu) pieejamība. Bez tam, ir nepieciešama informācija par tīkla datu plūsmu, dati no *IDS* sistēmas, ugunsmūra dati, informācija par ievainojamību noteikšanas sistēmu pārbaužu rezultātiem, ļaunprātīgā koda noteikšanas sistēmu datiem, kā arī cita informācija. Minētie auditācijas pieraksti, ja tiek izmantoti atsevišķi, var nesniegt pilnīgu priekštatu par to, kas ir noticis, piemēram, atsevišķi ielaušanās noteikšanas sistēmas brīdinājumi var radīt viltus pozitīvos brīdinājumus, uz kuriem netiktu tērēts laiks. Labākus rezultātus var sasniegt gadījumos, kad ir pieejama papildu informācija par brīdinājumu no citiem datu avotiem, piemēram, ugunsmūra. Katra sistēma ģenerē savus auditācijas pierakstus, tādēļ ir svarīgi spēt tos pārveidot tā, lai būtu iespējams veikt daudzdimensionālu ar IS drošības pārvaldību saistīto datu analīzi. Veicot literatūras analīzi var secināt, ka pilnvērtīgas drošības analīzes nodrošināšanai ir nepieciešami:

1) *NetFlow* dati, kurus, iespējams iegūt izmantojot dažādus atvērtā koda rīkus, piemēram, nfdump [59]. Šie dati ļauj identificēt tīkla komunikācju, kura atšķiras no "normālas" uzvedības tīklā šādi identificējot, piemēram, ļaunprātīgā koda darbību;
2) sistēmu auditācijas pieraksti, izmantojot kurus ir iespējams identificēt netipisku ierīces uzvedību, identificējot ļaunprātīgu darbību;
3) "meduspoda" funkcionalitāte, kura, pievilinot uzbrucējus ļauj saprast uzbrukuma metodes un iespējamos rīkus, kā arī sniedz papildu laiku reālo informācijas sistēmu aizsardzībai gadījumos, kad uzbrucējs jau ir iekļuvis iekšējā datortīklā;
4) ugunsmūra dati (auditācijas pieraksti), kuros tiek uzkrāta informācija par ieejošajām un izejošajām datu plūsmām. Šī informācija var tikt izmantota, lai identificētu vai ugunsmūra likumi strādā pareizi un, vai nav identificēta ļaunprātīga aktivitāte tīklā;
5) DNS dati, kuri satur informāciju par avota un galamērķa IP adresi un portu, kā arī informāciju par pieprasīto DNS vārdu. Domēna vārdu sintakse var tikt analizēta, lai identificētu algoritmiski ģenerētus domēna vārdu pieprasījumus, kas, savukārt, var norādīt uz ierīces atrašanos robotu tīklā.

Darbā definētais no konteksta atkarīgais, adaptīvais drošības pārvaldības modelis paredz minēto datu avotu izmantošanu. Platformas tehniskā realizācija demonstrē šo datu izmantošanu un ir paplašināma ar papildmoduļiem papildu funkcionalitātes nodrošināšanai.

2.3 RQ3: MŪSDIENU AUTOMATIZĀCIJAS METODES IS DROŠĪBAS PĀRVALDĪBĀ

Šajā nodaļā ir apskatītas mūsdienu automatizācijas metodes, kuras tiek pielietotas organizācijās, lai samazinātu kiberdraudu risku līmeni. Lai nodrošinātu pieņemamu risku līmeni, daudzi ievērojami uzņēmumi, valdības organizācijas un militārās vienības ir ieguldījušas nopietnus līdzekļus, izveidojot savas kiberaizsardzības sistēmas un veidojot sistēmu drošības operāciju centrus (*SOC*), lai veiktu 24×7 uzraudzību, ielaušanās atklāšanu un diagnostiku. *SOC* parasti izmanto dažādus automatizētos drošības pasākumus datoru tīkla uzraudzībai, ugunsmūru uzraudzībai, ievainojamību skenēšanai un ielaušanās noteikšanas/novēršanas sistēmu (IDS/IPS) darbināšanai. *SOC* bieži vien paļaujas uz kiberdrošības analītiķiem, kuri pēta drošības ziņojumus, lai identificētu patieso viņu ietekmi un atbildētu uz tādiem jautājumiem kā, piemēram, vai tīklam šobrīd uzbrūk, ko dara uzbrucēji un kādi varētu būt uzbrucēju turpmākie soļi.

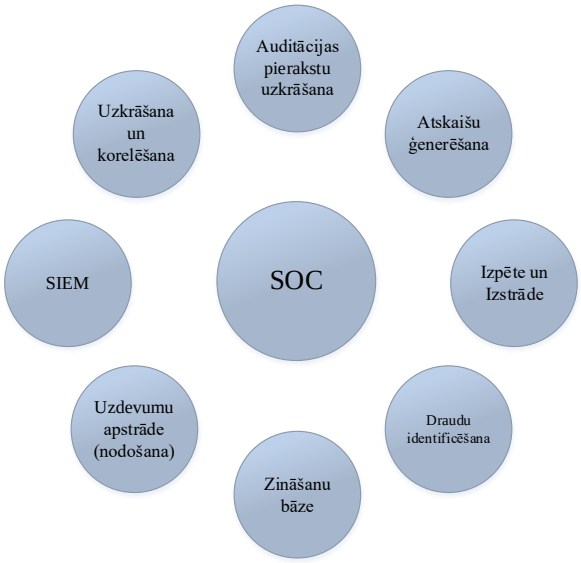
Pamatojoties uz drošības kompāniju Macfee [81], *SOC* var definēt šādi:

1) *SOC* identificē dažādas ierīces, procesus un lietojumprogrammas, kā arī palīdz nodrošināt to aizsardzību.
2) *SOC* īsteno preventīvus pasākumus, kurus var iedalīt divās galvenajās kategorijās:
    a. informētība par jaunumiem drošības jomā, kā arī kibernoziegumu jaunākajām tendencēm un jaunu draudu attīstību. Gatavošanās dažādiem uzbrukumu scenārijiem, kas var palīdzēt izveidot drošības ceļvedi, lai sniegtu norādes uzņēmuma turpmākajiem kiberdrošības attīstības scenārijiem, un nepārtrauktās darbības un atjaunošanas plānu izstrādei un kalpotu kā norādījumi katastrofas gadījumā.
    b. profilaktiskie darbi, kas ietver visas darbības, kuras ir veiktas, lai apgrūtinātu veiksmīgus uzbrukumus, ieskaitot regulāru esošo sistēmu uzturēšanu un atjaunināšanu; ugunsmūra likumu pārskatīšanu un atjaunošanu; ievainojamību labošanu; balto un melno sarakstu izveidi un to uzturēšanu.
3) Nepārtraukta proaktīva uzraudzība un rīki nepārtrauktai tīkla skanēšanai, lai identificētu jebkādas novirzes vai aizdomīgas darbības. Tīkla uzraudzība visu diennakti ļauj *SOC* nekavējoties paziņot par iespējamiem draudiem, sniedzot iespēju novērst vai mazināt kaitējumu. Uzraudzības rīki var ietvert SIEM vai EDR, no kuriem vismodernākie var izmantot uzvedības analīzi, lai "iemācītu" sistēmām atšķirību starp regulārām ikdienas

darbībām un faktisko draudu uzvedību, samazinot cilvēku veikto datu šķirošanas un veicamās analīzes apjomu.

4) Brīdinājumu klasifikācija un pārvaldība, kurā *SOC* ir rūpīgi jāziskata visi ienākošie ziņojumi, jāidentificē visi viltus pozitīvie ziņojumi un jānosaka, cik kritiski ir faktiskie draudi un uz ko tie varētu būt vērsti. Tas ļauj atbilstoši šķirot jaunos draudus, prioritāri risinot steidzamākos jautājumus.
5) Reakcija uz draudu, kur tiklīdz incidents ir apstiprināts, *SOC* darbojas kā pirmais proaktīvais rīks, veicot tādas darbības kā ierīču izslēgšana vai izolēšana, kaitīgu procesu pārtraukšana, failu dzēšana un citas darbības. Mērķis ir reaģēt uz incidentu pēc iespējas minimāli ietekmējot organizācijas darba nepārtrauktību.
6) Atgūšanās pēc incidenta, kur *SOC* darbojas, lai atjaunotu sistēmas un atgūtu zaudētos vai bojātos datus. Tas var ietvert ierīču restartēšanu, sistēmu pārkonfigurēšanu vai, šifrējošās ļaunprātīgās programmas uzbrukuma gadījumā, datu kopiju izvietošanu ļaunprātīgai programmatūrai nepieejamā vietā. Galvenais mērķis ir atjaunot sistēmas, datus un procesus tādā stāvoklī kā tie bija pirms ir iestājies incidents.
7) Auditācijas pierakstu pārvaldība, kur *SOC* ir atbildīgs par visu tīkla darbību un komunikāciju auditācijas pierakstu apkopošanu, uzturēšanu un regulāru pārskatīšanu. Šie dati palīdz noteikt bāzes līniju "normālai" tīkla darbībai, kā arī var atklāt incidentus. Iegūtos datus var izmantot sistēmu koriģēšanai, kā arī izmeklēšanai pēc incidenta. Daudzi *SOC* izmanto SIEM, lai apkopotu un korelētu datu plūsmas no lietojumprogrammām, ugunsmūriem, operētājsistēmām un ierīcēm.
8) Cēloņu izmeklēšana pēc incidenta, kur *SOC* ir precīzi jānoskaidro, kas notika, kad, kā un kāpēc. Šīs izmeklēšanas laikā *SOC* izmanto auditācijas pierakstu datus un citu informāciju, lai izsekotu incidentam līdz tās rašanās brīdim, kā arī, lai novērstu līdzīgu incidentu rašanos nākotnē.
9) Drošības uzlabošana, kur *SOC* ir nepārtraukti jāīsteno dažādi uzlabojumi, jo kibernoziedznieki pastāvīgi pilnveido savus rīkus un taktiku. Šajā solī nepieciešams iedzīvināt drošības uzlabošanas plānus, kuros var ietvert arī praktiskus uzbrukumu – aizsardzības scenārijus.
10) Atbilstības pārvaldība, kur *SOC* procesi tiek īstenoti ņemot vērā labo praksi, kā arī dažādas atbilstības prasības, kā ISO27001, MK noteikumi 442, GDPR, PCI DSS, *SOC* ir jāveic regulāra atbilstības izvērtēšana. Rīcība saskaņā ar šiem noteikumiem palīdz ne tikai aizsargāt uzņēmumam uzticētos datus, tai skaitā personu datus, bet arī pasargā organizāciju no reputācijas riskiem un nesaskaņām ar tiesībsargājošām iestādēm.

Lai izveidotu *SOC* ir nepieciešama auditācijas pieraktu uzkrāšana, to korelēšana, SIEM sistēmas funkcionalitāte, uzdevumu nodošanas un kontroles funkcionalitāte, noteikta un papildināma zināšanu bāze, kas iekļauj atbilstošu ekspertu pieejamību, draudu identificēšanas funkcionalitāte un to izpēte, kā arī atskaišu ģenerēšanas iespēja (2.12 attēls) [82].



2.12. *SOC* elementi (adaptēts no *[82]*)

Lai arī dažkārt organizācijas, ņemot vērā viņu darba specifiku, pievērš lielu uzmanību kiberdrošībai, organizācijām joprojām trūkst spēju atklāt un adekvāti reaģēt uz ielaušanos viņu tīklos. Tam iemesls ir analītisko spēju trūkums korelēt milzīgu datu apjomu piemēram, *SOC* ietilpstošajos *IDS* brīdinājumos un identificēt tajos svarīgāko. Daudzos gadījumos datu analīzei ir izšķiroša nozīme, jo automatizētie pasākumi daudzos gadījumos nespēj "izprast" sarežģītu kiberuzbrukumu stratēģiju, pat ar uzlabotas korelētas diagnostikas palīdzību. Analītiķiem ir jāveic virkne pārbaužu, tostarp datu šķirošana, korelācija, draudu analīze, reaģēšana uz incidentiem un ekspertīze. Datu analīze ietver dažādu datu avotu (piemēram, *IDS* brīdinājumu un ugunsmūra žurnālu) detalizētu pārbaudi, viltus pozitīvo rezultātu atsijāšanu, saistīto rādītāju grupēšanu tā, lai dažādas uzbrukuma kampaņas būtu iespējams nodalīt vienu no otras. Datu šķirošana nepieciešama, lai sapratu kas ir analizējams detalizētāk kā arī, lai sagatavotu uzticamu, uz datiem balstītu atskaiti par uzbrukumu. Informācija par incidentiem kalpo par galveno pamatu turpmākai lēmumu pieņemšanai attiecībā uz to, kā mainīt pašreizējo drošības

konfigurāciju un kāda būs stratēģija rīcībai pret uzbrukumiem. Datu šķirošana ir laikietilpīgs process kiberanalītikā. Cilvēka smadzeņu spēja apstrādāt lielu datu apjomu ir objektīvi daudz mazāka salīdzinot ar datoru. Turklāt cilvēkiem rodas dažādas cilvēkiem raksturīgas problēmas, piemēram, nogurums, trauksme un pat depresija. Analītiķis var nespēt apstrādāt milzīgo datu apjomu no dažādiem datu avotiem, tāpēc ir nepieciešama datu šķirošanas automatizācija [83]. Šķirotie dati apjoma ziņā vairs nav pārmērīgi un analītiķim ir vienkāršāk izvērtēt dažādus notikumus. *SOC* speciālistiem jābūt proaktīviem, tāpēc kognitīvo tehnoloģiju izmantošana var palīdzēt analītiķiem veidot uzbrukumu modeļus, kas, savukārt, var mazināt uzbrukumu iestāšanās varbūtību, kā arī uzbrukumu ietekmi. Drošības analītiķiem jāņem vērā vismaz šāda informācija:

- Informācija par ievainojamībām;
- Dati no drošības iestādēm, piem. *CERT.LV*;
- Informācija par tīkla topoloģiju;
- URL savienojuma informācija;
- DNS informācija;
- IDS dati;
- Informācija par operētājsistēmām;
- Auditācijas pieraksti.

Reaģējot uz kiberdrošības speciālistu trūkumu, daudzas organizācijas piedāvā *SOC* pakalpojumu kā servisu [24] [25] [26] [27] [28] [29]. Vienojošais elements visiem apskatītajiem piedāvājumiem ir incidenta bīstamības pakāpes noteikšana un spēja reaģēt uz incidentu. Autoraprāt vislabāk incidenta bīstamības pakāpi var noteikt persona, kura ir detalizēti iepazinusies ar organizācijas darba specifiku un pārzina tās kritiskos informācijas resursus. Ārpakalpojums fokusēsies uz vispārpieņemtiem draudiem un to identificēšanu, nevis uz uzņēmumama specifiku, it īpaši, ja tas nodrošina *SOC* pakalpojumu lielam klientu skaitam, tādēļ autoraprāt, labāk *SOC* veidot organizācijas iekšienē. *SOC* var iedalīt 4 līmeņos [84]:

1) Minimālais *SOC* līmenis: galvenokārt koncentrējas uz atklāšanu (ne tik daudz uz izmeklēšanu). Analītiķi galvenokārt strādā ar SIEM, kas tika izvietots pirms vairākiem gadiem, un tas nav atjaunināts. Kopumā šīs tehnoloģijas piedāvā apmierinošas draudu identificēšanas iespējas, taču nav pietiekami labas, lai veiktu detalizētu izmeklēšanu.

2) Vidēja līmeņa *SOC*: papildu SIEM, kas nodrošina notikumu žurnālus, tiek pielietots EDR un tīkla kriminālistikas tehnoloģiju kombinācija, kas nodrošina uzlabotu draudu noteikšanu. Drošības analītiķi darbojas vairākos līmeņos. Komanda pievērš lielu uzmanību proaktivitātei, taču realitātē tā ir apgrūtināta.
3) Uzlabots *SOC*: organizācija ir ieguldījusi milzīgus resursus instrumentos, kā arī atbrīvojusi savu analītiķu laiku. Pirmā un otrā līmeņa *SOC* analītiķi galvenokārt strādā ar SIEM, lai pielāgotu savus korelācijas noteikumus un adaptētu dažādus specializētākus produktus SIEM. Viņi iegūst datus no tīkla un galiekārtām. Tas uzlabo viņu izmeklēšanas kvalitāti (un ātrumu). Kad tiek identificēts incidents pirmajā vai otrajā līmenī, trešā līmeņa analītiķi to apstrādā ar īpašiem, specializētiem analīzes rīkiem. Progresīvie *SOC* bieži var iekļaut "medību" komandu, kas nav daļa no *SOC* komandas. Viņi koncentrējas tikai uz to, kā medīt incidentus, kuras *SOC* tehnoloģijas nespēja identificēt. Kaut arī viņi darbojas ar SIEM, tomēr lielāko daļu laika pavada, veidojot un darbinot pielāgotus skriptus, lai atrastu draudus, par kuriem viņu drošības produkti neinformē. Visbeidzot, inženieri izveido programmatūru, kas viņu drošības produktiem liek sazināties savā starpā. Tas palīdz pilnveidot viņu procesus un pēc iespējas labāk automatizēt datu vākšanu, kā arī reaģēt uz incidentiem. CISO periodiski piesaista trešās puses, lai veiktu "sarkanās" komandas vingrinājumus un nodrošinātu, ka *SOC* darbojas tā, kā paredzēts.
4) Pilnvērtīgs *SOC*: tāpat kā trešā līmeņa *SOC*, organizācija ir ieguldījusi lielus resursus automatizācijā un analīzē. Pārsvarā notiek koncentrēšanās uz automatizāciju un cilvēks tiek piesaistīts tikai īpašos gadījumos, kurus var veikt tikai cilvēki. Tiek sasaistītas drošības tehnoloģijas ar drošības pārvaldības sistēmu un piesaistīti IT resursi, lai automatizētu izmeklēšanas procesu.

Eksperti [84] norāda, ka 24x7 drošības komandu veidošanā, 12 cilvēki ir minimālais cilvēku skaits darbam ar *SOC*, tādēļ izmaksas ASV sākas no 1milj. dolāriem gadā.

2.7.tabula

SOC ieviešanas un uzturēšanas izmaksas ASV

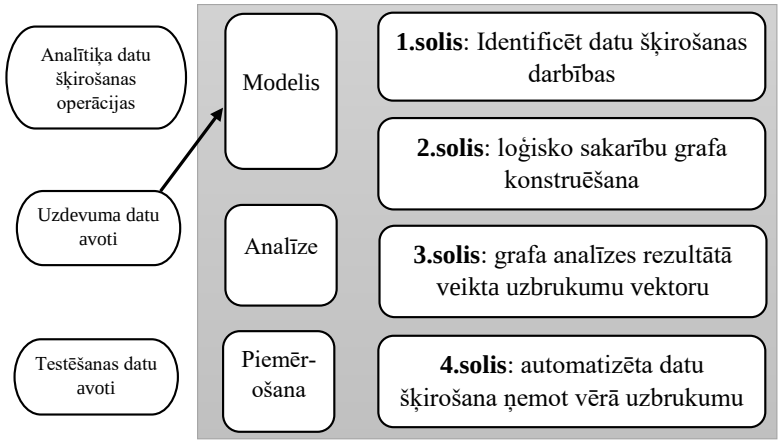
Izmaksas gadā Vašingtonā, ASV				
	1.līmeņa *SOC*	2.līmeņa *SOC*	3.līmeņa *SOC*	4.līmeņa *SOC*
Vidējās izmaksas uz	0,25	0,35	1,1	1,9

rīkiem (milj. ASV dolāri)				
Vidējās 12 cilvēku personāla izmaksas (milj. ASV dolāri)	1,42	2,38	4,9	6,25
Vienreizējās ieviešanas izmaksas (milj. ASV dolāri)	0,1	0,25	0,4	0,75

Diemžēl izmaksas *SOC* sistēmu izveidē un ieviešanās ir ļoti lielas (skat. 2.7.tabulu), Latvijā varētu būt mazākas tikai darbaspēka izmaksas. Tāpēc, lai samazinātu izmaksas autors piedāvā *SOC* lēmumu pieņemšanu maksimāli automatizēt, iesaistot ierīces lietotāju.

2.3.1 Automatizēta datu šķirošana *SOC* sistēmā

Pētnieki Zhong et al. [83] piedāvāja veikt datu šķirošanu, lai noskaidrotu uzbrukuma vektorus (2.13.att.).



2.13.att. *SOC* datu šķirošanas pieeja (adaptēts no *[83]*)

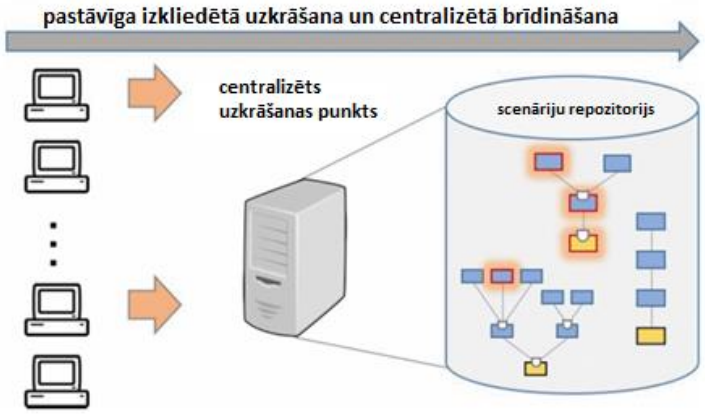
Pēc pētnieku domām, šādi šķirojot datus bija iespējams atvieglot analītiķa darbu, analizējot milzīgus datu apjomus *SOC* sistēmā, tādēļ, ka analītiķa spēja apstrādāt milzīgus datu apjomus ir ierobežota. Pētnieki atzina, ka lietojot viņu piedāvāto sistēmu, viltus pozitīvo skaits, lai gan ir samazinājies, tomēr joprojām pastāvēja.

2.3.2 Datu korelēšana

Pētnieki Rose et al. [50] piedāvāja sistēmu žurnāldatu analīzei gandrīz reāllaikā, kas balstīta uz procesu modeļiem, izmantoja izkliedētu datu apstrādi un sūtīja rezultātus uz centrālo datu uzkrāšanas punktu korelācijas un kontekstualizācijas mērķiem. Pētnieki uzskata, ka izkliedētā datu apstrāde ir mūsdienīgs veids kā sasniegt mērogojamību un apstrādāt lielu datu apjomu. Šāda pieeja palīdz saīsināt reakcijas laiku nopietnu drošības incidentu gadījumā, apstrādājot tikai svarīgos notikumus. Pētnieku piedāvātā metodes pamatā bija:

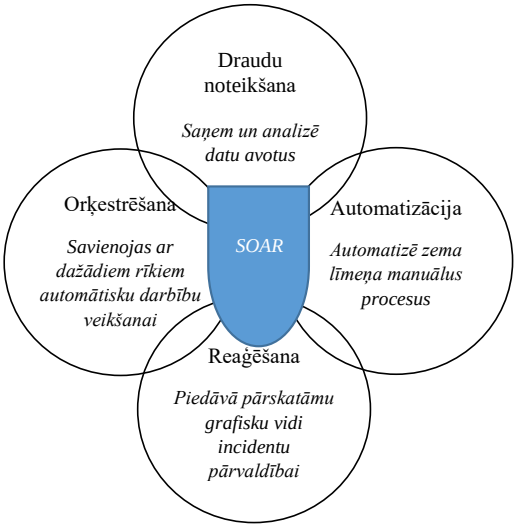
1) Kritisko resursu un sensitīvo datu prioritizēšana, nosakot sistēmas, kuras būtu jāuzrauga.
2) Interesējošo scenāriju identificēšana – noteikta soļu secība, kas novedusi pie dažādiem iznākumiem. Scenārijus autori sasistīja ar paredzamo sistēmas uzvedību. Piemēram, sensitīvas datu bāzes integritātes apdraudējuma gadījumā, bija nepieciešami pasākumi, kas saistīti ar piekļuvi datu bāzei un manipulācijām ar tās saturu.
3) Tikai interesējošo darbību monitorēšana. Izplatīta kļūda ir ieslēgt pilnīgi visus auditācijas pierakstus un tādā veidā "noslīkt" tajos. Galvenais, ir identificēt kuri auditācijas pieraksti ir spējīgi sniegt vislielāko pievienoto vērtību drošības draudu identifcēšanai.
4) Rezultātu apkopošana un korelēšana iekļaujot tos vienotā repozitorijā. Tas ļauj korelēt datus izmantojot to kontekstu, nodrošinot prioritāru, gandrīz reāllaika uzraudzību un brīdinājumu nodošanu *SOC* operatoram.

Autoru piedāvātā pieeja ir parādīta 2.14.attēlā.



2.14.att. Centralizētas datu korelēšanas sistēmas darbība (adaptēts no *[84]*)

Vēl viena mūsdienu datu automatizācijas metode ir *SOAR* (*Security automation and orchestration*), kura sastāv no 4 komponentēm (2.15.attēls).



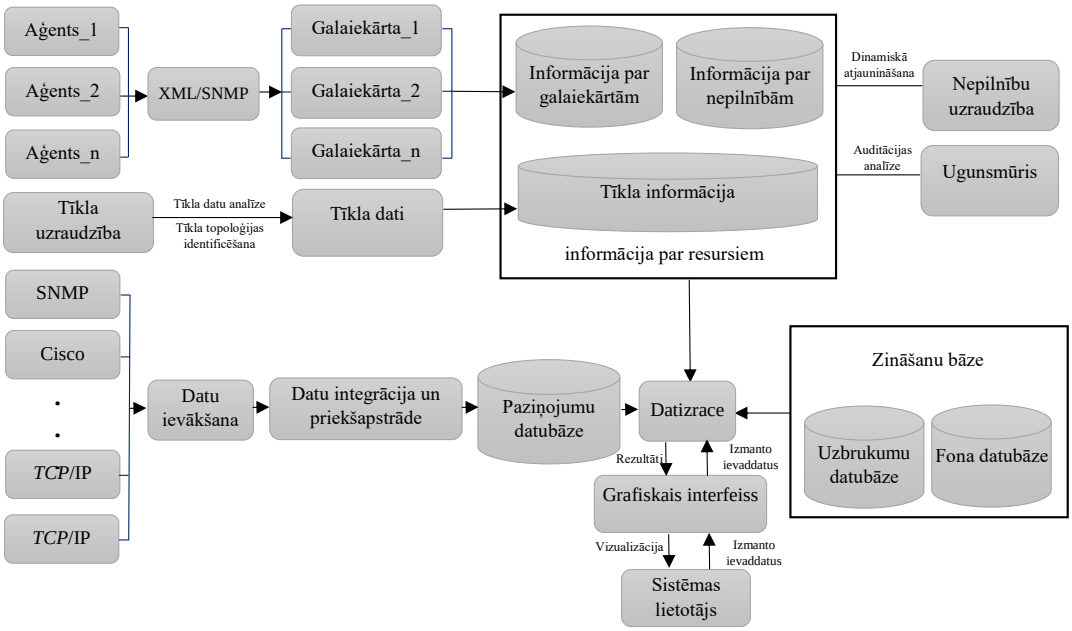
2.15. attēls. *SOAR* komponentes (adaptēts no *[136]*)

Šī metode ietver draudu noteikšanu, drošības automatizāciju, orķestrēšanu un automātisku reaģēšanu uz incidentiem, lai novērstu kiberuzbrukumus. Automatizācija tiek veikta pamatojoties uz izstrādāto reaģēšanas scenāriju uz incidentiem un definējot, kādi uzdevumi ir jāveic, lai incidentu novērstu. Izmantojot *SOAR* metodi, drošības operāciju centra (*SOC*) komandas, kuras iepriekš bija pārslogotas ar atkārtotiem un laikietilpīgiem uzdevumiem, ir spējīgas efektīvāk risināt incidentus, samazinot izmaksas un palielinot produktivitāti. Drošības automatizācija sniedz iespēju programmēt uzdevumus, brīdinājumus vai automātiskas atbildes uz identificētajiem incidentiem. Automatizācija palīdz paātrināt procesus, piemēram, draudu meklēšanu un novēršanu, veicot maz vai neveicot nekādas manuālas darbības. Šādi racionalizējot procesus, *SOC* komandām ir iespēja koncentrēties uz svarīgiem drošības incidentiem, kuru novēršanai nepieciešama manuālu iejaukšanās. Drošības orķestrēšana sniedz iespēju koplietot informāciju veidojot savienojumus ar dažādiem rīkiem. Ar orķestrēšanas palīdzību šie rīki veikt dažādas darbības, piemēram liegt piekļuvi datiem vai datu tīklam [85].

Predefinēti scenārji ir viena no svarīgākajām efektīvas *SOAR* metodes sastāvdaļām. Piemēram, ja skenēšanas laikā darbinieka e-pastā tiek atrasts ļaunprātīgs *URL*, var izveidot scenāriju, kas ievieto e-pastu nevelēlamo e-pastu mapē, brīdina darbinieku par iespējamo pikšķerēšanas mēģinājumu un bloķē sūtītāja IP adresi [86].

2.3.3 Datizrace

Datizrace ir datu analīzes process, ko var izmantot anomāliju noteikšanai, piemēram, no savāktajiem žurnālfailiem. Pētnieki Li et al. [40] piedāvā drošības modeli, kura pamatā ir datizrace (2.16.att.). Pētnieku pamatideja ir integrēt visu tīkla drošības produktu ģenerētos datus vienotā sistēmā, pēc tam tos automātiski analizēt un reaģēt, atstājot tikai *SOC* operatoram interesantos datus.



2.16.att. Datizraces drošības modelis (adaptēts no *[40]*)

Šāda pieeja [40] nodrošina organizācijas aizsardzības spēju pret dažāda veida kiberuzbrukumiem.

2.3.4 Secinājumi

Mūsdienās automatizētas draudu identificēšanas metodes ir svarīga drošības pārvaldības komponente. Šīs komponentes uzlabo ļaunprātīgas darbības atklāšanas precizitāti, kā arī novērš cilvēku izdegšanas un kļūdas elementu. Izmantojamie paņēmieni var variēties no pusautomātiskiem līdz pilnīgi automatizētiem (pat bez cilvēka klātbūtnes). Saskaņā ar šajā nodaļā apskatītām metodēm (2-8.tabula) daudzi pētnieki ir pievērsušies datu apstrādei, korelācijai un klasificēšanai, jo milzīgs auditācijas datu apjoms bez automatizācijas metodēm ir nelietojams mūsdienu drošības pārvaldības sistēmās.

2.8.tabula

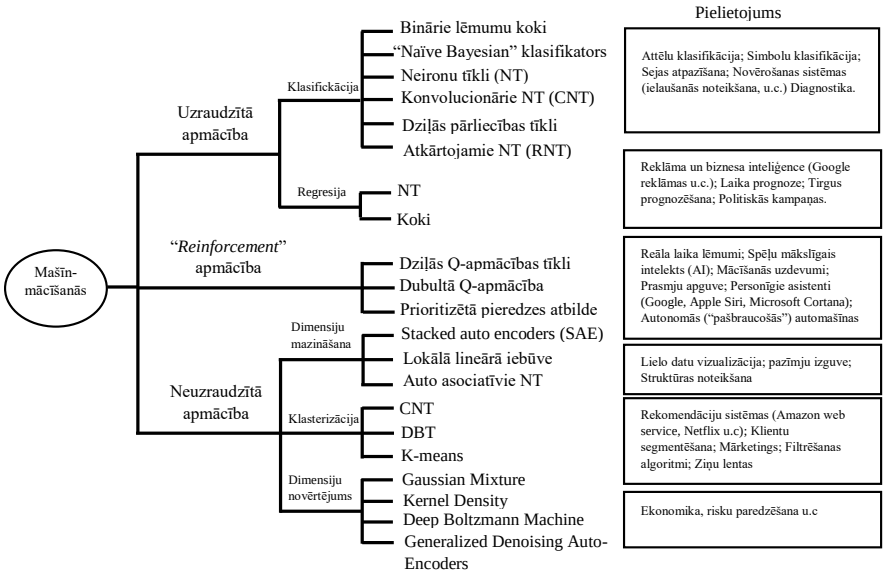
Apskatīto mūsdienu automatizācijas metožu pārskats

Metode	Metodes mērķis	Identificējamie uzbrukumi	Priekšrocības	Trūkumi
Automatizēta datu šķirošana *SOC* sistēmā	Liela datu apjoma šķirošana, analītiķa darba atvieglošanai	Uzbrukumu vektori	Samazināts viltus pozitīvo incidentu skaits, kā arī spēja analizēt lielus datu apjomus no dažādiem avotiem	Nepieciešama analītiķa klātbūtne, kā arī joprojām pastāvošie viltus pozitīvie ziņojumi
Datu korelēšana	Korelēt un kontekstualizēt dažādu avotu datus apvienojot vienā centrālajā sistēmā	Uzbrukumu vektori	Pieeja palīdz paātrināt reakcijas laiku uz nopietniem drošības incidentiem	Nepieciešama analītiķa klātbūtne, iespēja automatizēt, ja pielietots SOAR
Datizrace	Anomāliju noteikšana no savāktajiem žurnālfailiem	Anomāliju identificēšana	Iespēja apstrādāt lielus datu apjomus un paātrināt reakcijas laiku uz nopietniem drošības incidentiem	Nepieciešama analītiķa klātbūtne, iespēja automatizēt

No konteksta atkarīgajā, adaptīvajā drošības pārvaldības modelī un tā tehniskajā realizācijā tika izmantoti visi iepriekš apskatītie automatizācijas līdzekļi, tādi kā datu korelēšanu, datizrace, automātiska datu šķirošana, kā arī SOAR metode. Šāda pieeja ļāva automatizēt drošības pārvaldības procesus un pielietot uz scenārijiem bāzētu drošības incidentu pārvaldību.

2.4 RQ4: MAŠĪNMĀCĪŠANĀS METODES IS DROŠĪBAS PĀRVALDĪBAS NODROŠINĀŠANAI

Mašīnmācīšanās metodes (2.17.att.) ir automatizētas metodes, kuras uzlabo savu sniegumu,



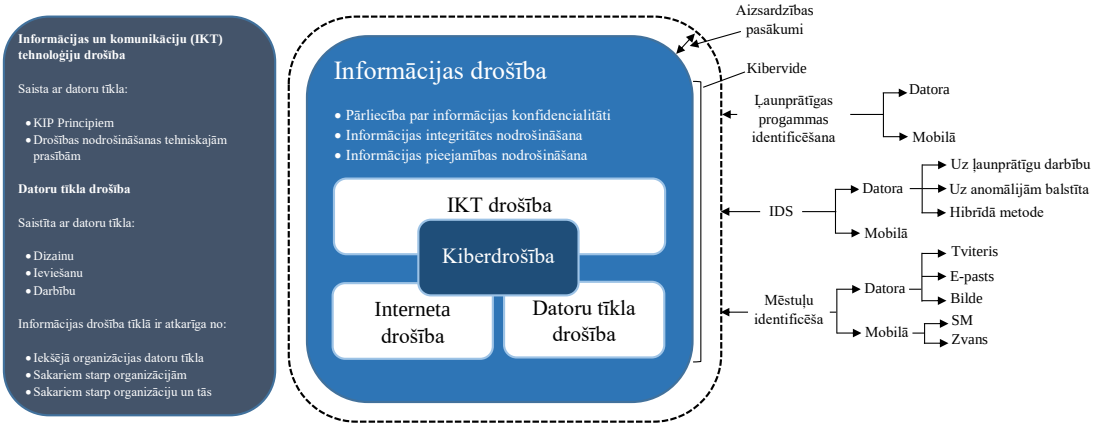
2.17.att. Mašīnmācīšanās metodes (adaptēts no *[87]*)

ņemot vērā apmācības laikā apstrādātos datus un iegūto iepriekšējo pieredzi. Mašīnmācīšanos var pielietot, piemēram, attēlu klasificēšanā, reāla laika lēmumu pieņemšanā, drošības pārvaldībā un citos uzdevumos, kuros nav precīzi nodefinēta likumu kopa lēmumu pieņemšanai. Mašīnmācīšanās var tikt klasificēta:

1) Uzraudzītajos mašīnmācīšanās algoritmos, kuri izmanto marķētus datu piemērus, lai mācītos un pēc tam veiktu prognozes, piemēram, atbalsta vektora mašīnas, neironu tīkli, NïB.
2) Neuzraudzītajos mašīnmācīšanās algoritmos, kuri apraksta datu struktūru no nemarķētiem datiem, lai izdarītu secinājumus, kā piemēram, klasterizācija.
3) Daļēji uzraudzītajos mašīnmācīšanās algoritmos, kuri apmācībai izmanto gan neliela apjoma marķētus, gan liela apjoma nemarķētus datus.
4) Stimulētās *(Reinforcement)* mašīnmācīšanās algoritmos, kuru metode ir mijiedarbojoties ar vidi, lai identificētu savas kļūdas un saņemtu atlīdzību.

Izmantojot mašīnmācīšanos, drošības pētnieki piedāvā dažādus veidus, kā identificēt, piemēram, vai ierīce ir robotu tīkla sastāvdaļa. Pārsvarā visos mūsdienu pētījumos tiek pielietoti mašīnmācīšanās (ML) algoritmi ļaunprātīga koda aktivitātes identificēšanai, bet diemžēl lielākā

daļa pētījumu tiek veikti izmantojot simulētas vai publiski pieejamas datu kopas [87]. Uz ML balstītas metodes darbojas labāk nekā uz signatūrām balstītās sistēmas, jo nelielas uzbrukuma modeļa variācijas var viegli apiet uz signatūrām balstītu *IDS*. ML tiek pielietota abās pusēs, t.i., uzbrucēja un aizsardzības pusē. Kibernoziedznieku pusē uzbrucēji izmanto ML paņēmienus, lai atrastu sistēmas ievainojamības un sarežģītus uzbrukuma veidus, kā apiet ugunsmūri vai apmānīt lietotāju. Aizsardzības pusē ML modeļi spēlē būtisku lomu, lai veicinātu uzbrukumu agrīnu atklāšanu un mazinātu uzbrukuma radīto ietekmi. Autori [87] uzsver būtisku mašīnmācīšanās algoritmu lomu kiberdrošībā (2.18.att.)

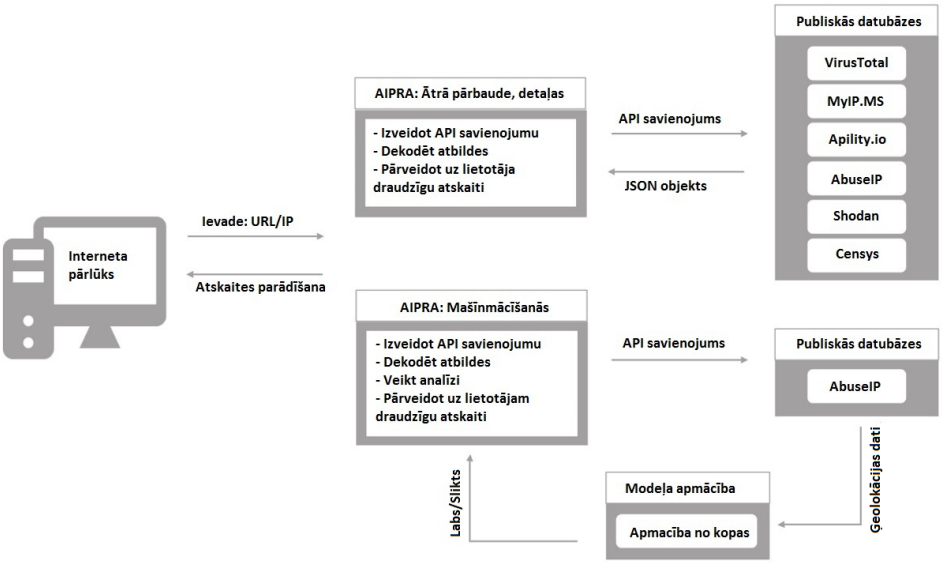


2.18.att. Mašīnmācīšanās algoritmu loma kiberdrošībā (adaptēts no *[87]*)

2.4.1 Uzraudzītie mašīnmācīšanās algoritmi

Uzraudzītie mašīnmācīšanās algoritmi var tikt izmantoti, lai paredzētu vai IP adrese vai domēna vārds ir iesaistīti ļaunprātīgā darbībā. To var sasniegt apmācot mašīnmācīšanās algoritmu, klasificējot datus "sliktajos" un "labajos", pētnieki Lewis et al. [88] piedāvāja automatizētu IP reputācijas analīzes rīku. Apmācības nodrošināšnai pētnieki bija ņēmuši datus no AbuseIPDB [89], izveidojot "labo" un "slikto" IP adrešu datubāzi. Apmācības nodrošināšanai tika eksportēti tādi parametri kā geolokācija un cita datus atšķirošā informācija.

Tika izmantotas dažādas klasificēšanas metodes: *RFC*, NïB metode, Regresijas metode un, rezultātā par labāko ar 76% precizitāti tika atzīta *RFC* klasificēšanas metode (2.19. att.).



2.19.att. Piedāvātā klasificēšanas metode (adaptēts no *[88]*)

Citi autori [90] ar uzraudzīto mašīnmācīšanās algoritmu palīdzību veica DNS pieprasījumu analīzi, klasificējot pieprasījumus kā leģitīmus vai algoritmu pieprasītus (*DGA*), jeb "labajos" un "sliktajos" pieprasījumos. "Labie" domēna vārdi tika ņemti no Alexa top miljons domēna vārdiem [30], savukārt "sliktie" no dažādiem drošības pētnieku resursiem [91], [89], [92]. Domēna vārdiem tika identificētas dažādas pazīmes, tādas kā, vai domēns ir tiešsaistē, cik IP adreses ir piesaistītas domēnam, neparasto simbolu skaits domēna vārdā, domēna vecums un citi parametri. Mašīnapmācībai tika izvēlēti NïB, *SVM*, lēmumu koki, gadījuma meži, regresija un neironu tīkli. Vislabāko rezultātu uzrādīja lēmumu koki ar 92% precizitāti. Pētnieku piedāvātā lēmumu koku klasifikācija spēja nodrošināt rezultātu 10 sekundēs, kas reālā laika sistēmās varētu nebūt labs rezultāts.

Literatūras analīzes rezultātā var secināt, ka vairums pētnieku (piemēram [71] [72]) piedāvāja izmantot uzraudzītos mašīnmācīšanās modeļus, lai identificētu *DGA*. Bieži vien domēna vārds tiek sadalīts un pētīts izmantojot n-gram pieeju, izveidojot no sadalītā vārda pazīmju kopu, kuru vēlāk var izmantot, lai apmācītu mašīnmācīšanās algoritmu. Pētnieki [71] guva labus rezultātus eksperimentos, izmantojot n-gram analīzi. Viena no šīs metodes problēmām ir dažādu pakalpojuma sniedzēju pāreja uz šifrētu DNS pieprasījumu apstrādi, piemēram Firefox piedāvā jau iebūvētu moduli DNS over HTTPs, kur tiek izmantots standarta

HTTPs savienojums DNS pārraidīšanai [93]. Līdzīga pieeja ir Google [94] un Cloudfare [78], kur tiek piedāvāts šifrēt pieprasījumus izmantojot transporta datu slāni (TLS). Lai arī kā, bet joprojām liels DNS pieprasījumu skaits tiek apstrādāts izmantojot nešifrēto DNS protokolu.

2.4.2 Daļēji uzraudzītie mašīmācīšanās algoritmi

Pētnieki Sun et al. [95] piedāvāja HGDom sistēmu ļaunprātīgu domēnu noteikšanai, kas ņēma vērā gan domēna pazīmes, gan to globālās asociācijas. Piedāvātā sistēma bāzējas uz informāciju no trim skatu punktiem: 1) simbolu sakarību domēna vārdā, 2) uzbrucēju informācijas apkopošanu no dažādiem avotiem, 3) klientu DNS pieprasījumu veida. Ļaunprātīgas domēnu identifikācijai tiek pielietota grafu metode. Pētnieki bja izstrādājuši dziļās mācīšanās metodi, kura bija spējīga noteikt ļaunprātīgus DNS pieprasījumus reālā organizācijas vidē. Sava modeļa darbspējai pētnieki apstrādāja trīs veidu datus: 1) aktīvie DNS dati tīklā ar dažādiem papildu laukiem, kā avota IP adrese, TTL; 2) pasīvie DNS dati, kuros atspoguļojas cik reizes un kurš ir pieprasījis konkrēto domēnu; 3) DNS dati no lokālā DNS servera ar klientu pieprasījumu laika intervāliem, kas atspoguļo klientu uzvedību attiecībā uz domēnu pieprasīšanu. Piedāvātās sistēmas arhitektūra sastāvēja no datu ievākšanas un priekšapstrādes, grafa konstruēšanas komponentes kā arī ļaunprātīga DNS identificēšanas komponentes. Lai uzlabotu precizitāti pētnieki piedāvāja, ievācot datoru tīkla datus, atsijāt sekojošas datu kategorijas:

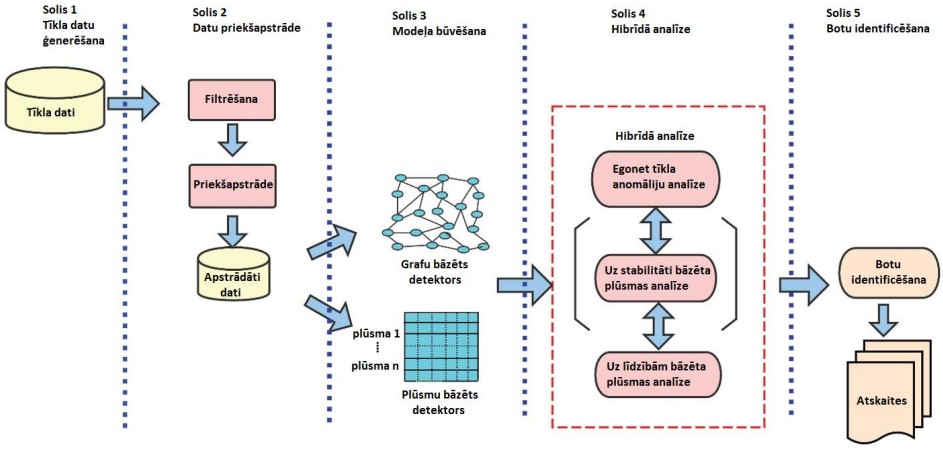
1) neaktīvi klienti, jeb tādas IP adreses, kuras pieprasa domēnus retāk par noteiktu laika intervālu t, piemēram t>2;
2) pārāk aktīvi klienti, kuri izsauc lielu domēnu skaitu, kas visdrīzāk ir starpniekserveri;
3) domēnu vārdu izsaukumi, kuri neatbilst (*RFC 1035*) prasībām;
4) populāri domēnu vārdi, kurus ir pieprasījušas vairāk kā 50 tīkla iekārtas;
5) retas IP adreses, kuras pieprasa tikai vienu domēnu.

Autoru piedāvātā HGDom sistēma spēja sasniegt kopējo precizitāti 98.92%, klaisfikatora precizitāti 97.52% un pārklājumu 98.17%.

2.4.3 Neuzraudzītie mašīnmācīšanās algoritmi

Neuzraudzītie mašīnmācīšanās algoritmi tiek pielietoti, lai klasificētu datus dažādās datu kopās. BotMark ir viena no sistēmām robottīklu noteikšanai, kurā tika izmantota datu plūsmas un uz grafu bāzēta uzvedības hibrīda analīze [96]. Metode sastāvēja no pieciem posmiem: datu plūsmas ģenerēšana, datu priekšapstrāde (neatbilstošu datu plūsmu filtrēšana un saistīto plūsmu apkopošana C-plūsmās), modeļa veidošana (15 avoti, kas balstīti uz statistisko plūsmu) un

galamērķa IP un ports, plūsmas ilgums utt., kā arī 3 uz grafu balstītas funkcijas), hibrīdā (līdzības analīze, stabilitātes un grafu analīze) un robotu identifikācija (2.20.att.).



2.20.att. Autoru piedāvātā sistēma (adaptēts no *[96]*)

Analīzes mērķis bija grupēt līdzīgas C-plūsmās (kopās), apkopojot uz plūsmām balstītas pazīmes, veicot K-NN klasterizāciju un aprēķinot līdzības parametrus. Pazīmes noteiktas 2.9.tabulā.

2.9.tabula

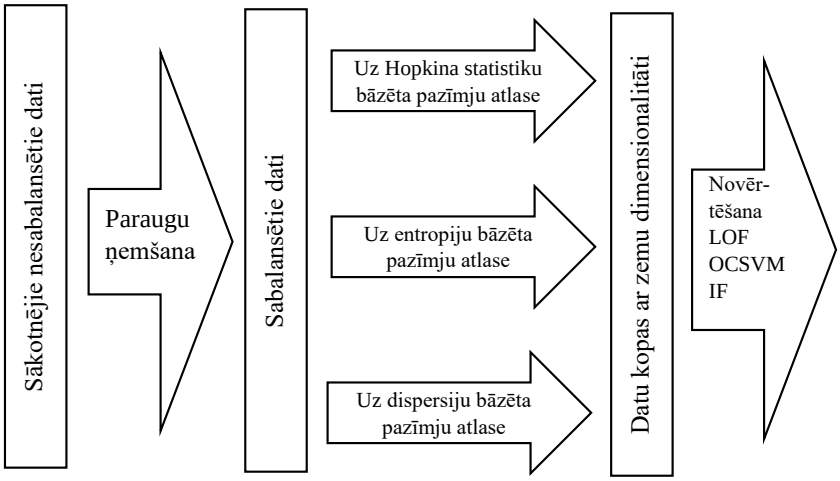
Pētnieku piedāvātās pazīmes mašīnmācīšanās algoritmam (adaptēts no *[96]*)

Avota IP adrese
Avota porta adrese
Galamērķa IP adrese
Galamērķa porta adrese
Plūsmas ilgums
Kopējais nosūtīto pakešu skaits
Mazo pakešu skaits (garums 63–400 baiti)
Vidējais pakešu ierašanās laiks
Kopējais pārsūtīto baitu skaits
Vidējais derīgo datu apjoms laika intervālā
Nosūtīto derīgo datu apjoma standartnovirze

Plūsmas pirmās paketes lielums
Kopējais dažādu izmēru pakešu skaits, salīdzinot ar kopējo pakešu skaitu
Maksimālās paketes garums plūsmā
Maksimālā garuma pakešu skaits plūsmā
Kopējais baitu skaits, ko pārsūta lielākā pakete
Vidējais bitu skaits sekundē
Pakešu skaits sekundē
C-plūsmu skaits stundā

Atsevišķas C-plūsmas stabilitāte tika noteikta pamatojoties uz datu pakešu garuma sadalījumu. Uz grafu balstītās analīzes mērķis bija identificēt robottīkla komandcetrus (C&C), ņemot vērā grafu visrotnes un izmantojot autoru piedāvāto mazākā kvadrāta pieeju, lai identificētu anomālijas. Sistēmas veiktspēja, izmantojot datu pakešu garuma sadalījuma metodi, nodrošināja labus rezultātus un sasniedza 99,49% precizitāti un 0,51% viltus pozitīvu rādītāju. Uz grafu balstīta metode sasniedza 91,66% precizitāti un 8,35% viltus pozitīvu rādītāju. Apvienojot abas metodes, BotMark panāca 99,94% precizitāti un 0,06% viltus pozitīvu rādītāju.

Citi pētnieki [97] piedāvāja anomāliju identificēšanu, lai identificētu vai IoT ierīce atrodas robotu tīklā. Autori no datu plūsmas eksportēja 115 pazīmes, izmantojot dažādu IoT ierīču, tādu kā drošības kameras, bērnu monitorēšanas sistēmas, viedie termometri u.c. tīkla komunikācijas dati. Autoru piedāvātā datu apstrādes diagramma attēlota 2.21. attēlā. Metodes pamatā ir normālas uzvedības atšķiršana no inficētas ierīces uzvedības izmantojot *local outlier factor (LOF), one class SVM* un *isolation forest (IF)*. Autori secinājuši, ka pielietojot viņu metodi iespējams sasniegt precizitāti, kas pārsniedz 99,8%.

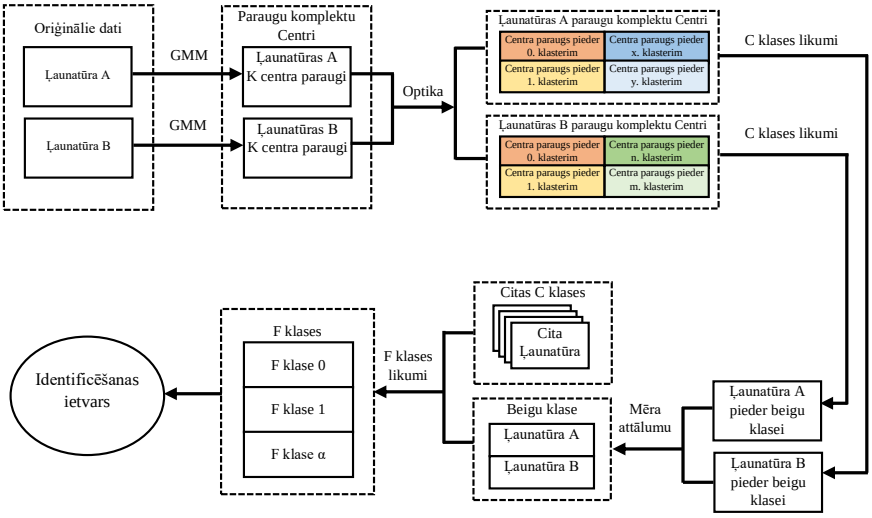


2.21.att. Datu apstrādes diagramma (adaptēts no *[97]*)

Citi autori [98] piedāvāja metodi kā identificēt robotu tīkla aktivitāti šifrētos tīkla datos, jo mūsdienās robotu sazinās izmanojot šifrētus kanālus un arvien sarežģītāk ir tos identificēt. Autoruprāt [98], klasisks ļaunprātīgas darbības identificēšanas ceļš ir izmantot ekspertu sistēmas, kurās tiek klasificēta "labā" un "sliktā" komunikācija, bet ne vienmēr šādi eksperti ir pieejami un viņiem ir pietiekošas zināšanas, lai klasificētu komunikāciju. Tādēļ vat tikt pamatota neuzraudzītu mašīnmācīšanas algoritmu izmantošana. Autori [98] ļaunprātīgo programmatūru iedala divos veidos:

1) inficēšanas ļaunatūra, kur ļaunprātīgas programmatūras galvenokārt veic inficēšanas uzdevumu. Pēc inficēšanas tā veic turpmākas ļaunprātīgas darbības. Ir divi populāri infekcijas veidi. Pirmais ir balstīts uz Exploit komplektu (EK) tīmekļa lapā, piemēram, RIG. Otra iespēja ir izmantot pikšķerēšanas e-pastu, lai inficētu ierīci lietotājam uzklikšķinot uz saites vai pievienotā faila;
2) komandu un kontrolcentra ļaunatūra (C&C). Pēc iekļūšanas sistēmā ļaunprātīga programmatūra sazinās ar komandcentru, lai iegūtu jaunas instrukcijas vai nozagtu sensitīvu informāciju. Šāda veida datplūsma parasti ir paslēpta normālā datu plūsmā.

Pētījumā metodes pamatā tika izstrādāts attāluma mērs, lai noteiktu ļaunprātīgas programmatūras līdzības un definēta F-klase, kas satur vairākus ļaunprātīgas programmatūras veidus, kuros ļaunprātīgajai programmatūrai ir līdzīgs komunicēšanas veids (2.22.att.).



2.22.att. Piedāvātā attāluma mērīšana un klasterizācija (adaptēts no *[98]*)

Piedāvātās metodes precizitāte bija 91.74%.

2.4.4 Secinājumi

Mašīnmācīšanās izmantošana informācijas sistēmu drošības nodrošināšanai mūsdienās ir atrodama gandrīz jebkurā komerciālā produktā. Industrija ir sapratusi, ka cilvēka spējas šajā jomā ir ierobežotas. Šajā nodaļā tika aprakstītas dažādas mašīnmācīšanās metodes un katrai no metodēm ir savas priekšrocības un savi trūkumi, piemēram, ar uzraudzīto mašīnmācīšanās metodi iespējams izmantot tādus parametrus kā geolokācija, definējot riskanto valstu grupas, savukārt neuzraudzītās mašīnmācīšanās metodes ir spējīgas identificēt anomālu ierīces uzvedību "mācoties" no tīkla datiem. Uzraudzītās mašīnmācīšanās metodes ir atkarīgas gan no pirms tam uzkrāto datu precīzas klasificēšanas, gan no ekspertu ekspertīzes. Lielākā problēma ir uzkrāt pietiekošu skaitu ar ļaunprātīgās aktivitātes datiem, lai apmācītu algoritmu. Savukārt neuzraudzītās mašīnmācīšanās metodes var nepamanīt ļaunprātīgu komunikāciju, kuras darbības ir specifiski pielāgotas, lai līdzinātos normālai komunikācijai. Mašīmācīšanās algoritmu pielietojums informācijas sistēmu drošības nodrošināšanā ir ļoti plašs (2.10.tabula), bet pārsvarā pētnieki izvēlās izmantot uzraudzītās mašīnmācīšanās metodes. Turpmāk darbā ir izmantotas uzraudzītās mašīnmācīšanās metodes, izmantojot *DTC, RFC, NNC, SVM* un *KNN* algoritmus *DGA* un *NFAI* modeļu būvēšanai. Darbā piedāvātais, no konteksta atkarīgais, adaptīvais drošības pārvaldības modelis nodrošina vairāku neatkarīgu, uz atšķirīgām

mašīnmācības metodēm balstītu moduļu izmantošanu un to datu agregēšanu ar mērķi noteikt ierīces kopējo apdraudējumu.

2.10.tabula

Mašīmācīšanās algoritmu pielietojums informācijas sistēmu drošības nodrošināšanā

Mašīnmācīšanās algoritms	**Algoritma tips**	**Apraksts**	**Pielietojums**	**Trūkumi**
Gadījuma meži *(Random Forest – RFC)*	Uzraudzīts	Sastāv no dažādām lēmumu koku (LK) datu kopām. Katrs LK veic prognozēšanu. Prognoze ar vislielāko balsu skaitu ir algoritma kopējā prognoze	Ļaunprātīga IP, domēna, zombētas ierīces identificēšana, *IDS* uzlabošana, anomāliju noteikšana	Apstrādes izmaksas ir augstas. Lēna prognozēšana
(Support Vector Machine-SVM)	Uzraudzīts	Maza pārapmācīšanas *("overfitting")* iespēja	Ļaunprātīga IP, domēna, ļaunprātīgas programmatūras identificēšana izmantojot IP reputāciju, zombētu ierīču noteikšana, *IDS* efektivitātes uzlabošana	Nespēja efektīvi tikt galā ar lielām un diversificētām datu kopām. Augstas apstrādes izmaksas
Naïve Bayes (NïB)	Uzraudzīts	Klasifikators, kuram nepieciešams	Ļaunprātīga IP un domēna identifikācija	Nodefinē iespējamību 0, ja dažas testa

Mašīnmācīšanās algoritms	Algoritma tips	Apraksts	Pielietojums	Trūkumi
		maz resursu. Pieņem, ka pazīme ir pilnībā neatkarīga un nav saistīta ar citām pazīmēm		kopas kategorijas nav pārstāvētas apmācību kopā. Uzglabā visus apmācību datus. Nepieciešama liela datu kopa, lai sasniegtu labus rezultātus
Neironu tīkli (*NNC*)	Uzraudzīts	Adaptīvs un sastāvošs no savā starpā savienotiem neironiem algoritms. Nākamā slāņa ieejas dati ir atkarīgi no iepriekšējā slāņa rezultāta	Ļaunprātīga IP un domēna identificēšana. Ļaunprātīgas programmatūras komunikācijas datu identificēšana	Augstas apstrādes izmaksas. Liels apstrādes laiks. Melnās kastes tipa modelim nav saistības starp ievades un izvades mainīgajiem
Lēmumu koki (*DTC*)	Uzraudzīts	Strādā pēc ja-tad principa, lai atrastu labāko mezglu. Process tiek turpināts līdz tiek atrasta labākā prognozes klase	Ļaunprātīga IP un domēna identificēšana. Ļaunprātīgas programmatūras un zombētu ierīču identificēšana	Grūti mainīt datus neietekmējot kopējo struktūru. Sarežģīts algoritms ar augstām apstrādes izmaksām un

Mašīnmācīšanās algoritms	Algoritma tips	Apraksts	Pielietojums	Trūkumi
				lielu apstrādes laiku
K-vidējie (*K-means*)	Neuzraudzīts	Uzsāk darbu ar gadījuma centriem tad, veicot iterācijas atrod nepieciešamos klāsterus	Tīkla ielaušanās noteikšana un zombētu ierīcu identificēšana	Liela algoritma atkarība no sākotnējiem centriem. Neefektīva klasterizācija gadījumos, kad ir dažādi klasteru izmēri
Dziļās pārliecības tīkls (*Deep belief networks*)	Uzraudzīts	Augstas veikstspējas algoritms ar daudziem neironu slāņiem, tai skaitā slēptajiem slāņiem. Labi apstrādā diversificētus datus	Zombētu ierīcu identificēšana. Ļaunprātīgas IP un domēna identificēšana	Augstas apstrādes izmaksas. Liels apstrādes laiks daudzu slāņu dēļ. Nespēja pamatot lēmumu
Uz grafu balstīts tīkls (*Graph Convolutional Network*)	Daļēji uzraudzīts	Neironu tīklu arhitektūra darbam ar grafiem	Ļaunprātīga domēna identificēšana	Augstas apstrādes izmaksas, tai skaitā apmācības veikšanai. Liels apstrādes laiks. Nepieciešamas lielas datu kopas

Mašīnmācīšanās algoritms	Algoritma tips	Apraksts	Pielietojums	Trūkumi
				lai sasniegtu labu rezultātu

Ņemot vērā 2.10.tabulā sniegto izklāstu, kā arī to, ka autoram ir pieredze identificēt ļaunprātīgas ierīces darbību tīklā, autors izvēlējies darbā pielietot uzraudzītās mašīnmācīšanās metodes.

2.5 RQ5: RISINĀJUMI AUTOMATIZĒTAS ĻAUNPRĀTĪGAS AKTIVITĀTES DATORU TĪKLĀ APTURĒŠANAI

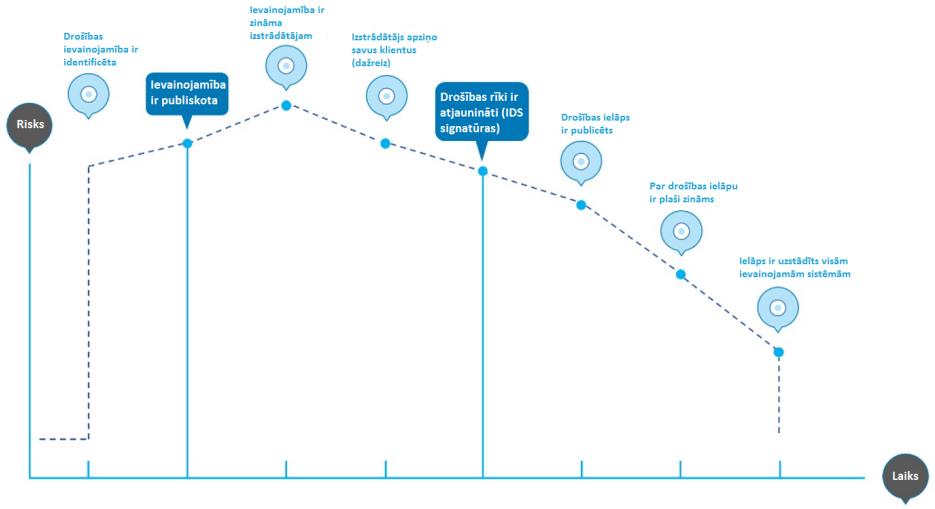
Šajā nodaļā ir pētīti pieejamie atvērtā koda rīki un risinājumi, kurus ir iespējams izmantot IS drošības pārvaldības nodrošināšanai, lai apturētu ļaunprātīgu aktivitāti datoru tīklā.

2.5.1 Ievainojamību uzraudzība

Saskaņā ar Kembridžas universitātes vārdnīcu [99] ievainojamība ir iespēja būt ietekmētam vai tikt pakļautam uzbrukumam. Ievainojamības informācijas sistēmās pastāvēja un pastāvēs vienmēr. Saskaņā ar kompānijas Rapid7 skaidrojumu [100], biežāk sastopamās sistēmu ievainojamības ir:

1) apdraudēta autentifikācija: ja tiek nozagti lietotāja akreditācijas dati, ļaundaris var uzdoties par lietotāju;
2) SQL injekcijas: viena no visizplatītākajām drošības ievainojamībām ar kuru palīdzību ļaundari mēģina piekļūt datu bāzes saturam, izmantojot netipiska koda ievadīšanu. Veiksmīga SQL injekcija var ļaut uzbrucējiem piekļūt sensitīviem datiem, viltot identitāti un veikt citas kaitīgas darbības datubāzē;
3) starpvietņu skriptēšana: Līdzīgi kā SQL injekcija, arī starpvietņu skriptēšanas (XSS) uzbrukuma gadījumā, vietnē tiek ievadīts ļaunprātīgs kods. XSS uzbrukums ir vērsts uz lietotājiem, kuri apmeklē vietni nevis uz pašu vietni. Veiksmīga uzbrukuma gadījumā zādzības riskam var tikt pakļauti lietotāju personu dati;
4) stapvietņu pieprasījumu viltojums: starpvietņu pieprasījumu viltošanas (CSRF) uzbrukuma mērķis ir maldināt autentificētu lietotāju veikt darbību, kuras viņi neplāno darīt. Šādā veidā ļaundaris var izmatot sociālo inženieriju un iegūt lietotāja personu datus;
5) nepareiza drošības konfigurācija: jebkura drošības sistēmas komponente, ko uzbrucēji var izmantot konfigurācijas kļūdas dēļ, var tikt uzskatīta par "nepareizu drošības konfigurāciju".

Ievainojamību dzīves cikls saskaņā ar OWASP [101] atkarībā no riska līmeņa ir iedalās 8 posmos (2.23.att.).



2.23.att. OWASP ievainojamību dzīves cikls (adaptēts no *[101]*)

Pirmais posms: ievainojamība tiek identificēta. Ievainojamības identificē izmantojot dažādas pārbaužu metodes, mēģinot izprovocēt informācijas sistēmas darbības apturēšanu, piemēram padodot programmas ievades laukā netipiskus simbolus, netipisku simbolu daudzumu kā arī veicot dažādas citas darbības ar ievades lauku. Šādas, tikko identificētas ievainojamības mēdz dēvēt par nulles dienas ievainojamībām. Nākamais posms: ievainojamība ir kļuvusi zināma plašākai sabiedrībai. Risks atkarīgs no ievainojamībai piešķirtās CVSS bīstamības pakāpes intervālā no 0-10, kur 0 – nav bīstama, bet 10 ļoti bīstama ievainojamība [102]. Vislielākais risks ir gadījumā, kad jau ir pagājis noteikts laiks un ļoti bīstama ievainojamība ir nonākusi līdz izstrādātājam. Risks sāk samazināties, kad ievainojamas informācijas sistēmas lietotāji ir par to ir informēti un ir spējīgi veikt darbības riska mazināšanai pat ja ievainojamības labojums vēl nav pieejams, piemērm, izolējot informācijas sistēmu. Visbeidzot risks ir samazināts līdz minimumam gadījumā, kad izstrādātājs ir veicis ievainojamības labojumu un tas tiek uzstādīts ievainojamai informācijas sistēmai. Ņemot vērā augstākminēto, lai mazinātu ļaunprātīgas aktivitātes iespējamību, nepieciešams izmantot ievainojamību pārbaudes rīkus, piemēram OpenVas [103], Nessus [104], Qualys [105] vai citus un veikt regulāru ievainojamību skenēšanu, tādējādi uzraugot informācijas sistēmas un preventīvi tās pasargājot no ļaunprātīgas izmantošanas un personu datu zuduma.

2.5.2 Reaģēšanas ātrums

Lai samazinātu ļaunprātīgas aktivitātes nodarīto kaitējumu, pēc iespējas ātrāk ir nepieciešams reaģēt uz to, bloķējot piekļuvi datoru tīklam vai konkrētai sistēmai. Viens no risinājumiem, kuru piedāvāja pētnieki Chkirbene et al. [106] bija Hibrīda anomāliju klasterizācijas metode. Metodes pamatā ir anomāliju noteikšanas metode, kas apvieno *"Sub-Space Clustering"* (SSC) un *"OneClass Support Vector Machine"* (OC*SVM*). Pētnieki ar eksperimentu palīdzību bija pierādījuši metodes efektivitāti. Jāatzīst, ka pētnieki izmantoja diezgan novecojušu publiski pieejamu datu avotu [107].

Citi pētnieki [108] piedāvāja hibrīda ietvaru, kurā tiek apvienota klasterizācija un ekspertu sistēmas. Lai identificētu ļaunprātīgu darbību tika piedāvāts izmantot ekspertu sistēmas, bet, lai identificētu vēl nezināmus ielaušanās gadījumus, izmantot k-means klasterizācijas metodi. Piedāvātais hibrīdais ietvars sastāvēja no divām fāzēm: tiešsaistes fāzes un bezsaistes fāzes. Tiešsaistes fāze bija ļaunprātīgas darbības tīklā identificēšanas metode, kas atbildīga par tīkla savienojumu datu salīdzināšanu ar ģenerētajiem modeļiem. Trauksme tika ģenerēta gadījumos kad bija konstatēta ļaunprātīga darbība. Bezsaistes fāzē tika veidota apmācības datu kopa, kura tika izmantota tiešsaistes fāzē. Līdzīgi kā iepriekš apskatītajā avotā, pētnieki izmantoja publiskus datu avotus.

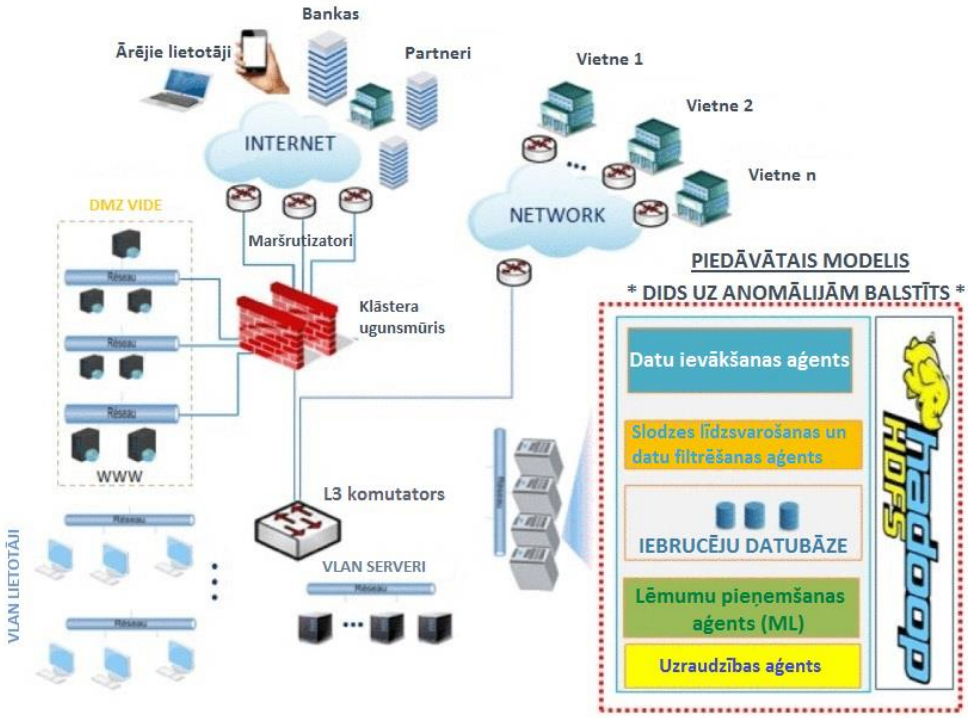
Ņemot vērā augstākminēto, var secināt, ka ir svarīgi maksimāli ātri reaģēt uz atklātajiem incidentiem un pēc iespējas ātrāk tos novērst. Šajā gadījumā var izmantot uz riskiem bāzētu pieeju, klasificējot informācijas sistēmas saskaņā ar MK442 [109] ņemot vērā to konfidencialitāti, integritāti un pieejamību un prioritāri mazināt drošības riskus paaugstinātas drošības sistēmām.

2.5.3 Lielo datu izmantošana

Tā kā datu apjoms pasaulē ar katru dienu tikai aug, lielie dati mūsdienās ir kļuvuši par neatņemamu drošības pārvaldības sastāvdaļu. Ar kiberdrošību saistītie dati mūsdienās ir pieejami ļoti lielā apjomā, kur datu apstrāde ar standarta rīkiem ir apgrūtinoša vai pat nav iespējama, tādēļ pētnieki [110] piedāvāja izmantot lielo datu konceptu (BIG DATA). Alguliyev et al. [111] ierosināja izmantot lielos datus IT drošības pārvaldībai, risinot tādus izaicinājumus kā pastāvīga draudu identificēšana, datu nonākšanas trešo personu rīcībā atklāšana, krāpšana un izlūkošana. Pētnieki izvirzīja vairākas problēmas lielo datu sakarā: privātums, uz lieliem datiem balstītu draudu identificēšanas algoritmu trūkums, drošības vizualizācijas trūkums un kvalificēta personāla trūkums. Autoruprāt [111] dati tiek definēti kā lielie dati gadījumos, kad:

1) datus nav iespējams apstrādāt ar standarta apstrādes rīkiem (piemēram, apstrāde ieilgs uz dienām vai nedēļām, vai arī datu ir tik daudz, ka tie neietilpst operētājsistēmas atmiņā);
2) dati tiek ražoti milzīgā ātrumā, kā arī to apstrādei jābūt tūlītējai (piemēram, maksājumu karšu ļaunprātīgas izmantošanas noteikšanas sistēmas).

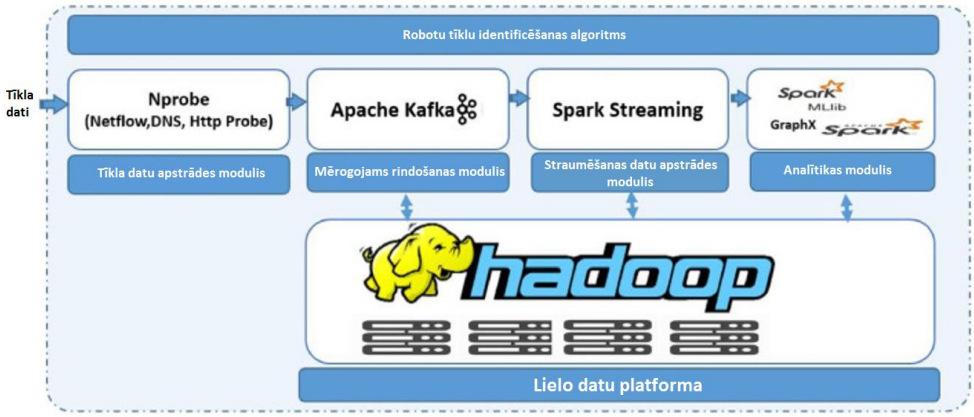
Šobrīd pastāv tādas platformas kā Apache Spark [112], ar kuras palīdzību ir iespējams apmācīt mašīnmācīšanās algoritmus lielo datu kontekstā. Lielos datus iespējams apstrādāt izmantojot paralēlo datu apstrādi, t.i. paralēli uz dažādām sistēmām tiek apstrādāti dati un apstrādes gaitā iegūtais rezultāts tiktu apkopots uz atsevišķas darbstacijas. Arī Ouiazzane et al. [46] piedāvāja izmantot Hadoop HDFS sistēmu paralēlai datu apstrādei anomāliju identificēšanai. Pētnieku piedāvātais modelis ir parādīts 2.24. attēlā.



2.24. att. Anomāliju identificēšana izmantojot lielos datus (adaptēts no *[46]*)

Pētnieki Mousavi et al. [110] piedāvāja izmantot Apache Kafka, Apache Spark atvērtā koda risinājumus (2.25.att.), lai veiktu tīkla datu apstrādi un analīzi robotu tīklu identificēšanai. Bez tīkla datiem pētnieki piedāvāja izmantot DNS datus, lai identificētu *DGA*. Datu kopas izveide sastāvēja no divām galvenajām daļām. Pirmajā daļā bija sagatavots zināmo robotu tīklu

komandas un kontroles centru (C&C) IP adrešu saraksts, otrajā – leģitīmu IP adrešu saraksts. Pirmajā daļā tika izmantoti dažādi bezmaksas IP melnie saraksti, baltais IP adrešu saraksts tika iegūts no populārajām IP adresēm, ņemot vērā Alexa top 500 [31] populārāko IP sarakstu, kā arī tika atlasīti 10 000 populārākie Cisco Umbrella [113] domēni. Lai no domēna vārda iegūtu IP sarakstu, tika izveidots un izpildīts skripts. Gadījuma mežu (*Random Forest*) algoritms tika pielietots, izmantojot SparkML [112] bibliotēku un sagatavoto datu kopu.



2.25. att. Uz lieliem datiem balstīta tīkla analīzes sistēma (adaptēts no *[110]*)

Pētniekiem izdevās apstrādāt 3 dienu datus 47.19GB apjomā 1299 sekundēs. Bez lielo datu apstrādēs sistēmu izmantošanas tas nebūtu iespējams.

2.5.4 Skriptošanas izmantošana

Kā minēts iepriekš (skat. 2.7.tabulu), uzturēt *SOC* ar 24X7 cilvēku pieejamību ir dārgi un ne visas organizācijas to var atļauties. Šajā gadījumā var izmantot skriptus, lai automatizētu ļaunprātīgas darbības bloķēšanas procesu. Mūsdienās gandrīz visi izstrādātāji saviem produktiem piedāvā aplikāciju programmēšanas interfeisu (API). Šis interfeiss ļauj sadarboties ar, piemēram, datoru tīkla nodrošināšanas servisu, liedzot inficētai ierīcei piekļuvi datoru tīklam. Piemēram, izmantojot Aruba Clearpass [114] un specifiski izstrādātus *Python* skriptus [115] ir iespējams bloķēt bezvadu tīkla piekļuves punktā specifisku MAC adresi. Līdzīgi iespējams veikt proaktīvas darbības bloķējot MAC adresi maršrutētājā, piemēram ar Hewlett Packard Rest API [116]. Arī ar ugunsmūriem iespējama komunikācija izmantojot REST API, piemēram Juniper ugunsmūrim [117], Palo Alto ugunsmūrim [118] u.c.

Pērnieki Garg et al [119] apgalvoja, ka mūsdienās API tiek īpaši plaši izmantots un pārsvarā visos API tiek izmantots JSON vai XML datu apmaiņas formāts, kuru viegli integrēt dažādās

aplikācijās. API padara komunkāciju drošāku, jo izmantojot autentifikāciju, iespējams novērst cilvēks pa vidu (MITM) uzbrukumus, kad komunikācija tiek pārtverta un ļaundaris veic turpmākās darbības ar informācijas sistēmu. Pētnieki norādīja, ka, lai nodrošinātu drošu API komunikāciju ar viena faktora autentifikāciju mūsdienās vairs nepietiek un nepieciešams vismaz vēl viens faktors, piemēram vienreiz ievadāmais kods, kas tiek atsūtīts īsziņas veidā.

2.5.5 Secinājumi

Savlaicīga ļaunprātīgas aktivitātes bloķēšana ir svarīgs aspekts drošības pārvaldības īstenošanai. Mūsdienās tiek pielietotas dadžādas metodes ļaunprātīga aktivitātes ierobežošanai: tas var būt gan API interfeiss, gan IP adrešu ievietošana specifiskā, ugunsmūrim paredzētā sarakstā, tas var būt ar syslog nosūtīts specifisks ziņojums, kā arī citi veidi. Gan autors, gan šajā apakšnodaļā apskatītie pētnieki atzīst, ka ļoti svarīgi ir ierobežot ļaunprātīgu aktivitāti pēc iespējas ātrāk, pirms nav nodarīts būtisks kaitējums infomācijas sistēmām un to datiem.

2.6 NODAĻAS KOPSAVILKUMS

Šajā nodaļā tika detalizēti analizēti 34 zinātniskās literatūras avoti, lielais vairums no kuriem koncentrēja uzmanību uz ļaunprātīgas darbības identificēšanu, izmantojot dažādas metodes, tai skaitā statistikas metodes un mašīnmācīšanās algoritmus. Pētnieki ļaunprātīgas darbības identificēšanai piedāvāja izmantot DNS datus, tīkla datus, tīkla metadatus, auditācijas pierakstus, "meduspodus" un *IDS*. Daudzi šajā nodaļā apskatītie pētnieki, izmantojot mašīnmācīšanās algoritmus ir sasnieguši labus rezultātus ļaunprātīgas darbības identificēšanā iegūstot precizitāti pat virs 99%, lai gan jāatzīmē, ka šie rezultāti tika sasniegti izmantojot mākslīgi ģenerētus datus, kas reālā vidē var neuzrādīt tik labus rezultātus.

2.11.tabula

Pētījumi, kuri saistīti ar kiberdrošības risku mazināšanu

Nr.p.k.	Pētnieku darbs	Lielo datu izmantošana	Auditācijas pierakstu izmantošana	IDS izmantošana	Meduspoda izmantošana	Tīkla datu analīze	DNS izmantošana	Mašīnmācīšanās metožu pielietojums	Automatizācijas skriptu izmatošana	Izmantotie mašīnmācīšanās algoritmi
1.	[40]		x	x		x			x	
2.	[44]					x		x		
3.	[46]	x	x	x					x	

Nr.p.k.	Pētnieku darbs	Lielo datu izmantošana	Auditācijas pierakstu izmantošana	IDS izmantošana	Meduspoda izmantošana	Tīkla datu analīze	DNS izmantošana	Mašīnmācīšanās metožu pielietojums	Automatizācijas skriptu izmatošana	Izmantotie mašīnmācīšanās algoritmi
4.	[47]			x		x		x		*DTC, K-NN, RFC, SVM*
5.	[48]			x				x		*NiB, NNC, DTC*
6.	[49]		x	x		x	x			
7.	[50]		x	x						
8.	[57]					x			x	
9.	[60]	x				x		x		*RFC, SVM, NiB*
10.	[64]			x		x		x		*NNC, Long Short-Term Memory Neural Network*
11.	[56]		x	x						
12.	[65]			x	x	x				
13.	[67]	x	x					x		*NiB, K-NN, One R, J48*
14.	[69]						x			
15.	[71]						x	x		*J48,DTC, RFC*
16.	[72]						x	x		*NiB, J48, RFC, K-NN, SVM*
17.	[73]			x			x	x		*RFC*
18.	[74]						x	x		*Logistic regression*
19.	[75]						x	x		*Graph*

Nr.p.k.	Pētnieku darbs	Lielo datu izmantošana	Auditācijas pierakstu izmantošana	IDS izmantošana	Meduspoda izmantošana	Tīkla datu analīze	DNS izmantošana	Mašīnmācīšanās metožu pielietojums	Automatizācijas skriptu izmatošana	Izmantotie mašīnmācīšanās algoritmi
20.	[76]						x			
21.	[77]						x	x		*NNC, Deep neural network*
22.	[83]		x	x				x		*Graph*
23.	[87]			x		x		x		*SVM, DTC, NiB, NNC etc.*
24.	[88]						x	x	x	*NiB, RFC, Logistic Regression*
25.	[90]						x	x		*NiB, SVM, DTC, RFC, Logistic Regression, NNC*
26.	[95]						x	x		*Graph*
27.	[85]					x		x		*Graph, Similarity*
28.	[97]					x		x		*K-NN, SVM*
29.	[98]			x		x	x	x		*OPTICS: Ordering points to identify the clustering structure ,G MM (Gaussian mixture models)*
30.	[106]		x	x		x		x		*RFC, DTC, SVM, "OneClass*

Nr.p.k.	Pētnieku darbs	Lielo datu izmantošana	Auditācijas pierakstu izmantošana	IDS izmantošana	Meduspoda izmantošana	Tīkla datu analīze	DNS izmantošana	Mašīnmācīšanās metožu pielietojums	Automatizācijas skriptu izmatošana	Izmantotie mašīnmācīšanās algoritmi
										Support Vector Machine" (OCSVM).
31.	[108]			x		x		x		*K-means, RFC*
32.	[110]	x				x	x	x	x	*RFC*
33.	[111]	x	x	x		x				
34.	[119]								x	

Analizētie pētījumi (2.11.tabula) lieto dažādas metodes, ļaunprātīgas aktivitātes identificēšanai organizācijas tīklā, kā arī veidus, kā efektīvi reaģēt uz atklāto incidentu. Viena no metodēm ir *SOC* sistēmu lietošana izmantojot *IDS*, kur tas var tikt pielietots gan atsevišķi, gan papildinot to ar mašīnmācīšanās komponenti. Diemžēl *SOC* izmaksas (skat. 2.7.tabulu) ir pārāk lielas lielākai daļai no Latvijā sastrādājošo organizāciju, tāpēc, lai mazinātu kiberdrošības riskus un ietaupītu organizāciju līdzekļus, darbā piedāvāts adaptīvs drošības pārvaldības modelis, kā arī veikta tā aprobācija, izstrādājot *ISMS* platformu. Liela daļa no apskatītajiem autoriem piedāvā izmantot lielo datu tehnoloģiju precīzākai un ātrākai incidenta identificēšanai. Pielietojot šīs tehnoloģijas ir iespējams apgādāt dažādas mašīnmācīšanās metodes ar lielāku apmācību datu kopu. Piemēram, robottīkla identificēšanai var pielietot *Apache Kafka* un *Apache Spark* lielo datu tehnoloģiju rīkus [110]. Daudzu pētnieku vidū DNS analīze ir ļoti populārs veids kā identificēt inficētu ierīci datortīklā [74] [73] [75] [76] [72] [71]. Apskatītie autori piedāvā izmantot dažādas pazīmes, bet populārākās ir domēna garums; vidējais "*unigram*"; vidējais "*bigram*"; vidējais "*trigram*"; "*unigram*" standartnovirze; "*bigram*" standartnovirze; "*trigram*" standartnovirze; patskaņu attiecība pret garumu; līdzskaņu attiecība pret garumu kā arī unikālo simbolu attiecība pret garumu. Tā kā mūsdienās pārsvarā visi tīkla datu savienojumi ir šifrēti, vairāki pētnieki [57] [60] piedāvāja tīkla datu analīzei izmantot tīkla metadatus. Galvenokārt kā pazīmes algoritma apmācībai tika izmantoti tādi parametri kā avota IP, mērķa IP, porti, kā arī protokoli un dažādas statistikas metodes, piemēram vidējais datu skaits, datu pakešu skaits to standartnovirze u.c.. Laicīga ievainojamību identificēšana ir viena

no drošības pārvaldības sastāvdaļām, ar kuras palīdzību organizācija var preventīvi samazināt drošības riskus labojot ievainojamu sistēmu pirms to izmantos ļaunprātīgas personas. Uzbrucējam pieejamās ievainojamās sistēmas ir viegls mērķis un parasti tiek "uzlauztas" pirmās. Arī robotu tīklu dalībnieki veic ievainojamu ierīču skenēšanu un ievainojamību izmantošanu ar mērķi inficēt ierīci un pievienot to robotu tīklam. Skriptošana ir svarīga sistēmas sastāvdaļa, lai padarītu procesus automatizētus, piemēram ierīces atslēgšanu no datortīkla ificēšanās gadījumā, tādēļ arī tā tika iekļauta *ISMS* platformā. 2.12.tabulā apkopotas dažādas pētnieku izvirzītās metodes ļaunprātīgā koda identificēšanai, kuras tika pielietotas *ISMS* platformā.

2.12.tabula

Biežāk izmatotās metodes informācijas sistēmu drošības nodrošināšanai

Metode	**Metodes priekšrocības**	**Metodes trūkumi**
Lielo datu koncepts (BIG DATA)	Var tikt apstrādāts liels datu apjoms, padarot apstrādes procesus paralēlus, un, apstrādājot lielāku datu apjomu var tikt sasniegts labāks rezultāts	Rada augstas veiktspējas prasības pret aparatūras un programmatūras risinājumiem
DNS analīze izmantojot mākslīgo intelektu	Var tikt identificēti robotu tīklu komandcentri, izmantojot metodes, kuras atšķir lietotāja ievadīto leģitīmo DNS adresi no mākslīgi ģenerēta DNS (*DGA*)	Ir pieejami rīki ar kuru palīdzību iespējams šifrēt DNS datus
NetFlow datu analīze izmantojot mākslīgo intelektu	Var tikt identificētas inficētas ierīces tīklā vērojot tīkla datus. Metode nav atkarīga no datu šifrēšanas tehnoloģiju pielietojuma, jo tiek izmantoti metadati	Ir grūti izveidot pazīmes algoritmu apmācībai, kā arī tīkla datos ir sarežģīti noteikt ļaunprātīgu aktivitāti

Ielaušanās noteikšanas sistēma (*IDS*)	Gadījumos, kad ir zināmas ielaušanās metodes (signatūras), diezgan precīzi nosaka inficētu ierīci tīklā	Uz statiskiem likumiem balstīta pieeja neļauj identificēt vēl nezināmus draudus
Skriptošanas izmantošana	Nav nepieciešamas lielas izmaksas nodrošinot cilvēkresursus 24x7, kuri pieņem lēmumus balstoties uz *SOC* informāciju. Gandrīz visi mūsdienu drošības rīki, piemēram ugunsmūris, marštutizatori u.c. atbalsta *REST API* tehnoloģiju, kura ļauj pielietot skriptus, piemēram, ļaunprātīgas aktivitātes tūlītējai bloķēšanai	Nepieciešama pilnīga pārliecība, ka bloķētā ierīce ir ļaunprātīga, tādēļ ir nepieciešama rūpīga izvērtēšana pirms pielietot bloķēšanas skriptus
Ievainojamību uzraudzība	Spēj laicīgi atklāt ievainojamību sistēmā pirms ļaundaris to nav paspējis izmatot	Līdz ievainojamību skenēšanas rīkiem informācija par ievainojamībām var nonākt novēloti un ievainojamība var nebūt laicīgi pamanīta

Literatūras apskats ļauj secināt, ka IS drošības pārvaldības nodrošināšanai jāietver daudzdimensionālu datu analīzi no plaša avotu klāsta. Eksistē arī liels skaits potenciāli izmantojamo draudu noteikšanas metožu, kuras būtu nepieciešams implementēt kā neatkarīgus draudu identifikācijas servisus. Visu pieejamo datu avotu un draudu noteikšanas moduļu efektīva izmantošana vienotā drošības pārvaldības risinājumā, kas būtu pielāgojams organizācijas kontekstam, ir problēma, kas zinātniskajā literatūrā nav pietiekami pētīta un kam autors paredz pievērst īpašu uzmanību, izstrādājot no konteksta atkarīgu, adaptīvu drošības

pārvaldības modeli, kurš balstās uz spējo metodoloģiju un tā tehnisko realizāciju – *ISMS* platformu. Vispārīgais un specifiskais uz spējo metodoloģiju balstītais modelis un tā tehniskā realizācija – *ISMS* platforma tika izstrādāta un tās moduļi tika papildināti ņemot vērā 2.12.tabulā minētās prasības, kuras izriet no zinātniskās literatūras analīzes (2.11.tabula). *ISMS* platformā tika īstenota uz scenārijiem bāzēta automatizēta lēmumu pieņemšana un reaģēšana uz tiem, kā arī multidimensionāla pieeja draudu identificēšanai, apvienojot dažādu moduļu ģenerētos paziņojumus.

3 SPĒJORIENTĒTĀ DROŠĪBAS PĀRVALDĪBA

Spējorientētā drošības pārvaldība ir mērķu, spējas, konteksta elementu un mērāmo rādītāju apkopojums. Darbā izstrādātais spējorientētais drošības pārvaldības modelis ir balstīts uz spējo izstrādi (*CDD*) [32]. Tas spēj ņemt vērā pilnu konteksta informāciju, kā arī veikt automātiskas adaptīvas darbības drošības līmeņa atjaunošanai draudu gadījumā.

Spējorientēto drošības pārvaldību raksturo:

- Spēja;
- Datu avoti;
- Datos balstīta pielāgošana;
- Automatizēta reaģēšna;
- Modulāra uzbūve;
- Mērāmie atribūti.

3.1 SPĒJORIENTĒTĀS IZSTRĀDES METODOLOĢIJAS PĀRSKATS

Spējorientētais drošības pārvaldības modelis balstās uz spējorientēto izstrādes metodoloģiju [32]. Metodoloģijas pamatkoncepti ir – spēja, konteksts, mērāmais atribūts. Spēja ir koncepts, kas tiek izmantots dažādos kontekstos un variācijās un var tikt definēta kā:

1) nemateriāla;
2) nepilnīga;
3) stabila laika gaitā;
4) kā neatkarīgs process, pat ja tas tiek ietekmēts;
5) hierarhiska un kombinējama;
6) spēcīgi ietekmējama no cilvēkresursiem (piemēram, personāla, apmācības);
7) attiecināma uz skaidri noteiktiem pienākumiem uzņēmumā;
8) biznesa vērtības nodrošināšana.

Spēju klasifikācija ir apskatīta

3.*1*.tabulā, un norāda, ka fokusējoties uz stratēģisku konkurences priekšrocību, liela daļa klasificēto spēju veidu ir saistītas ar IT.

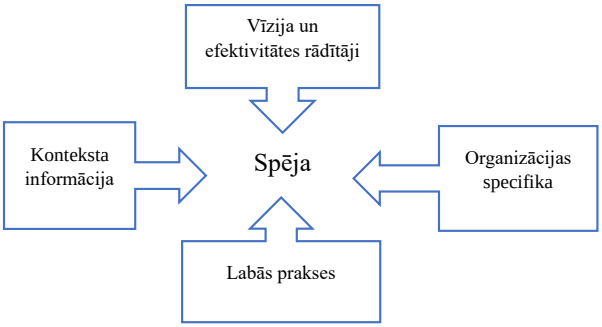
3.1.tabula

Spēju veidu klasifikācija (adaptēts no *[32]*)

Veids	Definīcija	Piemēri

Biznesa spēja	Uzņēmuma spēja saprast vispārējo biznesu vidi un specifiko organizatorisko kontekstu	Organizācijas specifiskas zināšanas un spēja apgūt biznesa funkcijas
Tehniskā spēja	Organizācijas tehniskās spējas, ņemot vērā to specifikās zināšanas tehniskajās jomās	Datu bāžu pārvaldība, kompetences jaunajās tehnoloģijas
Uzvedības spēja	Uzņēmuma personu un tā vadība spēja mijiedarboties savā starpā un vadīt citus	Efektīva personu komunikācija un sadarbība plānojot un vadot projektus
Infrastruktūras spēja	IT organizācijas spēja sniegt IT infrastruktūras pakalpojumus biznesa procesu nodrošināšanai	Sakaru pakalpojumi, datu uzturēšanas pakalpojumi, IT vadības pakalpojumi
IT sistēmu briedums	Spēja veikt izmaiņas informācijas sistēmās neradot būtiskas soda sankcijas laika vai izmaksu kategorijās	Izmaiņu vai uzlabošanas veikšanas izmaksu samazināšana, kā arī ātrāka sistēmu izstrāde
Uz IT sistēmām balstītas informācijas briedums	Spēja vienkāršoti veikt izmaiņas kā lietotāji piekļūst un lieto informācijas resursus	Ātrāka informācijas iegūšana, palielinot informācijas pieprasījumu elastību
Uz IT sistēmām balstīts stratēģiskais briedums	Spēja efektīvi reaģēt uz jaunām tirgus sniegtajām iespējām, izmantojot jau esošās IT iespējas	Ātrāka reakcija uz tirgus izmaiņām, iegūstot konkurētspējas priekšrocības

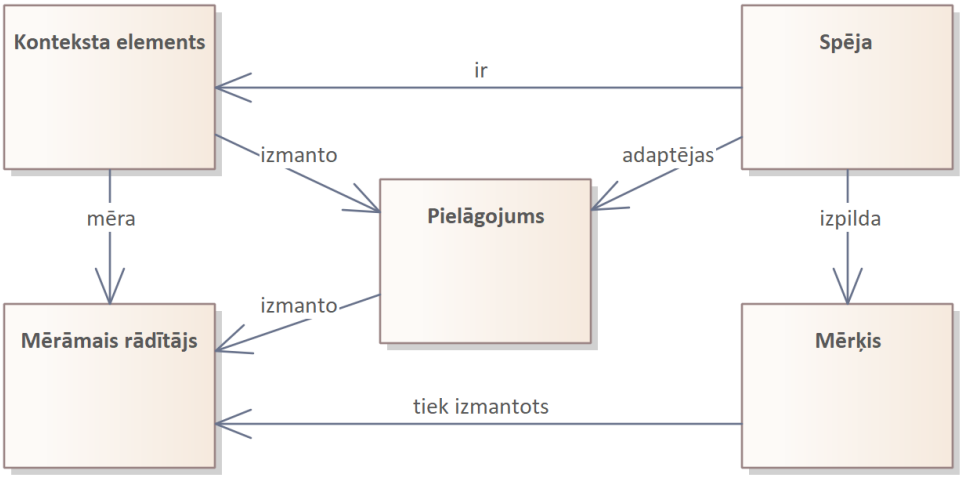
CDD metodes izmantošana palīdz palielināt uzņēmumu produktivitāti un elastību. Mūsdienās spēju pielietošanas potenciāls nav pietiekami izmantots. *CDD* var tikt pielietota biznesa modeļu, to servisu, kā arī IT, kā biznesa procesu atbalsta izstrādē. *CDD* tiek uztverta kā iemaņas, varēšana un pietiekamu resursu esamība, jeb kapacitāte, lai šo spēju varētu realizēt. Tas ietver kompetenci, kur kompetence tiek saprasta ar zināšanām, talantu un spēju to pielietot, lai sasniegtu mērķi. Kapacitāte nozīmē resursu pieejamību, tādu kā nauda, laiks, personāls un rīki. Jāņem vērā, ka kapacitāte kā resursu pieejamība ir spējas sastāvdaļa un spēja vienmēr tiek nodrošināta noteiktā kontekstā.



3.1. att. Spējas pamataspekti (adaptēts no *[32]*)

Starp daudzajiem iemesliem, kāpēc spēju pārvaldība ir aktuāla tēma un spējā domāšana kā princips saņem arvien lielāku uzmanību, ir tas, ka uzņēmumiem jāspēj ātri pielāgoties izmaiņām ekonomiskajā un normatīvajā vidē. *CDD* ir cieši saistīta biznesa pakalpojumu konteksta un variantu izpratni. Spēju pārvaldība ietver konteksta identifikācijas mehānismu un biznesa pakalpojumu variantus šajā kontekstā, kā arī adaptāciju mehānismu izstrādi. Spējas pamata aspekti ir demonstrēti 3.1.attēlā [32].

Spējorientētās izstrādes pamatkoncepti, kuri attiecas uz drošības pārvaldību ir parādīti 3.2.attēlā. Izstrādājamajam drošības pārvaldības modelim ir specifiski mērķi, kuru izpildei tiek noteikti mērāmie rādītāji, kā arī konteksta elementi, kuri arī tiek mērīti, izmantojot mērāmos rādītājus, spēja pielāgojas, un pielāgojums izmanto gan konteksta elementus, gan mērāmos rādītājus.



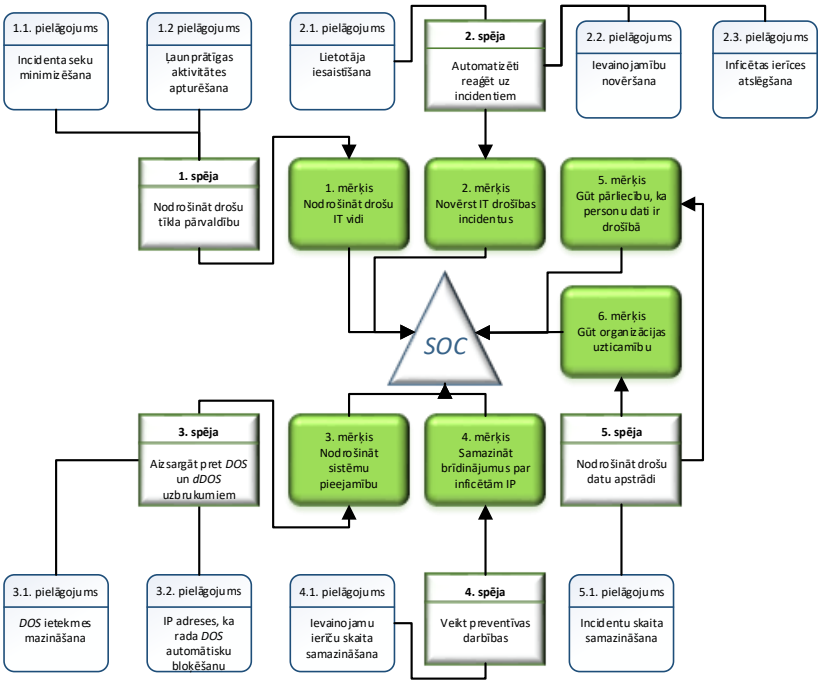
3.2.att. Drošības pārvalības sistēmas modeļa spējas pamataspekti

Izmantojot augstākminēto metodoloģiju ir iespējams aprakstīt drošības pārvaldības sistēmas konteksta elementus, mērāmos rādītājus un spējas, tādā veidā skaidri definēt tās darbības principus.

Bez tam, izmantojot *CDD* ir iespējams definēt pielāgošanas modeļus dažādu izmaiņu gadījumā. Drošības pārvaldības platformas modeļa izstrādei tika pielietots *CDD*, jo tas ietver nepieciešamo konteksta identifikācijas mehānismu, kā arī spēju adaptēties veidojot pielāgojumus.

3.2 VISPĀRĪGĀ IS DROŠĪBAS PĀRVALDĪBAS SPĒJA

Lai organizācijā ieviestu spējorientētu drošības pārvaldības modeli, ir nepieciešams izstrādāt tai atbilstošos spēju modeļus. Šajā nodaļā ir izveidots abstrakts spējorientētas drošības pārvaldības modelis, kurš varētu derēt jebkurai organizācijai. Modeli nepieciešams izvērst atbilstoši konkrētās organizācijas specifikai.



3.3.att. Informācijas drošības pārvaldības spēja

Informācijas drošības pārvaldības spēja tiek īstenota, lai sasniegtu dažādus nospraustos mērķus drošības pārvaldības uzlabošanai. Šī spēja izpilda galveno mērķi 1.Nodrošināt drošu IT vidi, kas tiek atbalstīta ar pieciem papildu mērķiem: 2.Novērst IT drošības incidentus, 3.Nodrošināt sistēmu pieejamību, 4.Samazināt brīdinājumus par inficētām IP adresēm, 5.Gūt pārliecību, ka personu dati ir drošībā, 6.Gūt organizācijas uzticamību. Katram mērķim ir definēta spēja un pielāgojumi ar kuru palīzību šī spēja tiek īstenota. 1.mērķim ir definēta spēja nodrošināt drošu tīkla pārvaldību, kur šai spējai ir divi pielāgojumi: Pielāgojums1.1. Incidentu seku minimizēšana un Pielāgojums 1.2. Ļaunprātīgas aktivitātes apturēšana. 2.mērķim ir noteikta spēja automatizēti reaģēt uz incidentiem un tās īstenošanai nepieciešami trīs pielāgojumi: Pielāgojums 2.1. Lietotāja iesaistīšana inicdenta risināšanā, Pielāgojums 2.2. Ievainojamību novēršana un Pielāgojums 2.3. Inficētas ierīces atslēgšana. 3.mērķis ir nodrošināt sitēmu pieejamību, kas tiek īstenots ar spēju aizargāt informācijas sistēmas pret *DOS* un *dDOS* uzbrukumiem, kas savukārt tiek īstenots, izmantojot Pielāgojumu 3.1 *DOS* un *dDOS* ietekmes mazināšana un Pielāgojumu 3.2. IP adreses, kura veic uzbrukumu, bloķēšana. 4.mērķim ir definēta spēja veikt preventīvas darbības un Pielāgojums 4.1., ir Ievainojamu ierīču skaita samazināšana. 5. un 6.mērķiem ir spēja nodrošināt drošu datu apstrādi, kā arī pielāgojums 5.1 Incidentu skaita samazināšana.

Informācijas drošības pārvaldības spēju modeļa elementu apraksts ir parādīts 3.2.tabulā.

3.2.tabula

CDD modeļa elementu apraksts

Elementa klase	**Elementa numurs**	**Elementa nosaukums**	**Elementa apraksts**
Mērķis	1	Nodrošināt drošu IT vidi	Mērķis ir nodrošināt drošu IT vidi, kur notiek drošības uzraudzība un ļaunprātīga aktivitāte no ierīcu puses tiek laicīgi identificēta
	2	Novērst IT drošības incidentus	Šis mērķis paredz operatīvu incidentu novēršanu veicot automatizētu incidentu apstrādi un reaģēšanu uz tiem
	3	Nodrošināt sistēmu pieejamību	Viena no svarīgākajām drošības sastāvdaļām ir pieejamība. Mērķis ir, lai orgainzācijas IT sistēmas būtu pieejamas un pieejas atteices uzbrukumi tās nevarētu ietekmēt
	4	Samazināt brīdinājumus par inficētām IP	Šis mērķis paredz to, ka organizācija nenonāk dažādu drošības kompāniju "melnajos sarakstos", kas turpmāk apgrūtina komunikāciju, kā arī ietekmē organizācijas reputāciju
	5	Gūt pārliecību, ka personu dati ir drošībā	Mērķis ir pārliecināt iekšējos un ārējos auditorus, kā arī organizācijas vadību, ka personu dati tiek apstrādāti drošā veidā
	6	Gūt organizācijas uzticamību	Mērķis ir pārliecināt ārējos klientus, ka organizācija ir uzticama un tai drīkst sniegt savus datus apstrādei
Spēja	1	Nodrošināt drošu tīkla pārvaldību	Šī spēja paredz veikt darbības, lai nodrošinātu drošu tīkla pārvaldību
	2	Automatizēti reaģēt uz incidentiem	Automatizēta reakcija uz incidentiem nozīmē to, ka drošības personāla iesaiste incidentu izmeklēšanā un reaģēšanā uz tiem ir minimāla
	3	Aizsargāties pret *DOS* un *dDOS* uzbrukumiem	Spēja identificēt un reaģēt uz piekļuves atteices uzbrukumiem

	4	Veikt preventīvas darbības	Spēja identificēt ievainojamības un veikt ievainojamību labojumus
	5	Nodrošināt drošu datu apstrādi	Šī spēja paredz izmantot metodes un rīkus, lai pierādītu, ka datu apstrāde organizācijā notiek droši
Pielāgojums	1.1	Incidentu seku minimizēšana	Pielāgojums paredz pietiekamu daudzumu ar auditācijas pierakstiem, lai spētu izmeklēt incidentus, kā plānus kā maksimāli īsā termiņā atgūties no incidenta, piemēram, izmantojot rezerves kopijas.
	1.2	Ļaunprātīgas aktivitātes apturēšana	Lai īstenotu šo pielāgojumu nepieciešams apstrādāt dažādus datu avotus, kā arī jāspēj identificēt ļaunprātīgu darbību tīklā
	2.1	Lietotāja iesaistīšana	Dažādu metožu izmantošana lai informētu lietotājus, piemēram SMS, iekšējais portāls, e-pasts.
	2.2	Ievainojamību novēršana	Identificējot ievainojamību ierīcē, nepieciešams savlaicīgs šīs ierīces ievinojamības labojums
	2.3	Inficētas ierīces atslēgšana	Nepieciešams ātri reaģēt un atslēgt ierīci no tīkla, ja tiek konstatēta tās augsta riska inficēšanās, kas var novērst pie citu ierīču kompromitēšanas, piemēram ierīce veic ievainojamību identificēšanu, portu skanēšanu un paroļu minēšanu
	3.1	Dos ietekmes mazināšana	Iespēja identificēt un reaģēt uz pieejas atteices uzbrukumiem samazinot to ietekmi
	3.2	IP adreses, kura rada *DOS* automātiska bloķēšana	Iespēja bloķēt IP adreses, kuras veic pieejas atteices uzbrukumus

	4.1	Ievainojamu ierīču skaita samazināšna	Preventīva darbība, identificējot un salabojot ievainojamas ierīces pirms ļaundaris tās izmanto
	5.1	Incidentu skaita samazināšana	Ja incidents netiek novēsts laicīgi, pastāv varbūtība, ka tas radīs gan finansiālu gan reputācijas ieteikmi organizācijai

3.3 PRASĪBAS TEHNISKAJAM RISINĀJUMAM

Nodaļā ir apkopotas prasības tehniskajam risinājumam (ISMS platformai), kas nodrošina IS drošības pārvaldību atbilstoši iepriekš definētajam spēju modelim iekļaujot trīs pamatkomonentes: 1) datu avoti, 2) datu analīze, 3) darbība, jeb reaģēšana. Prasības ir definētas atbilstoši literatūrā apskatītajam un ir attiecināmas uz jebkuru organizāciju, kura izvēlas ieviest spējorientētu IS drošības pārvaldību.

3.3.tabula

Prasību definēšana *ISMS* platformai

Nr.p.k.	Prasība	Sasniedzamais rezultāts
1.	Izmantot atvērtā koda tehnoloģijas un augsta līmeņa programmēšanas valodu	Iegūt uz atvērto kodu balstītu produktu, kurš nav atkarīgs no viena izstrādātāja
2.	Identificēt ierīci, kā arī tās pieslēguma vietu, arī gadījumos, kad tiek izmantotas dinamiskās IP adreses	Līdz minimumam samazināta ierīces viltus pozitīvas identificēšanas iespēja
3.	Identificēt lietotāju un iesaistīt viņu drošības pārvaldības procesā	Pēc iespējas ātrāka drošības incidenta novēršana
4.	Izmantot mērogojamu risinājumu, lai gadījumā, kad nepieciešams apstrādāt lielāku datu apjomu, to varētu izdarīt bez platformas pārbūves	Spēt tikt galā ar pieaugošiem datu apjomiem
5.	Izmantot uz atvērto kodu bāzētu ielaušanās noteikšanas sistēmu ar signatūru papildināšanas iespēju	Identificēt ļaunprātīgas darbības tīkla datos
6.	Izmantot auditācijas pierakstu analīzi ļaunprātīgas darbības identificēšanai	Identificēt ļaunprātīgas darbības auditācijas pierakstos

Nr.p.k.	Prasība	Sasniedzamais rezultāts
7.	Izmantot uz atvērto kodu bāzētu tīkla datu analīzes mehānismu	Identificēt ļaunprātīgas darbības tīkla datos, piemēram portu skanēšanu
8.	Izmantot uz atvērto kodu bāzētu tīkla metadatu analīzes mehānismu	Iespēja identificēt anomālijas tīkla metadatos (*NetFlow*)
9.	Izmantot "meduspoda" funkcionalitāti, lai gūtu iespēju padziļināti izpētīt ļauprātīgā koda vai cilvēka darbību	Gūt jaunu informāciju par ļaunprātīgā koda vai ļaundara izpildītajiem pieprasījumiem ar mērķi nākotnē automatizēt šādas darbības atklāšanu
10.	Identificēt lietotāja autentifikācijas datu zādzību	Novērst lietotāja autentifikācijas datu zādzības sekas, tai skaitā privātuma ietekmēšanu
11.	Identificēt ļaunprātīgu darbību tīklā, izmantojot tīkla metadatus un atvērtā koda bāzētas sistēmas, kā arī izmantot mašīnmācīšanos	Identificēt ļaunprātīgu darbību (tai skaitā nulles dienas uzbrukumus) tīklā, izmantojot tīkla metadatus
12.	Identificēt portu skanēšanu un paroļu minēšanu iekšējā tīklā	Identificēt inficētu ierīci vai ļaunprātīgu darbību tīklā
13.	Identificēt un uz noteiktu laiku bloķēt ārējas IP adreses, kuras nav iesaistītas nevienā komunikācijā – tikai veic portu skanēšanu	Lai uzbrucējs nebūtu spējīgs identificēt iespējamu ievainojamību, uz laiku bloķēt IP adreses, kuras veic iestādes IP adrešu portu skanēšanu
14.	Izmantot adaptīvu pieeju dažādu draudu gadījumā. Piemēram specifiska draudu gadījumā reaģēt ātrāk kā cita drauda gadījumā	Atšķirīga automātiskā platformas reakcija uz dažādiem drošības incidentiem, ņemot vērā to nopietnību
15.	Izmantot automatizētu ievainojamību identificēšanu un automātisku ziņošanu atbildīgajai personai	Automātiski un periodiski informēt atbildīgās personas par augstām un kritiskām ievainojamībām viņu sistēmās, lai pēc iespējas ātrāk tās tiktu novērstas

Nr.p.k.	Prasība	Sasniedzamais rezultāts
16.	Identificēt agoritmiski ģenerētus domēnus DNS informācijā, izmantojot statiskos sarakstus un mašīnmācīšanos	Identificēt ierīces, kuras atrodas robotu tīklā
17.	Atslēgt ierīces, kuras rada apdraudējumu iekšējā tīklā	Izolēt ierīces, kuras rada apdraudējumu iekšējā tīklā, vai ierīces, kuru lietotāji nereaģē uz brīdinājumiem
18.	Parādīt atklātos incidentus un nosūtītos paziņojumus par drošību atbildīgajai personai, izmantojot grafisko interfeisu	Spēja pārskatīt drošības pārvaldības platformas darbību ērtā, viegli saprotamā veidā izmantojot grafisko interfeisu

Darbā tika īstenotas 3.3.tabulā minētās prasības un izstrādāts *ISMS* platformas modelis, kā arī nepieciešamās komponentes (moduļi) efektīvai platformas darbībai. 3.4. tabulā minētās prasības ir pamatotas ar 2 nodaļā minēto literatūras izpēti (sk.2.1.tabulu).

3.4.tabula

Tabulā 3.3. minētā prasība	Pamatojums
1	RQ1, nodaļa 2.1
2, 3,9,13,15,17	RQ5, nodaļa 2.5
4	RQ5, nodaļa 2.5.3
5	RQ2 nodaļa 2.2.2
6	RQ2 nodaļa 2.2.3
7,8,12	RQ2 nodaļa 2.2.1
10,18	RQ1, nodaļa 2.1
14	RQ3, nodaļa 2.3
11, 16	RQ4, nodaļa 2.4

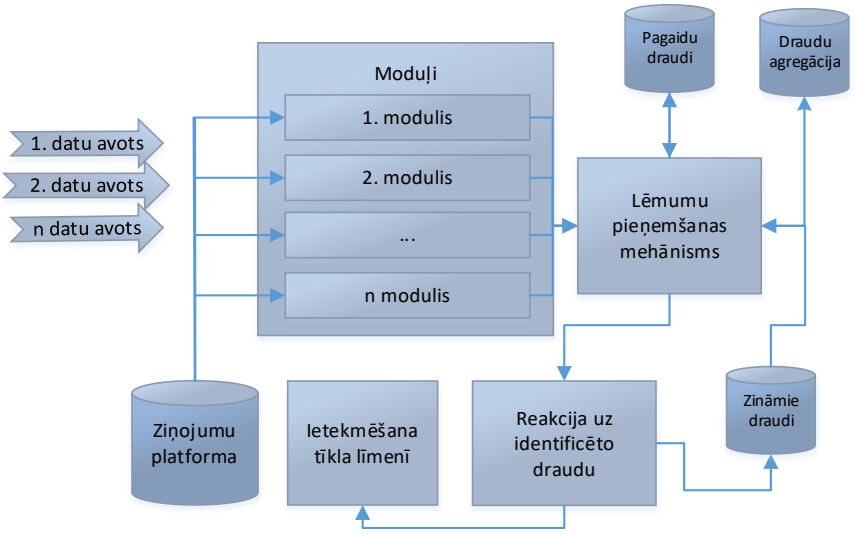
3.4 TEHNISKĀ RISINĀJUMA AUGSTA LĪMEŅA ARHITEKTŪRA

Lai ieviestu iepriekšējā nodaļā minētās prasības organizācijai nepieciešams nodrošināt:

1) Iespēju saņemt datus no dažādiem datu avotiem, kā minēts 2.2. nodaļā, kā arī saprast tīkla protokolus;

2) *Python* vai citas augsta līmeņa skriptošanas valodas pārzināšanu, lai spētu adaptēt automatizācijas procesus;
3) Mācēt izmantot atvērtā koda programmatūru, tai skaitā uz lielajiem datiem bāzētu programmatūru;
4) Izmantot statiskas IP adreses iekšējā tīklā, vai, izmantojot DHCP, spēt identificēt dinamiski piešķirtās IP adreses;
5) Mācēt identificēt gala lietotāju pēc IP adreses, lai spētu viņu iesaistīt drošības incidenta seku mazināšanā;
6) Mācēt izmantot atvērtā koda vai maksas risinājumu, lai identificētu ievainojamības;
7) Mācēt lietot atvērtā koda ielaušanās noteikšanas sistēmas;
8) Mācēt uzkrāt un lietot Netflow datus;
9) Mācēt izmantot atvērtā koda vai maksas "meduspoda" risinājumus;
10) Mācēt strādāt ar M365 datiem, tai skaitā GraphAPI;
11) Saprast API darbību un mācēt to pielietot gan veicot ierīces atslēgšanu no tīkla, gan saņemot un nosūtot datus no/uz dažādām informācijas sistēmām;
12) Mācēt izmantot ugunsmūra datus;
13) Spēt novērtēt, kuros drošības incidenta gadījumos reaģēt ir nepieciešams ātrāk un kuri gadījumi var gaidīt;
14) Saprast kā notiek DNS izsaukums un mācēt tos analizēt;
15) Mācēt koriģēt drošības pārvaldības sistēmas grafisko saskarni.

ISMS platforma nodrošina spējas sasniegšanu izmantojot komponentes, kuras attēlotas 3.4.att. Informācija no dažādiem avotiem, piemēram, tīkla, ugunmūra auditācijas pierakstu dati tiek nodoti ziņojumu apstrādes datubāzei.



3.4. att. Adaptīvā drošības pārvaldības modeļa komponentes

Turpmāk dažādi datu apstrādēs moduļi apstrādā ienākošos datus un nodod informāciju lēmumu pieņemšanas mehānismam, kurš pieņem lēmumu, vai ziņojums var tikt uzskatīts par draudu. Lēmuma pieņemšanai tiek izmantoti jau zināmie draudi, kā arī draudi, kas tiek noteikti izmantojot dažādas mašīnmācīšanās metodes. Ja lēmumu pieņemšanas mehānisms ir identificējis ziņojumu kā draudu, tiek identificēta reakcija uz to. Reakcija uz draudu var būt gan atbildīgā par ierīci lietotāja apziņošan izmantojot dažādus apziņošanas mehānismus, gan ierīces atslēgšana no tīkla gadījumā, ja lietotājs nav reaģējis uz brīdinājumiem un novērsis problēmu. Bez tam, lietotājam ir iespēja sniegt atgriezeniesko saiti, kura tiek ekspertu analizēta un tiek pieņemts lēmums par turpāko līdzīgo ziņojumu iekļaušanu zināmo ziņojumu datubāzē vai izņemšanu no tās. Darbā tika izstrādāti divi specializēti uz mašīnmācīšanos balstīti moduļi: *DGA* identificēšanas un ļaunprātīgas darbības tīkla datos identificēšanas modulis.

Platformas pamatkomponentes balstās uz šādām tehnoloģijām:

- *Python*3 – saskaņā ar IEEE pētījumu [120], 2023.gadā tā ir atzīta par polulārāko programmēšanas valodu, kā arī tā ir viegli lasāma un satur daudz izstrādātu moduļu.
- Influx DB – populāra laikrindu datu bāze, kas piedāvā vairākas priekšrocības, jo īpaši gadījumos, kad laikrindu datiem ir izšķiroša nozīme. Galvenais mērķis šī darba ietvaros ir izmantot Influx DB, lai attēlotu informāciju Grafana. Papildu mērķis ir pielietot Influx DB, lai īstenotu datu agregācijas moduļa darbību, analizējot datus noteiktos laika intervālos.

- Grafana – populāra atvērtā pirmkoda platforma datu uzraudzībai, kas pazīstama ar savām elastīgajām un jaudīgajām vizualizācijas iespējām.
- Apache Kafka – izcili piemērota moderna straumēšanas platforma, ko plaši izmanto reāllaika datu plūsmu izveidei. Darbā tiek izmantota Apache Kafka programmatūra, jo tā ir bāzēta uz lielo datu paradigmu un horizontāli mērogojama, tai ir iebūvēts kļūdu novēršanas mehānisms, replicējot datus uz dažādiem datu brokeriem, tā var nodrošināt reāllaika datu plūsmu, operatīvi reaģējot uz jauniem ziņojumiem, tā bieži tiek pielietota notikumu uzraudzībai, krāpšanas gadījumu identificēšanai, kā arī reāllaika lēmumu pieņemšanai.
- Apache Spark – izplatīta un jaudīga atvērtā koda skaitļošanas sistēma, kas nodrošina ātru lielo datu apstrādi. Apache Spark ātrums, lietošanas vienkāršība, vienotās datu apstrādes iespējas un plašā ekosistēma padara to par populāru izvēli organizācijām, kas nodarbojas ar lielo datu apstrādes un analītikas uzdevumiem.
- Nfdump – vērtīgs rīks *NetFlow* datu vākšanai un analīzei, sniedzot tīkla administratoriem un drošības speciālistiem ieskatu tīkla datos un atvieglojot dažādus ar tīkla uzraudzību un drošības analīzi saistītus uzdevumus.
- Fprobe – efektīvs *NetFlow* eksportētājs, ko izmanto tīkla datu plūsmas uzraudzībai. Tam ir izšķiroša nozīme, sniedzot ieskatu tīkla darbībā, atvieglojot tīkla pārvaldību.
- Tcpdump – jaudīgs un daudzpusīgs pakešu analizators, ko parasti izmanto tīkla problēmu novēršanai, protokolu analīzei un drošības izmeklēšanai. Tas spēj uztvert un analizēt paketes reāllaikā, kas padara to par nozīmīgu rīku tīkla administratoriem, drošības speciālistiem, kuri strādā ar tīkla protokoliem.
- Scikit-Learn – daudzpusīga un lietotājam draudzīga *Python* mašīnmācīšanās bibliotēka, kas piemērota plašam uzdevumu klāstam, sākot no vienkāršas datu analīzes līdz sarežģītai mašīnmācīšanās modelēšanai. Tās API, plašais algoritmu komplekts un integrācija ar citām *Python* bibliotēkām veicina tās popularitāti datu zinātnieku un mašīnmācīšanās praktiķu vidū.

Tehnoloģiju ietvars tiek paplašināts atbilstoši iegūtajai informācijas organizācijas IS drošības pārvaldības spējas modelēšanas laikā. Darbā Apache Kafka un Influx DB ļāva nodrošināt lielapjoma statisku un reāllaika datu integrāciju savukārt Apache Spark nodrošina lielo datu apstrādi, identificējot ļaunprātīgu darbību tīkla datos.

Galvenie ieguvumi piedāvātajā *ISMS* platformā ir:

- Galaiekārtu kontroles nodrošināšana;
- Dinamiska tīkla datu analīze;
- Procesu automatizācija;
- Adaptīva reaģēšana;
- Moduļu pievienoša, gadījumos, kad mainās uzbrukumu vektori;
- Atgriezeniskās saites nodrošināšana, automatizēta reaģēšana un ātra izmaiņu ieviešana sistēmā;
- Atvērtā koda izmantošana platformas būvēšanā.

Ļaunprātīgas darbības identificēšanai var tikt izmantoti dažādi datu avoti vienlaicīgi, piemēram, lai noteiktu vai ierīce ir inficēta var izmantot Influx datubāzi, kurā:

1) tiek uzkrāta ierīces iegūtā IP adrese, kā arī tās MAC adrese;
2) tiek uzkrāti notikumi (piemēram identificētie *DGA,* TOR tīkla izmantošana u.c.), kuri varētu netieši norādīt uz inficēšanās pazīmēm.

Notikumi Influx datubāzē var tikt analizēti noteiktā laika intervālā, piemēram ik minūti, pārskatot iepriekšējo triju stundu intervālu. Kritērijus, kas varētu liecināt par ierīces iespējamo inficēšanos var noteikt eksperti, ņemot vērā šādus aspektus:

a) Ienākošā ziņojuma kritiskumu dažādās kategorijās, ņemot vērā ziņojuma avotu, piemēram (*'informational', 'low', 'medium', 'high', 'critical'*) vai (1,2,3,4);
b) Ienākošā ziņojuma saturu, piemēram *'ML_DNS', 'DGA', 'generic', 'Grayware', 'Parked', 'Proxy', 'new', 'Suspicious DNS Query'.*

Visos gadījumos ir iespējams saskaitīt gan unikālus ziņojumus, gan arī atsevišķu ziņojumu skaitu, kā arī ziņojumu skaitu kopā.

Piemēram, eksperts var noteikt, ka, ja trīs stundu laikā tiek konstatēti trīs dažādi *DGA* paziņojumi, un papildu tam, ir kādas citas kategorijas ziņojums par noteikto ierīci, tad tas var tikt uzskatīts par ticamu ierīces inficēšanās pazīmi. Šāda pazīme var ģenerēt attiecīgu ziņojumu un nosūtīt to uz Apache Kafka. Sākotnēji šādi ziņojumi var tikt klasificēti kā zema riska ziņojumi, kur paziņojums lietotājam būtu nosūtāms tikai e-pastā. Ja turpmāk šādi ziņojumi atkārtotos, tad to riska vērtība pieaugtu un turpmāk lietotājs varētu saņemt paziņojumu gan portātā, gan izmantojot SMS, kā arī lietotāja ierīce var tikt atslēgta gadījumā, ja lietotājs nav reaģējis pēc trīs šādu ziņojumu saņemšanas.

Dažādu identifikatoru izmantošana, tos kombinējot samazina viltus poztīvo ziņojumu iespējamību, jo īpaši ziņojumiem, kuru avots ir *DGA*. Šādi ziņojumi var nebūt precīzi, tā kā arī leģitīma reklāmas, vai lietotāja klikšķu izsekošanas URL var tikt identificēti kā *DGA*. Autors piedāvā izmantot automatizētu pārbaudi vai identificētais *DGA* satur IP adresi DNS datubāzē un, vai tas nav atrodams Quad9 ļaunprātīgo domēnu datubāzē.

Minētā pieeja ļauj par drošību atbildīgajai personai iestādē noteikt kritērijus pēc kuriem pat mēģinājumi slēpt ļaunprātīgas programmas darbību var tikt konstatēti un, var tikt veiktas preventīvas darbības, tādas kā lietotāja ierīces pārbaude ar uzticamiem pretvīrusu risinājumiem.

3.5 DROŠĪBAS PĀRVALDĪBAS PLATFORMAS IMPLEMENTĀCIJA

Šajā apakšnodaļā izklāstītas augstākminētās vispārīgās IS drošības pārvaldības spējas modeļa ieviešanas vadlīnijas darbam jebkurā organizācijā neatkarīgi no tās specifiskas.

Lai adaptētu vispārīgos IS drošības pārvaldības spēju nepieciešami šādi soļi:

1) Jāpapildina spēju modelis ar organizācijas mērķiem par pamatu ņemot vispārīgo IS drošības pārvaldības spējas modeli, piemēram, ar atbilstības nodrošināšanu ISO27001 standartam;
2) jāpapildina spēju modelis ar datu avotiem un mērāmajiem atribūtiem, piemēram Netflow datiem, darbstaciju un serveru auditācijas pierakstiem un citiem organizācijai pieejamiem datu avotiem;
3) katram mērķim jādefinē mērāmie rādītāji, ar kuru palīdzību būs iespējams novērtēt mērķa sasniegšanas progresu, piemēram mērķim "nodrošināt sistēmu pieejamību" iespējams noteikt mērāmo rādītāju, ka sistēmas drīkst būt nepieejamas paredzētajā darba laikā ne ilgāk par 5 minūtēm;
4) katram mērķim nepieciešamas definēt servisu, kas nodrošina mērķa sasniegšanu, kā arī spējas konteksta elementus;
5) tāpat nepieciešams definēt mērāmos rādītājus konteksta elementa novērtēšanai, piemēram: konteksta elementam "ierīces draudu līmenis" iespējams noteikt mērāmo rādītāju, kas tiek mērīts ar *DGA* pieprasījumu skaitu dienā;

Pēc organizācijai specifisku IS drošības pārvaldības spējas mērķu definēšnas, nepieciešams definēt to mērāmos rādītājus, konteksta elementus un to mērāmos rādītājus, kā arī nepieciešams izveidot testa vidi IS drošības pārvaldības platformas izveidei.

Platformas un testa vides izveidei nepieciešami šādi soļi un aparatūra:

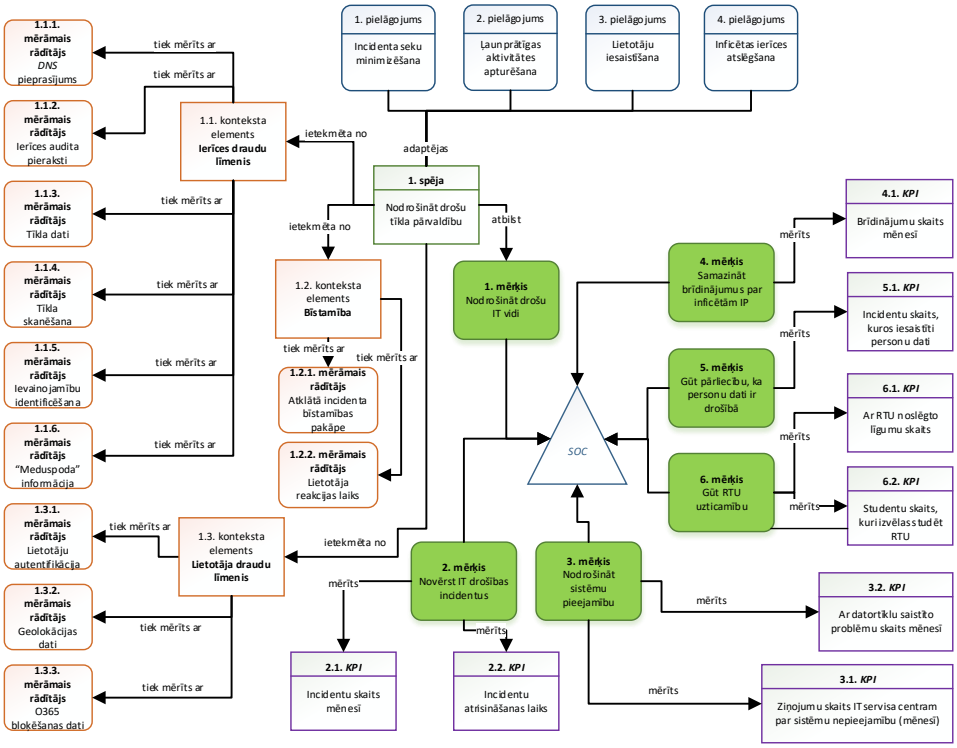
1) Nodrošināt *ISMS* platformu ar auditācijas pierakstiem; tai skaitā adaptēt auditācijas pierakstus, filtrējot liekos datu laukus, kuri nesniedz papildinformāciju, kas var būt noderīga drošības incidenta analīzē;
2) Izveidot Linux bāzētu serveri ar uzinstalētu Java 1.8. vai jaunāku un Apache Kafka. Minimālās prasības ir 8GB RAM, 4CORES un 500GB HDD;
3) Nokonfigurēt Apache Kafka, lai tā spēj saņemt datus no dažādiem datu avotiem;
4) Izveidot *Python* vai citā augsta līmeņa programmēšanas valodā veidotus skriptus, kuri spēj sadarboties ar Apache Kafka;
5) Izveidot Linux bāzētu serveri ar uzinstalētu InfluxDB un Grafana;
6) Veikt *Python* vai citas augsta līmeņa programmēšanas valodas izstrādāto skriptu pielāgojumus atbilstoši nepeieciešamās automatizācijas pakāpei.

4 SPĒJORIENTĒTA DROŠĪBAS PĀRVALDĪBAS MODEĻA IEVIEŠANA RTU

Lai organizācija gūtu maksimālu labumu no spējoritentētās drošības pārvaldības ieviešanas, tas attiecīgi jāpielāgo organizācijas vajadzībām. Tas nozīmē, ka nepieciešams pielāgot iepriekšējā nodaļā sniegto abstrakto modeli.

4.1 RTU IS DROŠĪBAS PĀRVALDĪBAS SPĒJU MODELIS

Šajā nodaļā ir dota iepriekš definētā abstraktā organizācijas IS drošības pārvaldības modeļa izvērstā versija, atbilstoši RTU lietošanas gadījumam.



4.1.att. *CDD* mērķa modelis

Šajā modelī drošības pārvaldības spējas tiek īstenotas, lai nodrošinātu drošu augstākās izglītības iestādes darbību. Šī spēja izpilda galveno mērķi 1.Nodrošināt drošu IT vidi, kas tiek atbalstīta ar pieciem papildu mērķiem: 2.Novērst drošības incidentus, 3.Nodrošināt sistēmu pieejamību. 4. Samazināt brdinājumus par inficētām IP, 5.Gūt pārliecību, ka personu dati ir drošībā un 6.Gūt RTU uzticamību. Katru apakšmērķi mēra ar atšķirīgu mērāmo rādītāju (KPI),

lai novērtētu šī mērķa izpildi. 2.mērķis tiek mērīts ar KPI 2.1.Incidentu skaits mēnesī un KPI 2.2.Incidentu atrisināšanas laiks, 3.mērķis tiek mērīts ar KPI 3.1.Ziņojumu skaits IT servisa centram par sistēmu nepieejamību (mēnesī) un ar KPI 3.2.Ar datortīklu saistīto problēmu skaits mēnesī. 4.mērķis tiek mērīts ar KPI 4.1.Brīdinājumu skaits mēnesī un 5.mērķis tiek mērīts ar KPI 5.1.Incidentu skaits, kuros iesaistīti personu dati. Visbeidzot 6.mērķis tiek mērīts ar KPI 6.1. un 6.2.Noslēgto līgumu skaits ar RTU un studentu skaits, kuri izvēlas studēt augstskolā.

Spēju ietekmē attiecīgie konteksta elementi: 1.1.Ierīces draudu līmenis, lai identificētu, vai ar tīklu savienota ierīce ir, iespējami inficēta un apdraud pārējās tīklā esošās ierīces, 1.2.Bīstamība, lai identificētu incidenta reaģēšanas laiku un 1.3.Lietotāja draudu līmenis, lai saprastu, vai lietotājs, piemēram, ir nozaudējis savus autentifikācijas datus. Visus konteksta elementus novērtē, izmantojot vairākus mērāmos rādītājus. Ierīces draudu līmeni mēra pēc 1.1.1.DNS pieprasījuma, kurā saprot, vai ierīce nav kompromitēta un nepieprasa robottīkla komandcentra domēnu komandu saņemšanai, 1.1.2.Ierīces audita pierakstiem, jeb žurnālfailu datiem, ja tādi pieejami, 1.1.3.Tīkla plūsmas datiem, kuros tiek identificēta ierīces netipiska uzvedība, 1.1.4.Tīkla portu skenēšanas identificēšanas un 1.1.5.Ievainojamību identificēšanas, kur ierīce tiek pārbaudīta attiecībā uz zināmām ievainojamībām. Draudu līmenis jeb konteksta elements 1.2.Bīstamība palielinās, ja identificētais incidents ir definēts, kā augstas prioritātes (*Critical*) (mērāmais rādītājs 1.2.1), kā arī ja 1.2.2.Lietotāja reakcijas laiks uz incidentu ir novēlots. 1.3.Lietotāja draudu līmenis ir atkarīgs no žurnālfailos reģistrētās lietotāja autentifikācijas informācijas (1.3.1.Lietotāju autentifikācija), 1.3.2.Geolokācijas datiem, izmantojot kurus, tiek identificēta lietotāja atrašanās vieta autentifikācijas laikā, kā arī 1.3.3.M365 automatizētās bloķēšanas informācija, kad lietotājs tiek bloķēts, ja tiek identificēta mēstuļu sūtīšana. Lietotāja reakcijas līmenis tiek novērtēts pēc 1.2.2.Lietotāja reakcijas laika.

Spēja 1 tiek īstenota, izmantojot četrus primāros pakalpojumus – ļaunprātīgas darbības identifikācijas pakalpojumu, kas izmanto vairākas metodes, lai identificētu inficētās ierīces tīklā, incidentu izmeklēšanu, kas nodrošina precīzu incidenta noteikšanu, kā arī nosaka tā bīstamības pakāpi, lietotāju informēšanas pakalpojumu, kas izmanto dažādus līdzekļus, lai informētu lietotājus par darbībām, kuras nepieciešamas incidenta novēršanai, kā arī inficētas ierīces atslēgšanas pakalpojumu, kas izmanto ugunsmūri un citas ierīces, lai atslēgtu inficēto ierīci no datoru tīkla. Spēju modeļa elementu apraksts ir parādīts 4.1.tabulā.

4.1.tabula

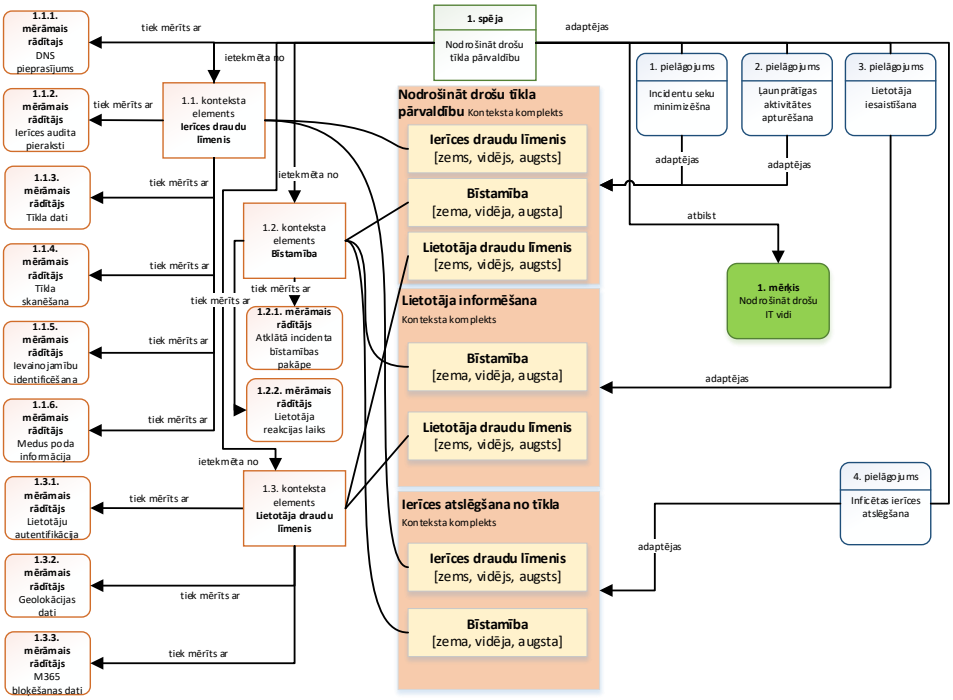
CDD modeļa elementu apraksts

Elementa klase	**Elementa numurs**	**Elementa nosaukums**	**Elementa apraksts**
Spēja	1	Nodrošināt drošu tīkla pārvaldību	Spēja nodrošināt drošu un nepārtrauktu augstākās iestādes darbību
Konteksta elements	1.1	Ierīces draudu līmenis	Nosaka vai pieslēgtā ierīce ir potenciāli inficēta
	1.2	Bīstamība	Nosaka vajadzību atrisināt drošības incidentu noteiktā laikā, pirms incidents nav izraisījis nopietnas sekas
	1.3	Lietotāja draudu līmenis	Identificē vai lietotājs var būt bīstams (piemēram hakeris), ņemot vērā lietotāja uzvedību tīklā
Mērāmie rādītāji	1.1.1	DNS pieprasījums	Ierīces DNS pieprasījumu informācija
	1.1.2	Ierīces auditācijas pieraksti	Konkrētās ierīces auditācijas pierakstu faili
	1.1.3	Tīkla datu informācija	Iekšējā tīkla datu pakešu informācija, kas ietver saņēmēja, sūtītāja IP adresi un portu, kā arī tīklā pārsūtāmo informāciju
	1.1.4	Tīkla skanēšanas identificēšana	Identificētā tīkla skanēšana, tai skaitā skanēšanas biežums
	1.1.5	Ievainojamību identificēšana	Identificētas ievainojamības informācijas sistēmās
	1.1.6	Medus poda informācija	Identificēti uzbrukumi, izmantojot “meduspodu”
	1.2.1	Atklātā incidenta bīstamības pakāpe	Uzrāda cik bīstams ir atklātais incidents
	1.2.2	Lietotāja reakcijas laiks	Uzrāda cik īsā laikā lietotājs ir reaģējis uz incidentu

	1.3.1	Lietotāju autentifikācija	Uzrāda lietotāja veiksmīgas autentifikācijas datus, kā arī citus ar lietotāju saistītos datus
	1.3.2	Geolokācijas dati	Uzrāda IP adrešu ģeolokācijas datus, lai identificētu valsti, no kuras veiksmīgi autentificējās lietotājs
	1.3.3	M365 bloķēšanas dati	Dati no M365 informācijas sistēmas par to, vai lietotājs ir bloķēts kā mēstuļu sūtītājs
Pielāgojums	1	Incidenta seku minimizēšana	Tiek pielāgotas darbības, lai minimizētu drošības incidenta ietekmi
	2	Ļaunprātīgas aktivitātes apturēšana	Tiek viekti pielāgojumi, lai apturētu ļaunprātīgu aktivitāti, piemēram, komutatorā atslēgta ierīce
	3	Lietotāja iesaistīšana	Lietotāju iesaistīšana incidenta novēršanā, piemēram, informējot lietotāju par veicamajām darbībām
	4	Inficētas ierīces atslēgšana	Pielāgojums, kas atslēdz ierīci no tīkla gadījumos, kad nav iespējama lietotāja identificēšana, vai lietotājs nereaģē uz brīdinājumiem
Mērķis	1	Nodrošināt drošu IT vidi	Galvenais mērķis ir nodrošināt drošu darbu datoru tīklā. Pārējie mērķi ir pakārtoti galvenajam mērķim
	2	Novērst IT drošības incidentus	Mērķis novērst drošības incidentus, kuru esamība apdraud galveno mērķi
	3	Nodrošināt sistēmu pieejamību	Sistēmām jābūt gan drošām gan pieejamām. Nepieejamas sistēmas apdraud galveno mērķi
	4	Samazināt brīdinājumus par inficētām IP	Lai padarītu drošāku iekšējo un ārējo vidi, nepieciešams samazināt *CERT* sūtīto brīdinājumu skaitu
	5	Gūt pārliecību, ka personu dati ir drošībā	Mērķis ir pielietot preventīvos, korektīvos un kompensējošos līdzekļus personu datu aizsardzībai
	6	Gūt RTU uzticamību	Mērķis iegūt uzticamību ir ļoti svarīgs, jo no tā ir atkarīga

			sadarbības partneru un studentu vēlēšanās sadarboties ar RTU
KPI	2.1	Incidentu skaits mēnesī	Cik reizes noteiktā periodā tika identificēts incidents
	2.2	Incidentu atrisināšanas laiks	Laiks, kādā incidents tika novērsts paziņojot lietotājam un, piemēram, lietotājs ir iztīrījis iekārtu iekārtas inficēšanās gadījumā, vai iekārta atslēgta no tīkla gadījumā, kad lietotājs noteiktu laiku nav reaģējis uz paziņojumiem, vai ierīces lietotājs nav identificējams
	3.1	Ziņojumu skaits IT servisa centram par sistēmu nepieejamību	Pieteikto ziņojumu skaits IT servisa centram, kur identificētā problēma ir nepieejama sistēma
	3.2	Ar datortīklu saitīto problēmu skaits mēnesī	Ziņojumu skaits, kuros ir identificētas problēmas ar datortīklu
	4.1	Brīdinājumu skaits mēnesī	*CERT.LV* brīdinājumu skaits mēnesī par inficētām IP
	5.1	Incidentu skaits, kuros iesaistīti personu dati	Pieteikto incidentu skaits, kuri saistīti ar personu datu integritāti vai konfidencialitāti
	6.1	Ar RTU noslēgto līgumu skaits	Līgumu skaits, kas noslēgti starp sadarbības partneriem un RTU
	6.2	Studentu skaits, kuri izvēlas studēt RTU	Studentu skaits, kuri izvēlas studēt RTU, jo ir pārliecināti, ka viņu dati ir drošībā, kā arī uzskata RTU par uzticamu iestādi

Konteksta un pielāgošanas modelis apskatāms 4.2.attēlā.



4.2.att. Konteksta un pielāgošanas modelis

Pielāgošanas modeļa apraksts ir definēts 4.2.tabulā.

4.2.tabula

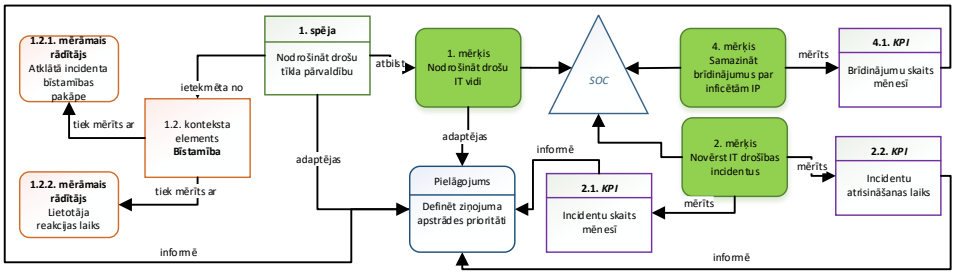
Pielāgojumu apraksts galvenajam modelim

Elementa klase	**Elementa numurs / satur elementus**	**Elementa nosaukums**	**Elementa apraksts**
Konteksta kopa	Ierīces draudu līmenis [Zems, Vidējs, Augsts]	Nodrošināt drošu tīkla pārvaldību	Zems līmenis nozīmē, ka drauds, lai arī ir identificēts, nerada zaudējumus organizācijai (piemēram "klikeri" un reklāmu pārvirzītāji), savukārt augsts līmenis nozīmē, ka kaitējums organizācijai var būt būtisks un var ietekmēt biznesa procesus
	Bīstamība [Zema, Vidēja, Augsta]		Augsts līmenis tiek noteikts, kad atklātā drauda bīstamība var radīt būtiskas sekas organizācijai, tai

			skaitā bet ne tikai veicot datu šifrēšanu, datu noplūdi u.c.
	Lietotāja draudu līmenis [Zems, Vidējs, Augsts]		Noteikta pēc iepriekš identificētas lietotāja uzvedības, piemēram tīkla skanēšanas, ievainojamību izmantošanas u.c. un netiek noteikta gadījumos, kad lietotājs nav identificējams
	Bīstamība [Zema, Vidēja, Augsta]	Lietotāja informēšana	Atkarībā no identificētā drauda bīstamības pakāpes lietotājs tiek informēts vai nu izmantojot e-pastu, portālu vai īzsiņas (SMS) pakalpojumus
	Lietotāja draudu līmenis [Zems, Vidējs, Augsts]		Noteikta pēc iepriekš identificētas lietotāja uzvedības, piemēram tīkla skanēšanas, ievainojamību izmantošanas u.c. un netiek noteikta gadījumos, kad lietotājs nav identificējams
	Ierīces draudu līmenis [Zems, Vidējs, Augsts]	Ierīces atslēgšana no tīkla	Tiek izmantota gadījumos, kad lietotājs netika identificēts, bet bīstamība draudam ir noteikta kā augsta, kā arī gadījumos, kad ierīces draudu līmenis noteikts kā augsts. Draudu līmeņa noteikšanas kritērijs ir gan drauda bīstamība, gan drauda īstenošanās laiks
	Bīstamība [Zema, Vidēja, Augsta]		Atkarībā no identificētā drauda bīstamības pakāpes ierīce vai nu tiek atslēgta no tīkla vai noūtīts ziņojums lietotājam. Gadījumos, kad lietotājs nav identificējams un bīstamība ir zema, drauds netiek ņemts vērā
Pielāgojums	1	Pārliecināties, ka ziņojums nav viltus pozitīvs	Nosaka spēju identificēt viltus pozitīvos ziņojumus. Spēja tiek nodrošināta izmantojot eksperta izstrādātu datubāzi ar kritērijiem patiesi pozitīvo noteikšanai. Katram no šiem kritērijiem ir izstrādāta lietotājam nosūtāmā informācija ar nepieciešamo rīcību un pamācību drauda novēršanai

	2	Definēt incidenta apstrādes prioritāti	Tiek dinamiski mainīta incidenta klasifikācija, ņemot vērā tā prioritāti
	3	Izvēlēties lietotāja apziņošanas veidu	Sistēma nosaka kā tiks nosūtīts ziņojums lietotājam, izmantojot lietotāja portālu, e-pastu vai īsziņu. Prioritāri ziņojumi tiek nosūtīti izmantojot uzreiz visus apziņošanas veidus
	4	Identificēt vai nepieciešams ierīci atslēgt no tīkla	Sistēma noska, vai ierīce būtiski apdraud tīklu un vai tā ir atslēdzama no tīkla

Reaģēšanas spējas adaptācija prioritāru ziņojumu gadījumā ir noteikta 4.3.attēlā.



4.3.att. Prioritātes korekcija

Atkarībā no ziņojumu skaita par inficētām IP un incidentu skaita mēnesī tiek papildu pielāgota reaģēšana uz šiem incidentiem. Ja ziņojumu skaits pārsniedz noteiktu līmeni, ziņojumu pritoritāte tiek paaugstināta un notiek iepriekš izmantotā apziņošanas mehānisma maiņa no e-pasta uz SMS.

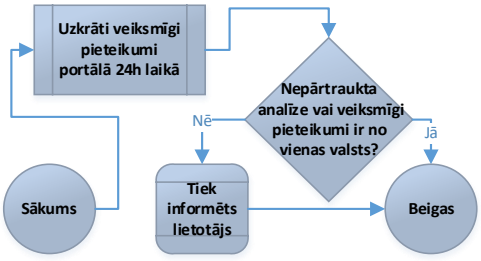
4.2 ARHITEKTŪRAS DETALIZĀCIJA RTU LIETOŠANAS GADĪJUMAM

Kopējā *ISMS* platformas arhitektūra parādīta 4.6.attēlā. Papildus augsta līmeņa arhitektūrā definētajiem risinājumiem tika izmantots:

- CentOS – operētājsistēma, kas atšķiras ar ilgspējīgu darbību pat kernela līmeņa atjaunināšanas gadījumā;
- Powershell – kas Microsoft izstrādāts, platformas neitrāls uzdevumu automatizācijas risinājums, kurš sastāv no komandrindas pamatkomponentes, skriptu valodas un konfigurācijas pārvaldības ietvara. PowerShell darbojas operētājsistēmās Windows, Linux un macOS;
- *Suricata IDS* – populāra atvērtā koda ielaušanās noteikšanas sistēma, kuras signatūras ir viegli adapatējamas;
- *Nfdump* – populārs un pielāgojams Netflow datu apstrādes rīks;
- *Tcpdump* – efektīvs atvērtā koda tīkla datu analizators, kurš darbojas komandrindā;
- *Fprobe* – pielāgojams, atvērtā koda Netflow datu apstrādātājs, kura galvenais uzdevums ir ģenerēt Netflow datus no saņemtajiem tīkla datiem;
- *Palo Alto FW* – Gartner kvadranta augsti novērtēts nākamās paaudzes ugunsmūris, kurš ietver gan IPS funkcinalitāti, gan *DGA* analīzes rīkus.

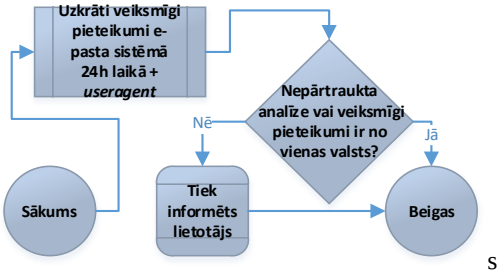
Platformas analīzes komponente ietver:

1) Pieteikšanās portālā komponenti, kuras darbība ir parādīta 4.4.attēlā. Analīzes pamatā ir RTU iekšējā portāla veiksmīgas autentifikācijas dati, kas iekļauj laiku, kad noticis pieteikums, pieteicēja lietotājvārdu, kā arī IP adrese no kuras veiksmīgais pieteikums ir noticis. Dati tiek uzkrāti *Infux DB* datubāzē, no kuras noteiktā intervālā tiek izgūta un apstrādāta izmantojot *Pyhon* skriptu informācija par veiksmīgiem autentificēšanās gadījumiem pēdējo 24 stundu laikā.



4.4.att. Pieteikšanās portālā komponente

2) Pieteikšanās E-pasta sistēmai komponenti, kura parādīta 4.5.attēlā. Šīs analīzes pamatā ir M365 autentifikācijas dati, kuri ir iegūti izmantojot Powershell skriptus. Autentifikācijas dati iekauj informāciju par laiku, kad pieteikums ir noticis, lietotājvārdu, kurš veicis pieteikumu, statusu, vai pieteikums ir bijis veiksmīgs kā arī pieteicēja IP adresi un pārlūka identificējošo informāciju (*user agent*). Līdzīgi kā Pieteikšanās portālā komponentei, dati tiek nosūtīti uz Influx DB datubāzi un vēlāk apstrādāti izmantojot *Python* skriptu.



s

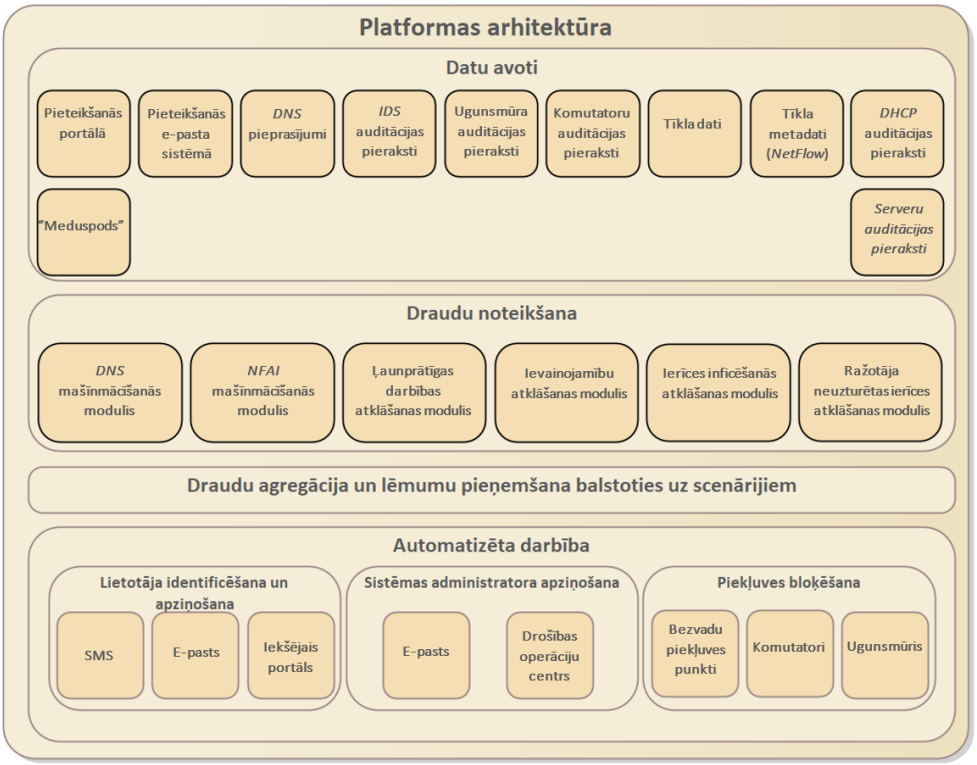
4.5.att. Pieteikšanās e-pasta sistēmā komponente

3) DNS pieprasījumu analīzi, izmantojot *Python* skriptu un vēlāku datu eksportu uz Apache Kafka [121]. *DNS* pieprasījumu analīze notiek divos veidos:
    1. identificējot vai *DNS* pieprasījums ir publiski zināmajos melnajos sarakstos [70];
    2. ugunsmūris ir noteicis *DNS* kā aizdomīgu.
4) IDS [122] datus, kuri tiek uzkrāti un apstrādāti izmantojot *Python* skriptus un Apache Kafka, nosakot vai ziņojums var tikt uzskatāms par viltus pozitīvo. Šim nolūkam ir izveidota patiesi pozitīvo ziņojumu datubāze. Katrs no šīs datubāzes ierakstiem paredz dažādu reakciju. Piemēram, ja ir identificēts aizdomīgs *DNS* pieprasījums, tiks nosūtīta e-pasta ziņa attiecīgajam ierīces lietotājam, bet ja identificēta zināma ļaunprātīgas programmatūras darbība, ziņa lietotājam tiks nosūtīta izmantojot gan e-pastu, portālu un SMS.
5) Ugunsmūra datus, kuri tiek saņemti Apache Kafka un analizēti izmantojot *Python* skriptus, līdzīgi, kā tas ir ar *IDS* datiem, nosakot specifiskus pasākumus specifiskiem notikumiem.
6) Komutatoru datus, kuri tiek izmantoti, lai identificētu ierīces pieslēguma vietu.
7) Tīkla datus, kuri tiek uzkrāti par specifiskiem definētiem notikumiem, lai nepieciešamības gadījumā būtu iespēja veikt incidenta analīzi.
8) *NetFlow* datus, kuri tiek uzkrāti par noteiktu laika periodu. Šie dati turpmāk tiks izmantoti gan incidentu analīzei, gan mašīnmācīšanās modeļa apmācībai, kā arī skanētāju noteikšanai un bloķēšanai.

9) DHCP datus, kuri tiek izmantoti, lai uzkrātu informāciju par IP adrešu piešķiršanu un IP adreses fizisko atrašanās vietu, kā arī ierīces MAC adresi. Tam tiek izmantoti dati gan no komutatoriem, gan *"Hewlett Packard Enterprise Intelligent Management Center"* [123] rīka, kas apkopo šos datus.
10) Ievainojamību identificēšanas komponenti, pielietojot ievainojamību skaneri *Nessus* [104]. Ievainojamību skanēšana notiek automatizētā režīmā, izmantojot *Python* skriptus. Ja rezultāti uzrāda, augstu vai kritisku ievainojamību, tie tiek automātiski e-pasta veidā nosūtīti atbildīgajai par ievainojamu ierīci personai. Ievainojamību pārbaudes tiek veiktas gan serveriem, gan darbstacijām, gan arī mobilajām ierīcēm. Lai nepārslogotu tīklu saknējot neatbilstošas ierīces, skanēšanas mērķi (ierīces) tiek automatizēti noteikti izmantojot specifisku portu skanēšanu ar Nmap [124].
11) DNS pieprasījumu apstrāde, kuras realizācijai ir izmantojami dažādi mašīnmācīšanās algoritmi.
12) Medus poda datu apstrāde un rezultātu nosūtīšana uz Apache Kafka [121].

Bez augstākminētām komponentēm *ISMS* platforma sastāv no mikroservisiem, kuri ir integrēti platformā izmantojot Apache Kafka ziņojumu sistēmu. Mikroservisu skaits piedāvājamajā platformā nav ierobežots. Mikroservisi nodrošina tādus papildu elementus, kā iekšējā tīkla skanēšanas identificēšanu, rupja spēka paroles pārlasīšanas uzbrukumus u.c.. Šie mikroservisi ir atrodami darba 2.pielikumā.

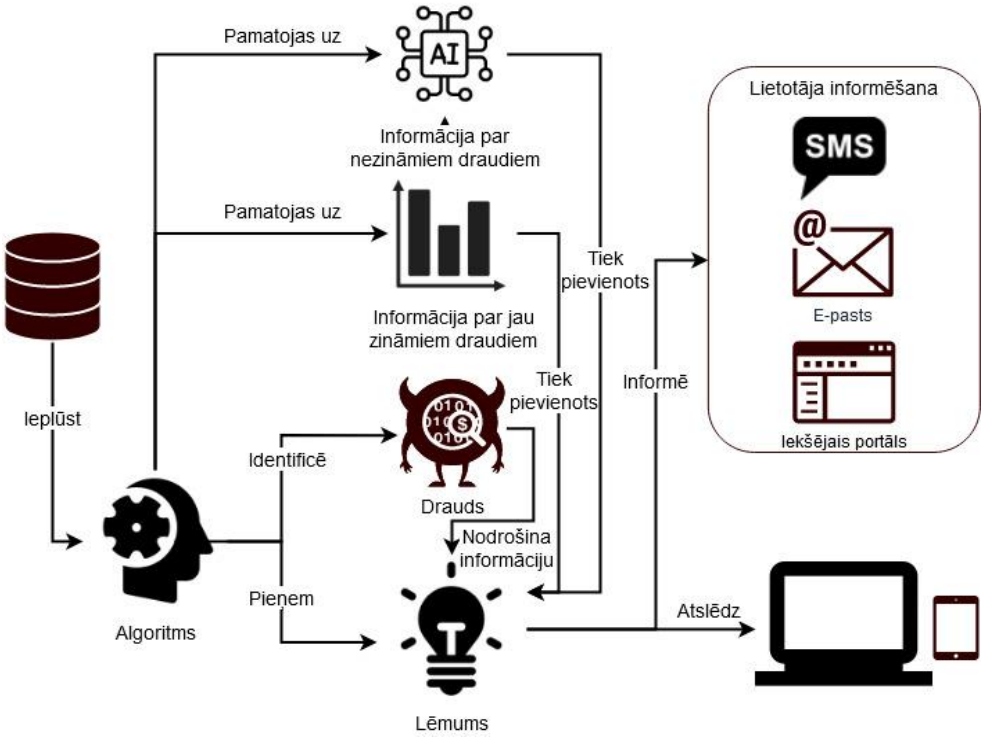
Darbā piedāvātās *ISMS* platformas pamatā ir lieto datu koncepts, šim mērķim izmantojot Apache Kafka un Apache Spark. Ieviešot lielo datu konceptu *ISMS* plaftormā, datu apstrādes un sistēmas informēšanas par incidentu laiks tika samazināts no iepriekš vidēji 70 sekundēm, kad tika izmantots linux operētājsistēmā iebūvētais rīks Cron, līdz vidēji 10 sekundēm, ar iespēju samazināt laiku zem 5 sekundēm, ja autors veiktu IP adrešu sasaisti ar lietotāju pirms ir identificēts incidents. IP adreses sasaiste ar lietotāju notiek pēc tam, kad tiek identificēta, iespējams, inficēta ierīce. Autors uzskata, ka 5 sekunžu ietekme nav nozīmīgāka par lietotāju privātuma saglabāšanu saskaņā ar GDPR [11].



4.6.att. *ISMS* platformas arhitektūra

ISMS darbības komponente ietver:

1) Lietotāja identificēšanas un apziņošanas moduli, ar kura palīdzību tiek identificēts lietotājs, kura ierīcei ir piešķirta IP adrese. Šajā modulī tiek izmantots Apache Kafka [121] un *Python* skripti. Modulis ietver 3 apakšmoduļus:
    a. Lietotāja apziņošana izmantojot SMS;
    b. Lietotāja apziņošana izmantojot e-pastu
    c. Lietotāja apziņošana izmantojot iekšējo portālu
2) Administratora apziņošana tiek nodrošināta izmantojot e-pastu un operācijas centra grafisko saskarni, kuras pamatā ir Influx datubāze [125] un Grafana grafiskais interfeiss [126].
3) Piekļuves bloķēšanas modulis iekļauj bezvadu tīkla bloķēšanu izmantojot Aruba Clearpass [127]. Piekļuve komuntatoriem var tikt bloķēta automātiski, izmantojot mikroservisus, kuri nododod informāciju Ugunsmūrim par bloķējamajām IP adresēm.



4.7.att. *ISMS* platformas funkcionālā shēma

Piedāvātās *ISMS* platformas funkcionālā shēma parādīta 4.7.attēlā. Shēmas pamatā ir datu avotu apstrāde izmantojot dažādus algoritmus, ļaunprātīgas aktivitātes konstatēšana, kā arī lēmuma pieņemšana. Datu apstrādes algoritmi pamatojas uz informāciju par jau zināmiem draudiem kā arī izmanto mašīnmācīšanās metodes, lai identificētu vēl nezināmus draudus. Lēmuma pamatā ir ierīces lietotāja informēšana vai atslēgšana no datortīkla.

5 IS DROŠĪBAS PĀRVALDĪBAS PLATFORMAS NOVĒRTĒJUMS

ISMS darbība tika vērtēta dažādos laika periodos sākot ar 2021.gada janvāri, veicot dažādus mērījumus, un turpmāk nodaļā tiks atspoguļoti dažādu novērtējumu rezultāti, gan attiecībā uz individuāliem moduļiem, kuru pamatā ir mašīnmācīšanās komponentes, gan attiecībā uz visu *ISMS* platformu kopā. Par periodu no 2021.gada janvāra līdz 2024.gada janvārim *ISMS* platforma ir ģenerējusi vairāk kā 88 tūkst. ziņojumus. Lielākā daļa no kuriem bija ieteikumi noskenēt iekārtu un veikt uzlabojumus drošībā, piemēram, atjaunināt novecojušu programmatūru.

5.1 ĻAUNPRĀTĪGĀ DNS PIEPRASĪJUMA IDENTIFICĒŠANA

Ļaunprātīga DNS pieprasījuma identificēšanas nolūks ir noskaidrot vai DNS pieprasījums var tikt klasificēts kā *DGA*, tadējādi var tikt uzskatīts par ļaunprātīgu pieprasījumu. Ļaunprātīgi pieprasījumi var netieši norādīt uz to, ka ierīce atrodas robotu tīklā un vēlās saņemt instrukcijas turpmākajam darbam.

$$DGA = F\ (DNS)$$

DGA ir funkcija *F* no izsauktā *DNS* pieprasījumu, ar kura palīdzību tiek noskaidrots, vai pieprasījums ir leģitīms vai automātiski ģenerēts.

Lai izstrādātu *DGA* domēnu identificēšanas moduli, tika veiktas šādas darbības:

1) Datu kopas sagatavošana;
2) Pazīmju un klasifikatoru izvēle pamatojoties uz literatūras analīzi;
3) Moduļa darbības izvērtējums.

5.1.1 Apmācības datu kopa

Apmācības datu kopa tika sagatavota 3 mēnešu laikā apstrādājot *DNS* pieprasījumus RTU iekšējā tīklā. Domēni tika klasificēti "labajos" un "sliktajos", ņemot vērā eksperta, VirusTotal un *Quad9* informāciju par *DNS* ierakstu (5.2.tabula). Dati tika attīrīti no neeksistējošiem domēniem, kuri tika ievadīti kļūdaini vai pieprasījums noticis uz .lan domēnu. Šī pārbaude tika veikta salīdzinot domēnus ar *ICANN* [128] datiem. Kritēriji domēnu klasifikācijai ir definēti 5.1.tabulā.

5.1. tabula

Kritēriji domēnu klasifikācijai

Nosacījums	**Pseido kods**	**Piemēri**
1	IF DOMĒNS NESATUR PATSKAŅUS THEN AIZDOMĪGS++	0sntp7dnrr.com, pnghst.com
2	IF DOMĒNS NESATUR LĪDZSKAŅUS THEN AIZDOMĪGS++	3458ee.com, o5o4o6.com
3	IF DOMĒNS SATUR TIKAI CIPARUS THEN AIZDOMĪGS++	127777.com, 10000114.com, 12688888.com
4	IF DOMĒNS SATUR TIKAI HEKSADECIMĀLUS SIMBOLUS THEN AIZDOMĪGS++	442d9f2ac50ca502.com, 8cb0309458c7b35e.com
5	IF DOMĒNS SATUR VISMAZ 3 PATSKAŅUS PĒC KĀRTAS THEN AIZDOMĪGS++	zwyr157wwiu6eior.com, zy16eoat1w.com. boooooooom.com, iiiiiiiiiii.net
6	IF DOMĒNS SATUR VISMAZ 3 LĪDZSKAŅUS PĒC KĀRTAS THEN AIZDOMĪGS++	yqezqofkb1nnmz.com, 6l1twlw9fy.com, eclkmpbn.com
7	IF DOMĒNS SATUR SKAITĻUS, KURI NAV PĒC KĀRTAS THEN AIZDOMĪGS++	eh8jq4cmq8j9g5.com, 5kv261gjmq04c9.com

Nosacījumi datu kopu izveidošanai ir definēti 5.2.tabulā.

5.2.tabula

Datu kopu izveidošanas kritēriji

Nosacījums	**Pseidokods**
1	IF AIZDOMĪGS == 0 THEN

	IF EKSPERTS NOTEICIS KĀ SLIKTU DOMAIN_MALICIOUS = 1; EXIT ELSE DOMAIN_MALICIOUS = 0; EXIT ELSE GOTO RULE2
2	IF DOMĒNS IN ALEXA TOP 100000 [31] THEN DOMAIN_MALICIOUS = 0; EXIT ELSE GOTO RULE3
3	IF DOMĒNS IN QUAD9 AS MALICIOUS THEN DOMAIN_MALICIOUS = 1; EXIT ELSE DOMAIN_MALICIOUS = 0; EXIT
4	IF DOMĒNS IN VIRUSTOTAL AS MALICIOUS THEN DOMAIN_MALICIOUS = 1; EXIT ELSE DOMAIN_MALICIOUS = 0; EXIT

Papildu uzkrātajai datu kopai domēni tika iekļauti no Alexa [31] top mājas lapām. Rezultātā tika iegūta datu kopa ar 17712 DNS ierakstiem, kas sastāvēja no reāliem domēniem. Datu kopa saturēja 8856 ierakstus, kuri tika klasificēti kā "labie" un 8856 ierakstus, kuri tika klasificēti kā "sliktie". Turpmāk tiks aprakstīti eksperimenti labākās datu kopas un labākā klasifikatora izvēlei.

5.1.2 Modeļa veidošana un rezultāti

Pazīmju kopa *DNS* klasifikatoru apmācībai tika izveidota par pamatu ņemot Selvi et al. [73] pētījumu (5.3.tabula).

5.3.tabula

Pazīmju definēšana (adaptēts no *[73]*)

Pazīmes Nr.	Apraksts	Rezultāts *DNS* pieprasījumam: 7hu8e1u001.com
F1	Garums	10
F2	Vidējais "unigram"	28.39
F3	Vidējais "bigram"	0.257
F4	Vidējais "trigram"	0.229
F5	"unigram" standartnovirze	0.0

F6	“bigram” standartnovirze	3.459
F7	“trigram” standartnovirze	3.430
F8	Patskaņu attiecība pret garumu	0.300
F9	Līdzskaņu attiecība pret garumu	0.100
F10	Unikālo simbolu attiecība pret garumu	0.001

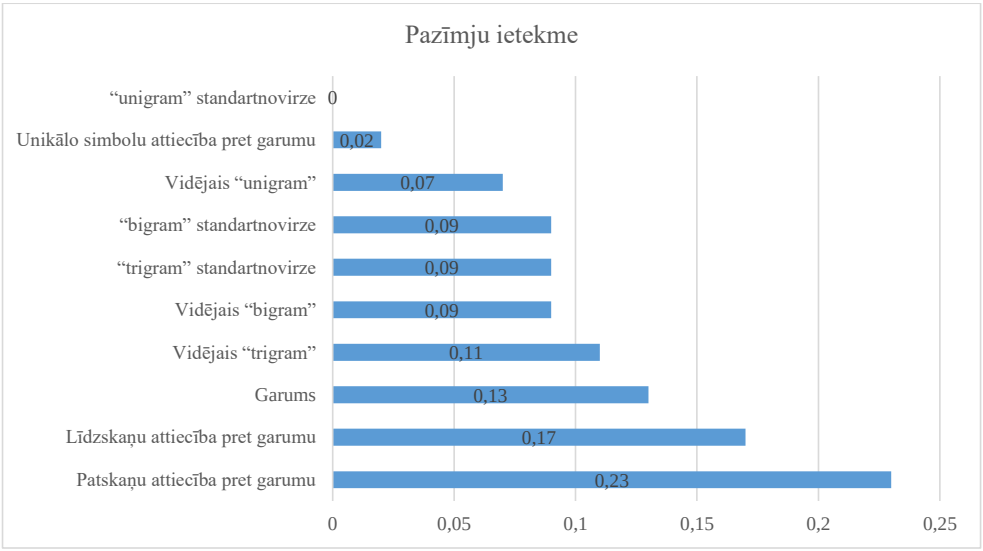
Tika izvēlēti četri klasifikatori: atbalsta vektoru mašīna (*SVC*), neironu tīkli (*NNC*), lēmumu koki (*DTC*) un lēmumu meži (*RFC*). Lai novērtētu klasifikatorus, katram no viņiem tika mērīta precizitāte, pārklājums, F1-mērs un ticamība. Rezultāti tika novērtēti izmantojot 10-kārtīgu šķērsvalidāciju. Pēc sākotnējā novērtējuma (5.4.tabula) tika veikts pazīmju ietekmes uz rezultātu novērtējums izmantojot gadījuma mežu klasifikatora *Python* bibliotēku pazīmju ietekmes novērtējumam (5.1.attēls).

DNS identificēšanai autors dod priekšroku ticamībai.

5.4.tabula

Klasifikatoru vērtības

Klasifikators	Precizitāte	Pārklājums	F1-mērs	Ticamība
RFC	0,934	0,948	0,941	0,940
DTC	0,916	0,928	0,922	0,921
NNC	0,869	0,854	0,862	0,852
SVM	0,858	0,860	0,859	0,859



5.1.att. *RFC* klasifikatora pazīmju ietekmes novērtējums

Eksperimentu rezultātā tika iegūti līdzīgi rezultāti kā Ahluwalia et al [71] pētniekiem, kur, par labāko klasifikatoru *DGA* klasificēšanai tika atzīts *RFC*. Nākamais solis - noskaidrot, kuras pazīmes vislabāk veicina precizitātes uzlabojumu *RFC* klasifikatoram, tādēļ primāri tika klasifikatora pazīmju ietekmes novērtējums (5.1.attēls).

Pazīmes (5.5.tabula) tika klasificētas 5 pazīmju kopās. Kopas tika sadalītas balstoties uz 5.1.attēlu par svarīgākajām pazīmēm *DGA* noteikšanā, pakāpeniski izņemot mazāk svarīgās pazīmes.

5.5.tabula

Pazīmju kopas

-	1.kopa	2.kopa	3.kopa	4.kopa	5.kopa
F1	1	1	1	1	0
F2	1	1	0	0	1
F3	1	1	1	1	1
F4	1	1	1	1	1
F5	0	0	0	0	1
F6	1	1	1	0	1
F7	1	1	1	1	1
F8	1	1	1	1	1

F9	1	1	1	1	1
F10	1	0	0	0	1

Lai noskaidrotu izvēlēto pazīmju ietekmi uz klasifikatora identificēšanas precizitāti, tika izmantota 10-kārtīga škērsvalitācija katrai pazīmju kopai (5.6.tabula).

5.6.tabula

RFC modeļa pilnveidošana

-	Precizitāte	Pārklājums	F1-mērs	Ticamība
1.kopa	0,934	0,947	0,940	0,940
2.kopa	0,933	0,946	0,939	0,939
3.kopa	0,933	0,946	0,939	0,939
4.kopa	0,930	0,946	0,938	0,938
5.kopa	0,934	0,948	0,941	0,940

Lai arī var iztikt bez "unigram" standartnovirzes, eksperimentu rezultāti parādīja, ka *DGA* identificēšanai vislabākie rezultāti ir izmantojot visas 10 pazīmes, kas 5.5.tabulā attēlota kā 5.pazīmju kopa.

5.1.3 DNS moduļa salīdzinošais novērtējums

DNS moduļa izmantošana ļāva uzlabot ierīces inficēšanās noteikšanas precizitāti. Lai arī mūsdienās DNS pieprasījumus ir iespējams slēpt izmantojot gan *DNS over HTTP*, gan arī *DNS over TLS* metodes, rezultāti apliecina, ka šādas metodes netiek bieži izmantotas. Ļaunprātīgas programmatūras identificēšana izmantojot *DNS* parādīja labus rezultātus un, integrējot šo moduli *ISMS* platformā, tika identificēta antivīrusu kompānijām vēl nezināma ļaunprātīgas programmas darbība. *DNS* modulis, kurš izmantoja *RFC* mašīnmācīšanās algoritmu, tika apmācīts ar 8856 un "sliktajiem" un 8856 "labajiem" domēniem. Tas tika salīdzināts ar Palo Alto ugunsmūri, kurš atrodas Gartnera maģiskā kavadranta [129] augšējā labajā stūrī, tātad ar pilnīgu vīziju un spēju šo vīziju īstenot. Salīdzināšana tika īstenota pēc tam, kad ugunsmūris tika atjaunināts un saņēma *DGA* identificēšanas funkcionalitāti. *DGA* modulis identificēja *DGA* 92.4% gadījumos, kad arī ugunsmūris identificēja *DGA*. Salīdzinot ar ugunsmūri, *DGA* modulis 8,5% gadījumos ir atklājis *DGA*, kuru ugunsmūris nebija atklājis, bet to ir apstiprinājusi vismaz viena no trim trešām pusēm (*Cloudfare* [130], *Norton* [131] vai *Quad9* [80])

Tika salīdzināta ugunsmūra un *DGA* moduļa darbība identificējot *DGA* (

5.7.tabula).

5.7.tabula

DGA moduļa salīdzināšana ar ugunsmūri

Nr.p.k.		Ugunsmūris identificējis *DGA* (kopā:87)	modulis identificējis *DGA* (kopā:167)
1.	Ugunsmūris neidentificēja *DGA*	-	142
2.	Ugunsmūris identificēja *DGA*	87	25
3.	modulis neidentificēja *DGA*	8	-
4.	modulis identificēja *DGA*	79	167
5.	Trešā puse identificēja *DGA*	31	18
6.	Trešā puse identificēja *DGA*, modulis identificēja *DGA*, bet ugunsmūris neidentificēja *DGA*	-	7

Rezultātos aspoguļotais (

5.7.tabula) 1.punkta jautājums nosaka: Vai esošais ugunsmūris ir identificējis tos *DGA*, kurus ir identificējis *DGA* modulis? Atbilde: 142 no 167 ugunsmūris nav identificējis kā *DGA*, tas sasaucās ar jautājumu Nr.2, kur 25 *DNS* pieprasījumus ugunsmūris arī ir identificējis kā *DGA*. Tabulas 3.punkts nosaka cik DNS pieprasījumu netika identificēti ar *DGA* moduli, rezultāti uzrāda, ka 8 ugunsmūra identificētos *DGA* izstrādātais modulis neidentificēja, kas atbilst

5.7.tabulas 4.punktam, kur izstrādātais modulis identificēja 79 no 87 ugunsmūra identificētajiem *DGA*. Tabulas 5.punkts nosaka trešo pušu (Norton, Cloudfare vai Quad9) apstiprinājumu, ka pieprasītā DNS ir uzskatāma par *DGA*. Rezultāti liecina, ka ugunsmūra identificētos *DGA* vairāk identificē arī trešās puses (kopā 31 no 87), tas, iespējams, ir skaidrojams ar faktu, ka ugunsmūris ņem informāciju no tiem pašiem avotiem, kur trešās puses. *DGA* moduļa gadījumā trešās puses apliecināja tikai 18 no 167 *DNS* pieprasījumiem kā *DGA*, tas, savukārt var tikt skaidrots ar faktu, ka nevienai no trešām pusēm šāds *DNS* vēl nav zināms, jo *DGA* modulis neizmanto ļaunprātīgo *DNS* datubāzes. Viss process ir pilnībā balstīts uz *DNS* vārda analīzi izmantojot mašīnmācīšanos. Tabulas 6.punkts pierāda to, ka *DGA* modulis, lai arī, iespējams, uzrāda vairāk viltus pozitīvo, tomēr ir atklājis par 7 vairāk *DNS* pieprasījumu, kurus arī trešā puse uzskata par *DGA*.

5.2 ĻAUNPRĀTĪGAS DARBĪBAS IDENTIFICĒŠANA TĪKLA DATOS (NFAI MODULIS)

NFAI moduļa izstrādes nolūks ir identificēt vēl nezināmus draudus izmantojot apmācītos mašīnmācīšanās modeļus un tīkla datos esošo informāciju. Galvenā šī moduļa priekšrocība salīdzinot ar klasiskajiem melnajiem sarakstiem ir iespēja identificēt ļaunprātīgu aktivitāti bez specifiski izstādātām *IDS* signatūrām.

$$B = G\ (NetFlow)$$

B – ļaunprātīga aktivitāte, kuras identificēšanai izmanto funkciju *G NetFlow* ievaddatiem.

NFAI moduļa izstrāde ietver šādus soļus:

1) datu kopas izveide,
2) pazīmju un klasifikatoru izvēle balstoties uz literatūras apskatu, kā arī autora veiktajiem eksperimentiem,
3) pazīmju dimensionalitātes samazināšana,
4) klasifikācijas modeļa novērtējums apstrādājot reāla laika tīkla datus.

Datu apmācības kopa tika iegūta veicot *NetFlow* datu uzkrāšanu 3 mēnešu garumā. Iegūtie *NetFlow* dati tika dalīti "labajos" un "sliktajos" ņemot vērā ugunsmūra un *IDS* ziņojumus par incidentiem. Visi iegūtie dati tika sadalīti 10 minūšu intervālos un, gadījumos, kad *IDS* un/vai ugunsmūris ir identificējis ļaunprātīgu darbību, tika iegūti vēl papildu 24 stundu *NetFlow* dati par attiecīgo IP adresi pirms inficēšanās tika konstatēta un arī šie dati tika definēti kā "sliktie". Visa iegūtā datu kopa tika pārskatīta un rezultātā iegūti "sliktie" un "labie" *NetFlow* datu 10 minūšu intervāli. Turpmāk tika izstrādāta sākuma pazīmju kopa, pamatojoties uz Sheng et al [44].

5.2.1 Apmācības datu kopa

Lai iegūtu datus tika izmantots *fprobe* [58] rīks, kurš ievāc tīkla datus un izveido *NetFlow* datu struktūru, kuru uzkrāj *nfcapd* rīks, kurš ir nfdump rīka sastāvdaļa [59]. Datu sagatavošana iekļāva:

1) *Nfdump* rīka izmantošanu ar papildu opcijām "*fmt:%ts %td %pr %sap %dap %flg %ipkt %ibyt %opkt %obyt %fl*", lai iegūtu 10 minūšu intervālus ar datiem un saglabātu tos teksta failā.

2) Teksta faila apstrādi izmantojot *Apache Spark* [112] un *Python* programmēšanas valodā rakstītus mikroservisus:
    a. datu kopas izveidi no teksta faila;
    b. operācijas ar datu kopu, izmantojot iebūvēto *SQL* funkcionalitāti, lai iegūtu visas nepieciešamās pazīmes turpmākai apstrādei (5.3.tabula)
3) Pēc datu apstrādes, rezultāts ir *json* fails (5.2.attēls).


```
{"src":"100.0.2.99","total_src_packets":1,"src_min_packet":1,"src_max_packet":1,"src_mean_packets":1.0,"src_variance_packets":0.0,"total_src_bytes":52,"src_min_bytes":52,"src_max_bytes":52,"src_mean_bytes":52.0,"src_variance_bytes":0.0,"total_src_flows":1,"total_dst_packets":0,"dst_min_packet":0,"dst_max_packet":0,"dst_mean_packets":0.0,"dst_variance_packets":0.0,"total_dst_bytes":0,"dst_min_bytes":0,"dst_max_bytes":0,"dst_mean_bytes":0.0,"dst_variance_bytes":0.0,"total_dst_flows":0,"total_flows":1,"percentage_src_flows":1.0,"percentage_dst_flows":0.0,"count_unique_src_to_dst_IP":1,"count_unique_dst_to_src_IP":0,"count_unique_src_ports":1,"count_unique_dst_ports_from_same_src":1,"src_high_ports_precentage":1.0,"src_low_ports_precentage":0.0,"dst_high_ports_precentage":0.0,"dst_low_ports_precentage":1.0,"src_udp_percentage":0.0,"src_tcp_percentage":1.0,"src_ping_percentage":0.0,"dst_udp_percentage":0.0,"dst_tcp_percentage":0.0,"dst_ping_percentage":0.0}
```


5.2.att. Iegūtais rezultāts *json* failā

Sagatavotie dati tika sadalīti "labā" un "sliktā" datu kopā, ņemot vērā informāciju no dažādiem avotiem, piemēram, ugunsmūra, ielaušanās noteikšanas sistēmas un lietotāju atsauksmēm. Iegūtie dati tika vēlreiz manuāli pārskatīti un tika izņemtas datu kopas, kuras satur pārāk maz informācijas. Tika sagatavota apmācību kopa, kura sastāvēja no 4548 "labajiem" un "290" sliktajiem 10 minūšu *NetFlow* datu intervāliem. Lai arī datu kopas nav sabalansētas, diemžēl, eksperimentu rezultātā vairāk pārliecinošu apmācāmo datu autoram nav izdevies iegūt.

Sākuma pazīmju kopa (5.8.tabula) tika izveidota sakaņā ar [44], kā arī ņemot vērā autora pieredzi tīkla datu analīzē, identificējot noderīgākās pazīmes.

5.8.tabula

Sākuma pazīmju kopa

Pazīmes Nr.	Apraksts
F1	Kopējais pārsūtīto pakešu skaits no avota adreses
F2	Minimālais pakešu skaits vienā plūsmā, kas pārsūtīts no avota adreses
F3	Maksimālais pakešu skaits vienā plūsmā, kas pārsūtītas no avota adreses
F4	Vidējais pārsūtīto pakešu skaits no avota adreses

Pazīmes Nr.	Apraksts
F5	Pakešu dispersija dalīta ar 100000, kas pārsūtītas no avota adreses
F6	Kopējais pārsūtīto baitu skaits no avota adreses
F7	Minimālais baitu skaits plūsmā, kas pārsūtīti no avota adreses
F8	Maksimālais baitu skaits plūsmā, kas pārsūtīti no avota adreses
F9	Vidējais pārsūtīto baitu skaits no avota adreses
F10	Pārsūtīto baitu dispersija no avota adreses
F11	Kopējais pārsūtīto plūsmu skaits no avota adreses
F12	Uz konkrēto avota adresi pārsūtīto pakešu kopējais skaits
F13	Minimālais pakešu skaits vienā plūsmā, kas pārsūtīta uz avota adresi
F14	Maksimālais pakešu skaits vienā plūsmā, kas pārsūtīta uz avota adresi
F15	Vidējais uz avota adresi pārsūtīto pakešu skaits
F16	Pakešu dispersija, dalīta ar 100000, pārsūtīta uz avota adresi
F17	Uz avota adresi pārsūtīto baitu kopējais skaits
F18	Minimālais baitu skaits vienā plūsmā, kas pārsūtīta uz avota adresi
F19	Maksimālais baitu skaits vienā plūsmā, kas pārsūtīta uz avota adresi
F20	Vidējais baitu skaits, kas pārsūtīti uz avota adresi
F21	Uz avota adresi pārsūtīto baitu dispersija
F22	Kopējais plūsmu skaits, kas pārsūtītas uz avota adresi
F23	Kopējais plūsmu skaits
F24	Pārsūtīto plūsmu procentuālā daļa pēc avota adreses
F25	Uz avota adresi pārsūtīto plūsmu procentuālā daļa
F26	Unikālo IP adrešu skaits, ar kurām savienota avota adrese
F27	Unikālo IP adrešu skaits, kas savienotas ar avota adresi
F28	Unikālu avota portu skaits, ko izmanto avota adrese
F29	Unikālu galamērķa portu skaits, ko izmanto avota adrese
F30	To avota portu procentuālā daļa, kuru vērtība pārsniedz 1024, izmantojot avota adresi
F31	To avota portu procentuālā daļa, kuru vērtība ir mazāka par 1024, ko izmanto avota adrese

Pazīmes Nr.	Apraksts
F32	Procentuālais galamērķa portu skaits, kas pārsniedz 1024, ko izmanto avota adrese
F33	Procentuālais galamērķa portu skaits, kas zemāks par 1024, ko izmanto avota adrese
F34	*UDP* protokola procentuālā daļa, ko izmanto avota adrese
F35	*TCP* protokola procentuālā daļa, ko izmanto avota adrese
F36	*ICMP* protokola procentuālā daļa, ko izmanto avota adrese
F37	*UDP* protokola procentuālā daļa, ko izmanto, lai izveidotu savienojumu ar avota adresi
F38	*TCP* protokola procentuālā daļa, ko izmanto, lai izveidotu savienojumu ar avota adresi
F39	*ICMP* protokola procentuālā daļa, ko izmanto, lai izveidotu savienojumu ar avota adresi

5.2.2 Modeļa veidošana un rezultāti

Ņemot vērā [60] un praktiskos eksperimentus ar datiem tika izvēlēti četri klasifikatori: Lēmumu meži (*RFC*), Lēmumu koki (*DTC*), Neironu tīkli (*NNC*), un K-tuvākie kaimiņi (KNN). Katrai apmācību kopai tika veikta 10.pakāpju šķērsvalidācija, izmantojot *Python* sklearn bibliotēku. Sākuma precizitātes novērtējums attēlots 5.9.tabulā.

5.9.tabula

Sākotnējās precizitātes novērtējums.

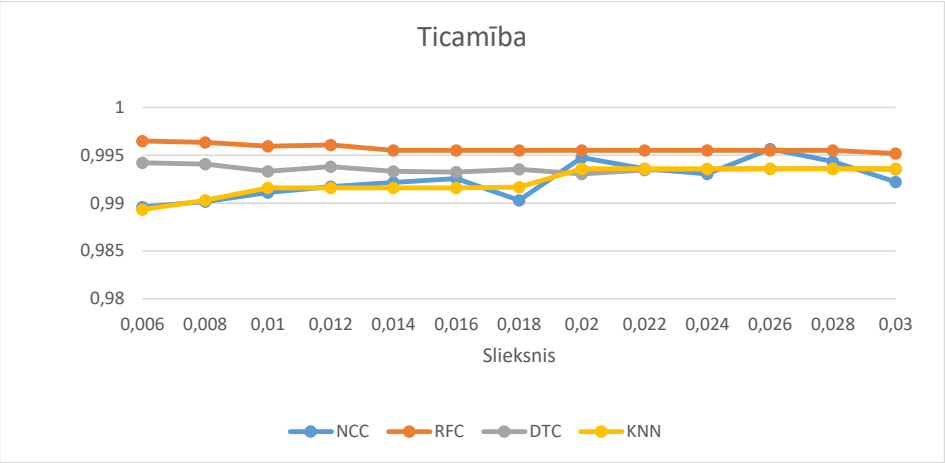
Klaisifikators	Precizitāte	Pārklājums	F1-mērs	Ticamība
DTC	0,941	0,946	0,936	0,992
RFC	0,964	0,938	0,950	0,994
NNC	0,745	0,312	0,386	0,953
KNN	0,683	0,642	0,660	0,961

Rezultāti liecina, ka ar sākotnējo pazīmju kopu *NNC* un *KNN* klasifikatori darbojas slikti, savukārt *DTC* un *RFC* nodrošina salīdzinoši labus rezultātus. Lai nodrošinātu optimālu klasifikācijas algoritmu darbību, tika samazināta pazīmju kopa. Šim nolūkam tika izmantots *Python* sklearn bibliotēkā ietilpstošais *exta-trees* klasifikators, jo tas ievieš papildu nejaušības elementus salīdzinot ar tipiskiem lēmumu kokiem. Minētais klasifikators ļauj atrast optimālo pazīmju kopu apmācības datiem, izmantojot *meta-estimator* funkcionalitāti. Pielietojot minēto

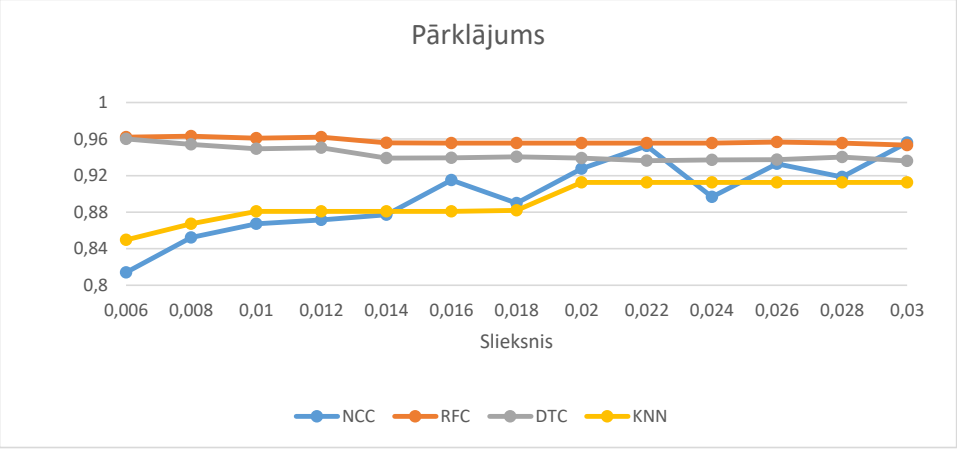
funkcionalitāti tika saņemta informācija par pazīmju nozīmīgumu lēmuma pieņemšanā, ko turpmāk var izmantot pazīmju kopas samazināšanai, atlasot nozīmīgāko pazīmju grupas. Svarīguma pazīmju slieksnis tika izvēlēts diapazonā [0,008-0,026], jo iegūtie tīkla dati iekļauj lielu trokšņu apjomu un, lai nepazaudētu pazīmes, kuras var būt noderīgas lēmuma pieņemšanā. Katram izvēlētajam slieksnim tika novērtēta pazīmju nozīmība 100 reizes. Korelācija starp izvēlēto slieksni un pazīmju kopu variantiem, ko nodrošina *extra-trees* klasifikators ir parādīta 5.3.attēlā.

5.4.-5.6. attēli parāda augstāko vidējo vērtību 10-kārtēju kopējo ticamību, pārklājumu, F1-mēru un klasifikatora precizitāti katram klasifikatoram ar vislabāko pazīmju kopu, kura tika noteikta ar *extra-trees* klasifikatora palīdzību.

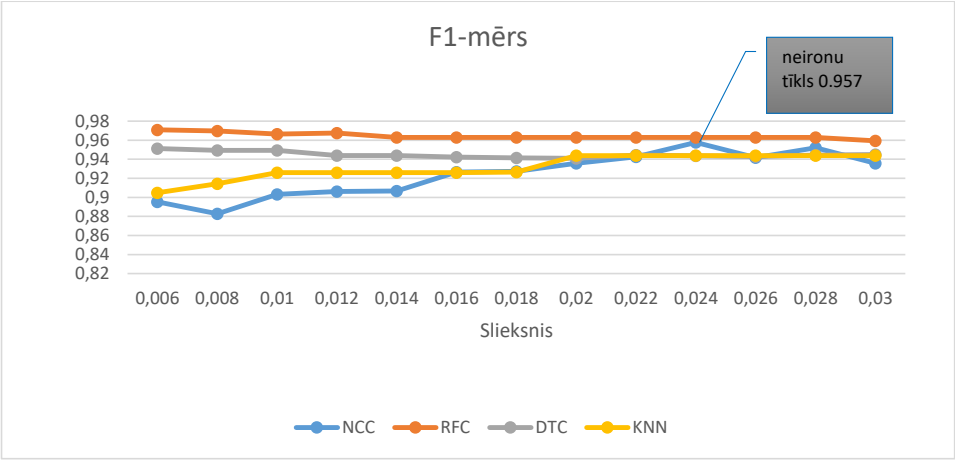
5.3. Korelācija starp izvēlēto slieksni un pazīmju kopu variantiem



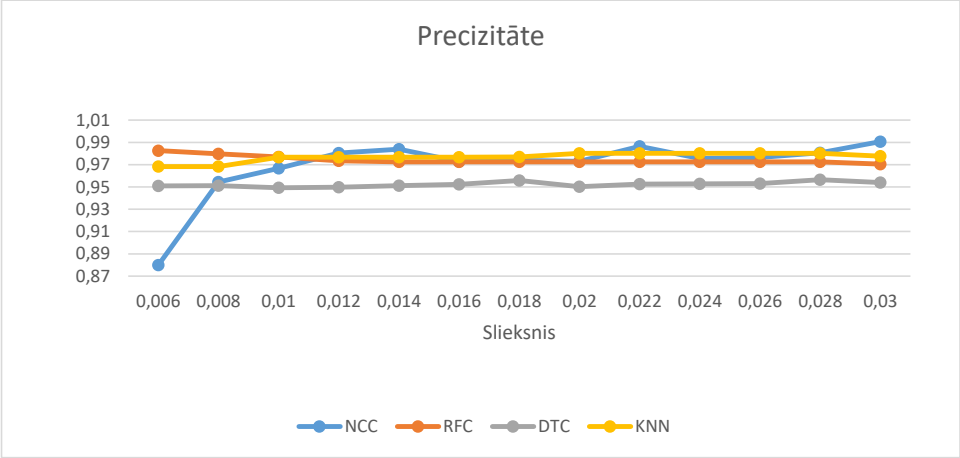
5.4.att. Augstākā 10 kārtīga ticamība



5.5.att. Augstākais 10 kārtīgais pārklājums



5.6.att. Augstākais 10 kārtīgs F1-mērs



5.7.att. Augstākā 10 kārtīgā precizitāte

Pamatojoties uz 5.4.attēlu, var secināt, ka ticamības izmaiņas visiem klasifikatoriem ir pavisam niecīgas, mainot dažādas pazīmju kopas. Papildu tam, var secināt, ka *NNC* ir ļoti jūtīgs pret neoptimāli definētām pazīmju kopām. Šoreiz labākā pazīmju kopa tika identificēta, ņemot vērā F1-mēru, kas, šajā uzdevumā, pamatojoties uz [132] ir labs veids, kā atrast balansu starp klasifikatora precizitāti un pārklājumu. Rezultātā tika iegūta sekojoša pazīmju kopa:

“F11,F24,F25,F26,F27,F28,F29,F32,F33,F34,F35,F38,F39”

10-kārtīgas šķērsvalidācijas rezultāti ir attēloti 5.10.tabulā.

5.10.tabula

Šķērsvalidācijas rezultāti

Klasifikators	Precizitāte	Pārklājums	F1-mērs	Ticamība
DTC	0,947	0,941	0,938	0,993
RFC	0,974	0,950	0,962	0,995
NNC	0,960	0,952	0,951	0,994
KNN	0,980	0,908	0,942	0,993

Šī tabula uzrāda līdzīgus rezultātus kā [48], kur *RFC* un *NNC* ir labākie klasifikatori, atbilstoši izvēlētajai pazīmju kopai. Lai pārliecinātos, ka izvēlētie modeļi ir atbilstoši reālajai videi un, tos izmantojot, par drošību atbildīgā persona būtu spējīga identificēt vēl nezināmus draudus. Visi klasifikatori tika adaptēti reālā vidē, analizējot RTU *NetFlow* datus. Dati tika iegūti no 2021.gada februāra līdz jūnijam.

5.11. tabula

NFAI darbības rezultāti

Mērījums	*DTC* (%)	*RFC* (%)	*NNC* (%)	KNN (%)
Viltus negatīvi rezultāti (NAT)	5,5	15,7	11,1	11,1
Patiesi pozitīvie rezultāti (NAT)	94,5	84,3	88,9	88,9
Viltus negatīvi rezultāti (PUBLIC)	43,2	47,7	45,4	45,4
Patiesi pozitīvie rezultāti (PUBLIC)	56,8	52,3	54,6	54,6

Rezultāti (5.11.tabula) parāda, ka, *NFAI* modulis var tikt izmantots kā papildu līdzeklis kiberdrošības draudu identificēšanā. *DTC* parādīja nedaudz labākus rezultātus kā pārējie izvēlētie klasifikatori. IP adresēm, kas izmanto *NAT*, tātad nav pa tiešo pieejamas no interneta bija labāki identificēšanas rādītāji. Rezultāti tika iegūti, ņemot vērā lietotāju atgriezenisko saiti, kur lietotājiem tika izteikts priekšlikums noskanēt ierīci ar uzticamu antivīrusu un tika lūgts sniegt atbildi vai ierīcē tika identificēta ļaunatūra. IP adreses, kuras bija pieejamas no interneta (5.11.tabula) uzrādīja daudz sliktākus rezultātus. Tas varētu būt skaidrojams ar faktu, ka apmācību kopai pārsvarā tika izmantotas *NAT* IP adreses, kā arī ar faktu, ka ārēji pieejamas IP adrešu porti bieži tiek skanēti, kas izraisīja nepareizu klasifikatora darbības rezultātu. Tika aprēķināta klasifikatoru ātrdarbība apstrādājot datus *Apache Spark* izmantojot *8 core* darbstaciju ar *32GB RAM*: *RFC* - 2 minūtes lēmuma pieņemšanai, *KNN* - 25 sekundes lēmuma pieņemšanai, *NNC* - 7 sekundes un *DTC* - 2 sekundes lēmuma pieņemšanai.

5.2.3 Secinājumi par *NFAI* moduli

Piedāvātā *NFAI* moduļa priekšrocība ir spēja identificēt ļaunprātīgā koda aktivitāti, izmantojot *NetFlow* tīkla datus. *NetFlow* tīkla dati ir noderīgs papildu avots kopējās informācijas sistēmu drošības uzlabošanai. Šo datu analīze un apstrāde paver vēl nebijušas iespējas identificēt ļaunprātīgā koda darbību pat ja komunikācija ir šifrēta. Darba rezultāti parāda, ka Neironu tīkli ar pareizu pazīmju definēšanu var tikt izmantoti, lai identificētu ļaunprātīgā koda darbību tīklā. Tomēr eksperimentu rezultātā, nosakot klasifikatoru precizitātes rādītājus, kā arī to ātrdarbību, autoraprāt labākais klasifikators ļaunprātīgas darbības identificēšanai izmantojot *NetFlow* datus ir *DTC*. Lai arī *DTC* eksperimentu rezultātā uzrādīja labākus rezultātus, tomēr, lai sasniegtu vēl labākus rezultātus līdzīgi kā [133], autors iesaka izmantot dinamisko pazīmju izvēli un saskaņā ar [48], izmantot vismaz divus dažādus klasifikatorus, piemēram neironu tīklus un lēmumu kokus. Integrējot *NFAI* moduli *ISMS* platformā, tika panākti sekojoši uzlabojumi:

1) tika samazinātas viltus trauksmes;
2) uzlabojās ļaunprātīgā koda identificēšanas iespēja;
3) tika identificēta antivīrusu kompānijām vēl nezināma ļaunatūra.

Adaptējot ansambļa apmācības metodes, var tikt sasniegti vēl labāki algoritma precizitātes rādītāji. Kopš *NFAI* modulis tika integrēts *ISMS* platformā, tika identificēti un apstiprināti 6 ļaunatūras varianti, kuri vēl nebija zināmi ugunsmūrim un *IDS* sistēmai.

5.3 DRAUDU AGREGĀCIJAS MODUĻA NOVĒRTĒJUMS

Draudu agregācijas pamatā ir dažādu moduļu rezultātu apvienošana ar mērķi identificēt inficētu ierīci. Šim mērķim pamatā tika izmantota Influx DB datubāze. Lai veiktu draudu agregāciju, tika izstrādāti scenāriji, kuri, īstenojoties ģenerē noteiktas prioritātes ziņojumu. Atkarībā no ziņojuma prioritātes, ierīces lietotājs par to tiek informēts, vai arī tiek liegta ierīces piekļuve tīklam. Ņemot vērā dinamisko IP adrešu izsniegšanu, lai datu agregācijas modulis spētu pilnvērtīgi darboties nepieciešama IP adreses sasaiste ar ierīces MAC adresi. Kad šāda sasaiste tika veikta, ierīces MAC adrese tika uzskatīta par unikālu vērtību un draudu agregācija notika pēc drauda kritiskuma, piemēram, nosakot, ka noteikts skaits ar vidējas prioritātes ziņojumiem tiek uzskatīts par draudu. Tāpat draudus iespējams agregēt pēc draudu kategorijas, piemēram, ja tiek identificēts *DGA* un vēl citi vidējas prioritātes ziņojumi noteiktā laika intervālā, tas tiek uzskatīts par draudu. Piemēram, datu agregācija attiecībā uz

10 DGA:bqpddrhdno.com

10 DGA:onogqdnghj.com

9 DGA:mhdgjcfgbd.com

9 DGA:ijraobngqd.com

9 DGA:fqjoqgmbdf.com

9 DGA:hmbhcqmrqm.com

9 DGA:nccbeqqqjp.com

8 DGA:aaojqoimhj.com

8 DGA:ppjiofgbei.com

8 DGA:cmeoieqrdb.com

8 DGA:inqmjbbgnn.com

8 DGA:iggjeegrhp.com

7 DGA:ngqajnrqgo.com

7 DGA:odbebqjfob.com

7 DGA:cbqioaaqec.com

7 DGA:irqrbdmjig.com

5.8.att. Datu agregācija pēc *DNS* ierakstiem

DNS ierakstiem, kas var tikt uzskatīti par algoritmiski ģenerētiem, ir attēlota 5.8.attēlā, kur pirmais cipars norāda uz izsaukumu skaitu mērāmajā periodā.

23 Mn-351875, Threat Spotlight_ MenuPass_QuasarRAT Backdoor [DST-IP: 2.0.0.0]

16 Non-RFC Compliant DNS Traffic on Port 53/5353_informational

6 Mn-455834, DigitalSide Malware report: MD5: 9b6c3518a91d23ed77504b5416bfb5b3 [DST-IP: 121.130.64.13]

6 DGA:tuuxptxnbw.top

4 DGA:blmuvrxoqql.com

4 DGA:qirkueafj.com

4 Mn-358938, Android Botnet IOC [DST-IP: 208.95.112.1]

2 Mn-305144, IXR_2023_LS-2531_1 [DST-IP: 64.91.248.15]

1 Mb-269521, Nueva actividad relacionada con la botnet Mozi [DST-IP: 82.221.103.244]

5.9.att. Datu agregācija izmantojot dažādas draudu avotu kopas

Attēls 5.9. parāda draudu agregāciju izmantot dažādas draudu avotu kopas, piemēram *IDS* auditācijas pierakstiem, ugunsmūra datiem, kā arī aloritmiski ģenerētiem domēnu vārdiem.

5.3.1 Secinājumi par draudu agregācijas moduli

Draudu agregācijas moduļa galvenais uzdevums ir samazināt viltus pozitīvo ziņojumu skaitu. Dažkārt mērķētu reklāmu atskaņošanai tiek izmantoti algoritmiski ģenerēti domēni, tāpēc tūlītēji uzskatīt šāda domēna izsaukšanu par draudu būtu nepareizi. Arī dažādu melno sarakstu IP adreses ne vienmēr būtu uzskatāmas par tiešu drauda norādi, jo bieži vien drošības kompānijas nepārliecinās, vai melnajā sarakstā nonākusī IP adrese nav dinamiski piešķirta, un, ja tas tā ir, tad nākamais klients, kurš saņēma šo dinamisko IP adresi arī tiks bloķēts. Draudu agregācijas moduļa ģenerētais ziņojums tiek uzskatīts par tipisku ziņojumu par identificētiem draudiem, nosūtot ierīces lietotājam ziņojumu e-pastā. Šāda draudu agregācija var tikt īstenota arī uz platformas ģenerētiem ziņojumiem, padarot šo ziņojumu apkopojumu par augstākas prioritātes draudu.

5.4 KOPĒJAIS DROŠĪBAS PĀRVALDĪBAS PLATFORMAS NOVĒRTĒJUMS

Lai novērtētu *ISMS* platformu, tas tika darīts vairākos posmos un izmantojot dažādas pieejas, piemēram, analizējot gan kopējo platformas sniegumu, gan veicot lietotāju reaģēšanas ātruma mērījumus, kā arī novērtējot atsevišķus platformas moduļus.

5.12.tabula

Novērtējuma struktūra		
Nr.	Platformas novērtējuma apraksts	Sagaidāmais rezultāts
1.	RTU IP adrešu un portu skanētāju identificēšanas moduļa novērtējums	Spēja identificēt un bloķēt RTU IP adrešu un portu skenētājus, ieliekot tos karantīnā uz noteiktu laiku. Karantīnas laiks var tikt dinamiski palielināts, ja atkārtoti tiek identificētas skanētāju IP adreses
2.	Platformas salīdzināšana ar *Palo Alto* ugunsmūra iebūvēto *IDS* funkcionalitāti	Salīdzināt darbības rezultātus attiecībā uz identificētiem apdraudējumiem, kuri atzīmēti kā svarīgi un tika automātiski apstrādāti
3.	Lietotāju reakcijas laiks saņemot automātisku ziņojumu no platformas	Novērtēt ziņojumu efktivitāti attiecībā no ziņojuma piegādes veida
4.	Platformas darbības efektivitātes novērtējums balstoties uz lietotāju atgriezenisko saiti	Novērtēt platformas efektivitāti saņemot agriezenisko saiti no iesaistītajiem lietotājiem
5.	Platformas efektivitātes novērtējums baltoties uz trešo pušu ziņojumiem par inficētām ierīcēm RTU tīklā	Identificēt *CERT* brīdinājumu par ificētām ierīcēm dinamiku ieviešot platformu RTU
6.	Gartner Magic quadrant top līderu salīdzinājums ar *ISMS* platformu	Salīdzināt citus drošības risinājumus ar *ISMS* platformu

Kā pirmais tika novērtēts RTU IP adrešu un portu skanētāju identificēšanas modulis, kura darbība tika vērtēta par 2023.gadu (no 1.janvāra līdz 18.maijam). Modulis kopumā bija identificējis 90813 unikālas skanētāju IP adreses. Skanētāju identificēšanas moduļa galvenais uzdevums ir identificēt kura IP adrese nav iesaistīta nevienā citā komunikācijā izņemot *SYN* paketes nosūtīšanu uz dažādām RTU IP adresēm, tādējādi pārbaudot vai pieprasītais ports ir atvērts. Šīs identificētās adreses uz noteiktu laiku tika automātiski iekļautas ugunsmūra IP

adrešu sarakstā, kurām ir liegta komunikācija ar RTU. Šāda portu skanētāju apturēšana samazina iespējamību, ka kāds no skanētājiem spēs identificēt ievainojamību tīklā un to veiksmīgi izmantos. Ņemot vērā lielo izglītības iestādes publisko adrešu skaitu šāds preventīvs pasākums, autoraprāt, ir uzskatāms par ļoti noderīgu.

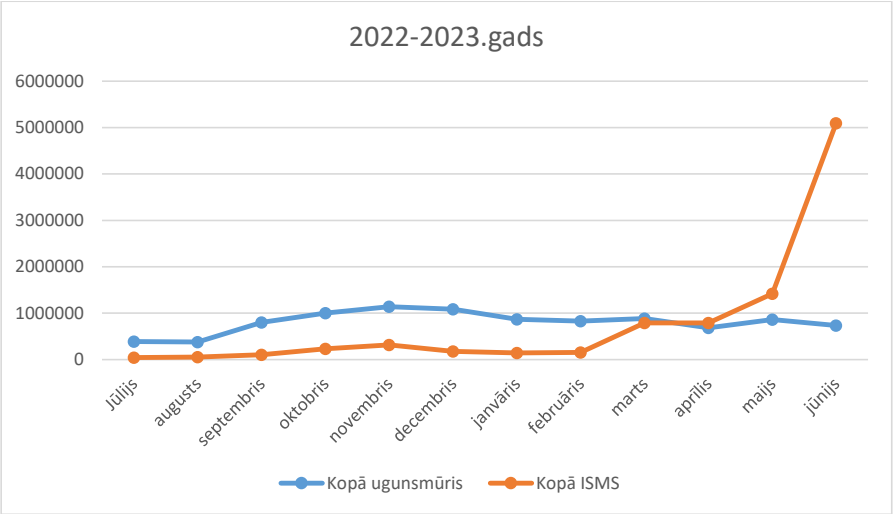
Lai novērtētu izstrādātās platformas efektivitāti, tā tika salīdzināta ar *Palo Alto* ugunsmūri ar *IDS* funkcionalitāti. Salīdzinājums tika veikts par platformas 1 gada darbību kopš 2022.gada jūlija līdz 2023.gada jūlijam (5.13.att.).

5.13.tabula

Platformas darbības salīdzinājums ar ugunsmūri

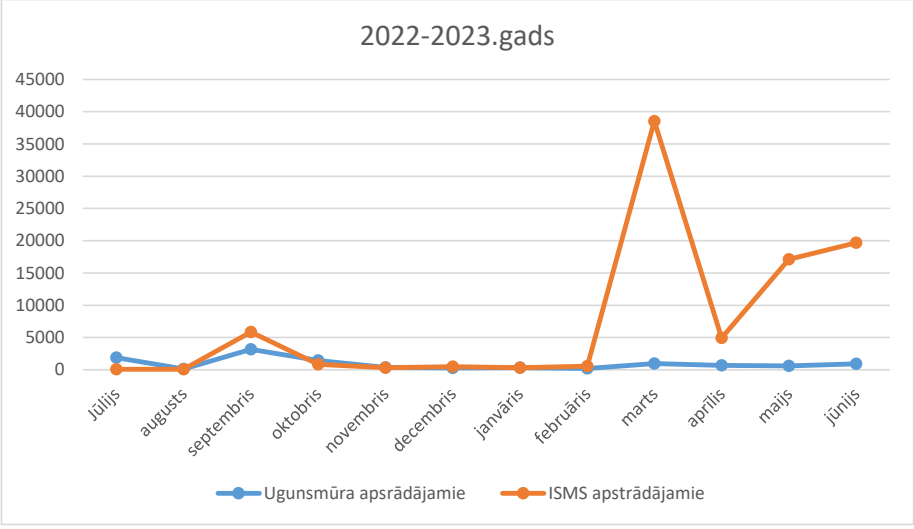
	Next Gen ugunsmūris	*ISMS* plaftorma (Bez ugunsmūra datiem)
Kopā identificēti apdraudējumi	9671288	9560316
Identificēti apdraudējumi, kuri atzīmēti kā svarīgi un tika automātiski apstrādāti	10882	88735

Tāpat tika salīdzināti *ISMS* platformas dati ar *Palo Alto* ugunsmūrī esošo *IPS* sistēmu. Dati tika salīdzināti par vienpadsmit mēnešu periodu no 2022.gada 28.jūnija līdz 2023.gada 27.maijam. Kopā *ISMS* platforma ir ģenerējusi 13milj ierakstu par šo periodu. Par augstākminēto periodu, ugunsmūra IPS inficēto ierīču statisktika salīdzinājumā ar *ISMS* (bez ugunsmūra IPS datiem) platformu, parāda, ka *ISMS* pareizi identificēja 96,4% gadījumos inficētu ierīci. Kas attiecas uz *DGA*, tad salīdzinot ugunsmūra identificēto ierīci ar *ISMS* platformu, tika iegūts 97,4% gadījumos tāds pats rezultāts. Salīdzinot *ISMS* platformas ierīces, kurām ir aizdomas par *DGA*, tikai 20% gadījumu ugunsmūris uzrādīja to pašu ierīci kā *DGA* aizdomīgu. Tas var būt skaidrojams ar faktu, ka *ISMS* platformā ir nepieciešams samazināt viltus pozitīvos *DGA*, kā arī to, ka, iespējams, mašīnmācīšanās modulis uzrāda labākus rezultātus par ugunsmūra IPS. Kopējie identificētie apdraudējumi pa mēnešiem ir parādīti 5.10. attēlā. Te iezīmējas tendence, ka *ISMS* platformas ģenerēto drošības paziņojumu skaits kopš 2023.gada marta pārsniedza ugunsmūra ģenerēto paziņojumu skaitu. Tas skaidrojams ar papildu moduļa – *CERT* agrās brīdināšanas sistēmas pievienošana *ISMS* platformai. Pēdējā mēneša (2023.gada jūnija) dati (5.10.att.) norāda uz to, ka *CERT* agrās brīdināšanas sistēma tika papildināta ar ļaunprātīgas darbības identificēšanas mehānismiem.



5.10.att. *ISMS* platformas un ugunsmūra ģenerēto kopējo paziņojumu skaits

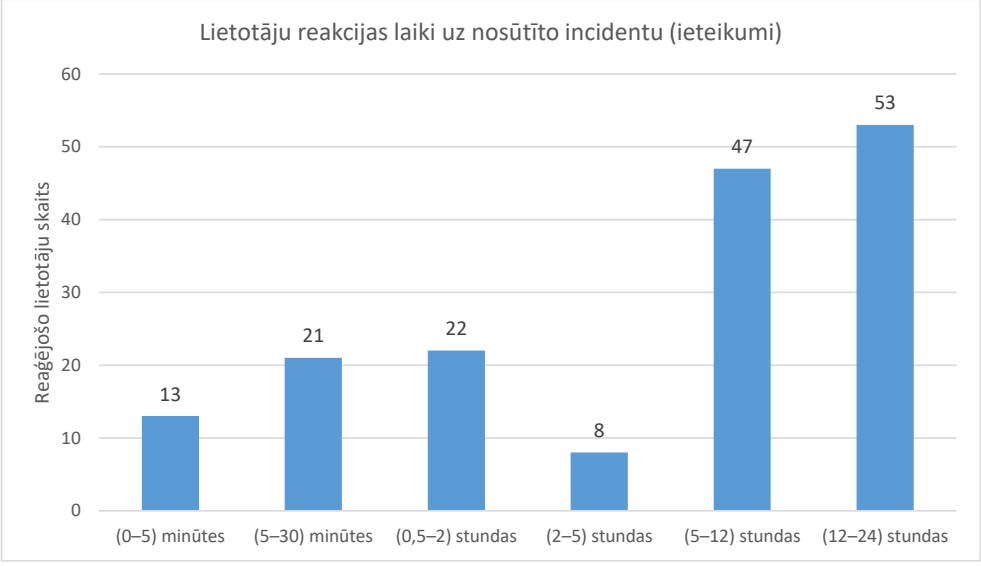
Tas, ka kopējo ziņojumu skaits pārsniedz 5milj. mēnesī nenozīmē, ka visi ziņojumi ir uzskatāmi par ļaunprātīgā koda vai cilvēka darbību. Tāpēc *ISMS* platforma tika aprīkota ar scenāriju datubāzi, pamatojoties uz kuru tiek atlasīti un automātiski apstrādāti platformas ziņojumi. Kā redzams 5.11.attēlā, apstrādāto ziņojumu skaits no *ISMS* platformas ģenerētajiem ziņojumiem būtiski pārsniedz apstrādājamo ziņojumu skaitu, kurus ir ģenerējis ugunsmūris.



5.11.att. *ISMS* platformas un ugunsmūra automātiski apstrādājamo ziņojumu skaits

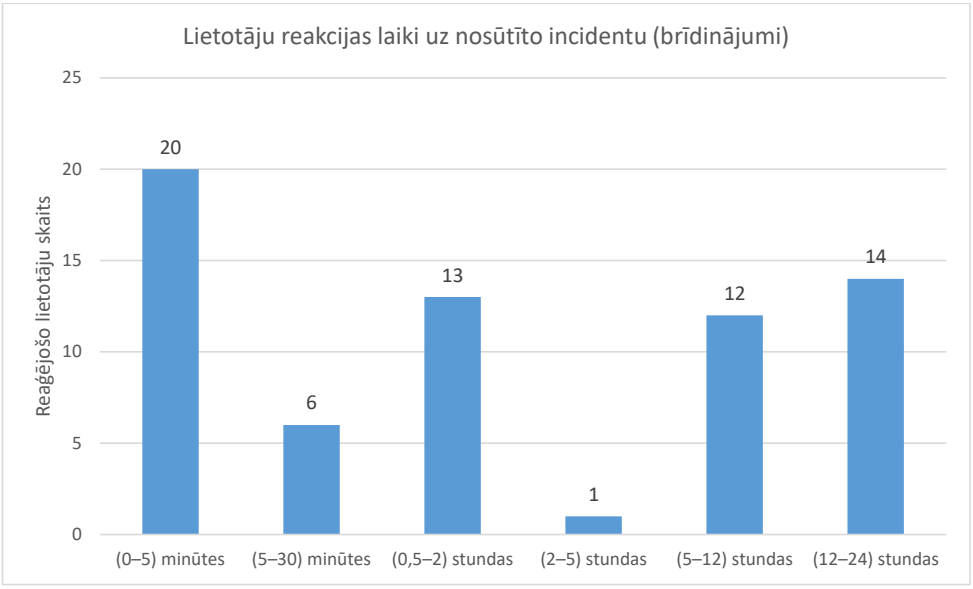
5.11.attēls uzskatāmi parāda, ka vienas drošības sistēmas izmantošana, piemēram ugunsmūra, nav pietiekama, lai identificētu arvien pieaugošos kiberdraudus. Tā kā *ISMS* platforma iekļauj arī ugunsmūri, tad tiek nodrošināta pilnvērtīgāka drošības incidentu identificēšana.

Lai identificētu iespējami labākos platformas apziņošanas paņēmienus, tika mērīts lietotāju reakcijas laiks, nosakot laika intervālu starp paziņojuma nosūtīšanu un RTU portāla apmeklējumu. Ziņojumā tika rekomendēts apmeklēt portālu, kurā sniegta papildu informācijā par incidentu, ieskaitot incidentā iesaistītās IP adreses, kā arī citu papildinformāciju. Lietotāju reakcijas laiks uz ieteikumiem, kur informācijas tika sūtīta tikai e-pastā un iekļauta portālā ir redzami 5.12.attēlā.



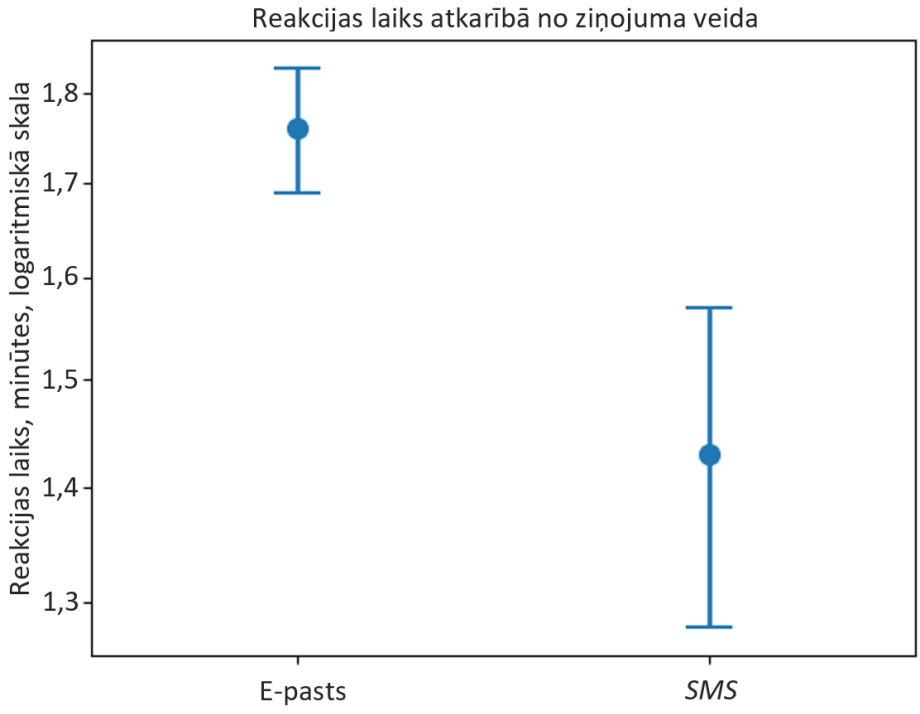
5.12.Reakcijas laiks uz ieteikumiem

Savukārt brīdinājumi, kuros tika nosūtīta informācija, tai skaitā *SMS* veidā, parāda daudz labākus rezultātus (5.13. attēls).



5.13. Reakcijas laiks uz brīdinājumiem

Lielākā lietotāju daļa reaģēja uz paziņojumu jau 5 minūšu laikā.



5.14 Reakcijas laiks atkarībā no ziņojuma veida

Apstiprinājums tam, ka lielākā daļa lietotāju atbildēja uz *SMS* paziņojumu 5 minūšu laikā, ir parādīts intervāla diagrammā (5.14.attēls), kur attēloti vidējie atbildes laiki atbilstoši saziņas veidam.

Lietotājiem tika lūgts sniegt atgriezenisko saiti, aizpildot *MS Forms* aptaujas anketu. Šo iespēju izmantoja tikai studenti. Kopumā (no 2021.gada 1.janvāra līdz 2023.gada 18.maijam) tika aizpildīta 123 unikālas anketas. Daži lietotāji aizpildīja anketu atkārtoti, tādēļ atkārtotās reizes netika ņemtas vērā. Atbildēs tika nodalīti Latvijas un ārvalstu studenti, jo pēc platformas datiem ārvalstu studentu ierīces daudz biežāk tika inficētas (5.14.tabula).

5.14.tabula

Lietotāju sniegtās atsauksmes

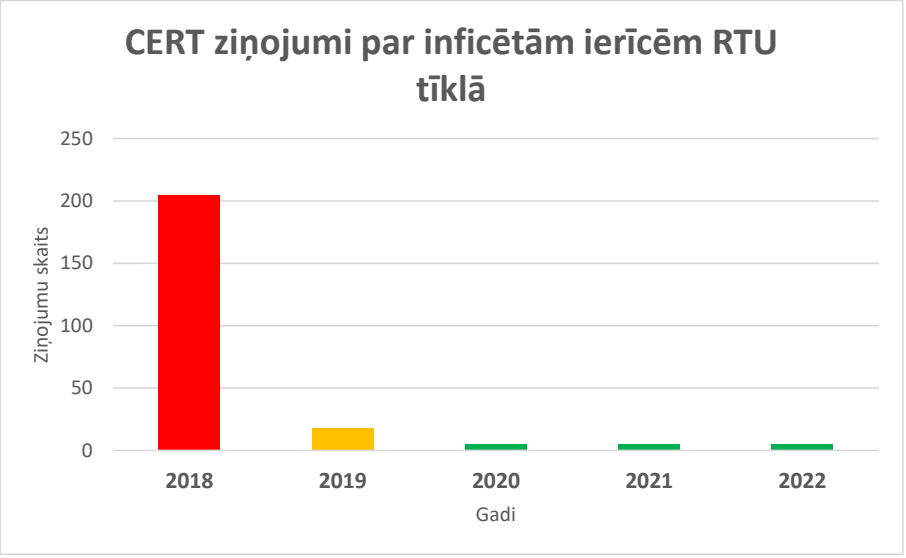
	Ārvalstu studenti	Latvijas studenti	**Kopā**
Kopā sniegtas atsauksmes	32	91	123
Sniegta atsauksme, ka ierīce patiešām bija inficēta	12	16	38

Rezultāti (5.14.tabula) apliecina, ka *ISMS* platformas izmantošana ir pasargājusi 38 lietotājus no draudiem, kas varēja rasties inficētas ierīces gadījumā (piemēram: personu datu noplūde, finanšu līdzekļu zādzība u.c.). Pēc uzkrātās statistikas datiem ārvalstu studentu vidū tika novēroti biežāki ierīču inficēšnās gadījumi, problēma ir arī tajā, ka ne visi ārvalstu studenti sniedz atsauksmes. Šie rezultāti apliecina to, ka piedāvātā *ISMS* platforma ir svarīga informācijas sistēmu drošības nodrošināšanas komponente un sniedz iespēju, ka tiks atklāta ļaunatūra, kuru standarta aizsardzības līdzekļi (kā uzstādītā antivīrusu programmatūra) neatklāj. Kopumā iepriekš minētajā laika periodā 104 reizes no RTU tīkla tika automātiski atslēgtas ierīces, liela daļa no kurām tika pieslēgtas pēc tam, kad lietotājs ir ziņojis par ierīces atbrīvošanos no ļaunprātīgās programmas. Tas apstiprina 1.tēzi, ka lai nodrošinātu efektīvu daudzdimesionālu datu analīzi drošības apdraudējumu identificēšanai, nepieciešams pielietot lielo datu tehnoloģijas un mašīnmācīšanās metodes.

Balstoties uz *CERT.LV* [23] (5.15.attēls) brīdinājumu dinamiku kopš 2018.gada, var apstiprināt 2.tēzi "Informācijas sistēmu drošības pārvaldības efektivitāte ir atkarīga no spējas identificēt draudus un reakcijas laika pēc drauda identificēšanas" un 3.tēzi "Lai nodrošinātu adekvātu drošības pārvaldību, nepieciešams izmantot automatizētas sistēmas, kuras reaģē uz drošības draudiem". Brīdinājumu skaits ievērojami samazinājās pēc tam, kad drošības incidenta

novēršanā tika iesaistīts lietotājs un automātiski no tīkla tika atvienotas inficētās ierīces, tādējādi būtiski saīsinot laiku, kurā drauds atradās organizācijas vidē.

Lai identificētu inficētas ierīces organizācijās *CERT.LV* ir izvietojis sensorus ar kuru palīdzību tiek analizēti tīkla dati Latvijas mērogā. 2023.gadā *CERT.LV* ir izvietojis atsevišķu agrās brīdināšanas sistēmu RTU telpās, kura arī tika integrēta kopējā *ISMS* platformā, tādēļ 5.15.attēlā redzamā statistika ir līdz 2022.gadam.



5.15.att. Ļaunprātīgā koda izplatības dinamika ieviešot *ISMS* platformu

Ņemot vērā visu augstākminēto autors apstiprina hipotēzi, ka apvienojot vairākus datu avotus, specializētus draudu identificēšanas modeļus un platformas, tiek iegūta pilnvērtīgāka drošības incidentu identificēšana, salīdzinot ar individuālu šim mērķim paredzētu risinājumu izmantošanu. Bezaģentu *ISMS* platforma ir īpaši aktuāli tādēļ, ka lielākā daļa no apskatītajiem līdzīgiem risinājumiem [134] [135] [136] piedāvā izmantot un uzstādīt aģentus uz galaiekārtām, kas, piemēram, attiecībā uz studentiem nevar tikt īstenota. Lai gan gadījumos, kad galaiekārtas ir organizācijas pārvaldībā, un ir iespējams uzstādīt tajās aģentus, platforma var palīdzēt sasniegt vēl labākus rezultātus drošības pārvaldības nodrošināšanā. Tā kā autora piedāvātā *ISMS* platforma var tikt uzstādīta vietās, kur organizācijai nav iespējas uzstādīt aģentus galaiekārtās, piemēram augstākās izglītības iestādes, interneta servisa pakalpojuma nodrošinātājos u.c., tad šīs organizācijas būtu ieguvējas no platformas darbināšanas, pārraugot datu tīklu un laicīgi reaģējot uz apdraudējumiem, kā arī pasargājot gala lietotājus no nonākšanas robotu tīklos. Otrs iemesls organizācijai būvēt un izmantot *ISMS* platformu ir apskatīto *SOC*

augstā cena un saskaņā ar 2.7.tabulu organizācijai pašai uzbūvēt informācijas sistēmu drošības pārvaldības platformu ir izdevīgāk kā pirkt jau gatavu risinājumu. Rezultatīvie rādītāji (5.15.attēls) uzrāda to, ka šāda platforma ir spējīga preventīvi novērst kiberdrošības apdraudējumus.

Tabulā 5.15. parādīta *Gartner Magic Quadrant 2021* [129] līdzīgu risinājumu top līderu salīdzinājums ar *ISMS*.

5.15.tabula

Gartner Magic quadrant top līderu salīdzinājums ar *ISMS* platformu				
Nr.p.k	**Funkcionalitāte**	**IBM Q Radar**	**SPLUNK**	**IS drošības pārvaldības platforma *ISMS***
1.	Reāla laika (vai tuvu reālam laikam) monitorēšana	+	+	+
2.	Draudu intelektuāla noteikšana	+	+	+
3.	Lietotāju aktivitāšu uzraudzība	+	+	+
4.	Uzstādīšanas iespējas	Mākonī vai specifiska iekārta	Mākonī vai programmatūra uz klienta iekārtas	Mākonī, uz atsevišķas iekārtas vai uz klienta iekārtas
5.	Mākslīgā intelekta pielietošana robottīklu identificēšanai	(APT detekcija)	-	+
6.	Platforma	Privāta	Privāta	Uz atvērto kodu balstīta

7.	Specifiku organizācijai domātu moduļu pievienošana	-	- (iespēja konfigurēt saņemtos auditācijas pierakstus)	+
8.	Mērogojamība	Mākonis	Mākonis	Nav ierobežojumu
9.	Automatizēta par resursu atbildīgā lietotāja iesaistīšana draudu gadījumā	- (par drošību atbildīgās personas informēšana)	- (par drošību atbildīgās personas informēšana)	+ (SMS/e-pasts/lietotāju portāls)
10.	Automatizēta palīdzība par resursu atbildīgajam darbiniekam drošības incidenta risināšanā	-	-	+ (SMS/e-pasts/lietotāju portāls)
11.	Mākslīgā intelekta pielietošana agregēto tīkla datu analīzē	-	-	+
12.	Incidentu pārvaldība	Izmantojot par IS drošību atbildīgo personu	Izmantojot par IS drošību atbildīgo personu	Automatizēta, iesaistot par resursu atbildīgo darbinieku
13.	Papildmoduļu uzstādīšana	Nepieciešama izstrādātāja programmētāju iesaiste, notiek lēni	Nepieciešama izstrādātāja programmētāju iesaiste, notiek lēni	Papildmoduļus ir iespējams pieprogrammēt jebkurā programmēšanas valodā, notiek ātri

Kopumā var secināt, ka *ISMS* platforma var sacensties ar tirgū piedāvātiem kiberdrošības risinājumiem. Labākus rezultātus *ISMS* platforma uzrāda gadījumos, kad tiek izmantoti vairāki datu avoti. Bet tā kā ugunsmūris ir arī integrēts *ISMS* platformā, var secināt, ka šāds

apvienojums padara iestādes iekšējo tīklu pārredzamāku un incidenta identificēšana, kā arī reaģēšana uz to notiks maksimāli īsā laikā.

REZULTĀTI UN SECINĀJUMI

Promocijas darba galvenais mērķis bija izstrādāt spējā metodoloģijā balstītu, no konteksta atkarīgu adaptīvu drošības pārvaldības modeli un atbilstošus tehniskos risinājumus kiberdrošības vides uzlabošanai, samazinot kiberdrošības incidentu skaitu un uzlabojot reaģēšanas ātrumu kiberincidenta gadījumā. Papildus galvenajam mērķim tika izveidota un aprobēta *ISMS* platforma, kā arī novērtēti tās darbības rezultāti. Mērķa sasniegšanai iegūti šādi rezultāti:

1) veikta analīze un novērtēta esošā situācija IS drošības pārvaldības jomā;
2) apzināti esošie pētījumi un izpētītas esošo drošības pārvaldības rīku iespējas, kā arī izpētīti pakalpojumi, kas nodrošina drošības pārvaldību;
3) identificētas nepieciešamās kontroles, ko var automatizēt, lai nodrošinātu IS drošības pārvaldību uzņēmumā;
4) izstrādāta vispārējā drošības pārvaldības spēja un augsta līmeņa tehniskā arhitektūra *ISMS* platformai;
5) izstrādāts *ISMS* platformas spēju modelis un identificēti nepieciešamie moduļi *ISMS* platformas darbībai;
6) definēta *ISMS* platformas implementācija;
7) adaptēts *ISMS* platformas spēju modelis RTU prasībām;
8) novērtēta *ISMS* platformas *ISMS* darbības efektivitāte;
9) sniegti secinājumi un priekšlikumi IS drošības pārvaldības uzlabošanai.

Izstrādātā *ISMS* platforma Rīgas Tehniskajā universitātē tika ieviesta kopš 2016. gada, papildinot to ar dažādiem moduļiem. Pirmsākumā tā bija tikai *IDS* sistēma ar automatizētu iespēju paziņot lietotājam par identificēto inficēto ierīci. Sākotnēji tas palīdzēja nākamajā gadā pēc ieviešanas vairāk nekā uz pusi samazināt *CERT* paziņojumu skaitu par inficētām ierīcēm RTU tīklā. Pēc tehnoloģiskā risinājuma izveides (*ISMS*) situācija uzlabojās vēl vairāk. Tika ieviesta spēja automātiski bloķēt ierīci, kuras lietotājs nereaģē uz paziņojumiem. Pakāpeniski, ieviešot papildu *ISMS* platformas moduļus, samazinājās *CERT* ziņojumu skaits par ificētajām ierīcēm RTU tīklā (5.15.att.). Ieviešot paroles zādzības identificēšanas moduli, tika konstatēts, ka visbiežāk paroļu zagļi mēģina izmantot leģitīmu lietotāju kontus, lai apietu e-pasta servera aizsardzības mehānismu pret mēstulēm, kā arī tika konstatēts, ka visbiežāk šie ļaundari izmanto Āfrikas kontinenta IP adrešu apgabalu. Ļaundari arvien uzlabo savas prasmes, un mūsdienās piķšķerēšanas e-pasts ir ļoti grūti atšķirams no leģitīma e-pasta, var tikt izmantoti arī *html* faili kā e-pasta pielikums. Piemēram, viena no veiksmīgākajām autora novērotajām piķšķerēšanas

kampaņām bija zagtas lietotāja paroles izmantošana, kā arī jau atsūtītas vēstules izmantošana, nosūtot atbildi uz to ar saiti, kurā it kā tiek kopīgots dokuments, lai to atvērtu, ir nepieciešams autentificēties viltus *M365* portālā.

Ieviešot *DGA* moduli *ISMS* platformā, tika atklātas robotu tīklā esošas ierīces, piemēram, stāvvietas vārtiņu atvēršanas kontrolieris, kas pēc tā uzstādīšanas netika uzturēts, un ierīce tika inficēta. Lai arī mūsdienās *DNS* ir iespējams slēpt, izmantojot gan *DNS over HTTP*, gan arī *DNS over TLS* metodes, tomēr ir novērots, ka ļaunprātīgu programmu veidotāji šādas metodes izmanto reti. *DNS* moduļa ieviešana palīdzēja būtiski samazināt *CERT* paziņojumu skaitu, jo daudzas robotu tīklā esošas iekārtas uzvedās ļoti piesardzīgi un tās varēja identificēt tikai pēc *DNS* pieprasījuma. Darbā tika salīdzināts *DGA* modulis ar vienu no labākajiem ugunsmūriem *Gartner* kvadranta ieskatā [129], kuram ražotājs bija paredzējis *DGA* identificēšanas funkcionalitāti. *DGA* modulis identificēja ļaunprātīgu pieprasījumu 92,4 % gadījumu, salīdzinot ar iepriekšminēto ugunsmūri. *DGA* modulis 8,5 % gadījumu ir atklājis ļaunatūru, ko ugunsmūris nav atklājis, bet to ir apstiprinājusi vismaz viena no trim trešām pusēm (*Cloudfare* [130], *Norton* [131] vai *Quad9* [80]).

Lai arī *NFAI* modulis ne vienmēr darbojās labi, tomēr tika identificētas inficētas ierīces tīklā pat tādos gadījumos, kad tika lietota *DNS* pieprasījumu šifrēšana, kā arī visas datu plūsmas šifrēšana. Atbilstoši sniegtajām lietotāju atsaucēm kopumā tika identificēti seši ļaunatūras varianti, kas vēl nebija zināmi ugunsmūrim un *IDS* sistēmai. Arī "meduspoda" modulis ir devis pozitīvus rezultātus, identificējot inficētas ierīces laikā, kad tās iesāk tīkla ierīču skanēšanu un meklē jaunus potenciālos upurus.

Apvienojot *ISMS* platformā dažādus moduļus, tika samazināti viltus trauksmes ziņojumi.

Iesaistot gala lietotājus incidenta novēršanas procesā, kā arī spējot atslēgt ierīci, tika samazināta iespējamība, ka inficētā ierīce inficēs citas tīklā esošas ierīces. Īpaši tas ir aktuāli gadījumos, kur inficētā ierīce nav RTU pārvaldībā (piemēram, studentu ierīces).

Papildus RTU, *ISMS* ir ieviests arī divās Latvijas valsts iestādēs – Centrālajā finanšu un līgumu aģentūrā (CFLA) un Iepirkumu uzraudzības birojā (IUB). 6.1. tabulā apkopoti galvenie novērojumi par *ISMS* izmantošanu minētajās organizācijās. CFLA aizstāja *Splunk* drošības informācijas un notikumu pārvaldības (*SIEM*) sistēmu ar *ISMS*, kas licenču maksās ļāva ietaupīt aptuveni 50 000 EUR gadā. Šie ietaupījumi galvenokārt tika panākti, atceļot *Splunk* licenču maksas un samazinot maksājumus ārējiem pakalpojuma sniedzējiem, kas bija atbildīgi par *Splunk* notikumu konfigurēšanu un uzraudzību. Daudzas trešo pušu lietojumprogrammas,

piemēram, dokumentu pārvaldības sistēma IUB, ģenerē žurnālu failus, kas iepriekš tika analizēti atsevišķi šajās lietojumprogrammās. IUB dokumentu pārvaldības sistēmas žurnālu failu analīze tika integrēta *ISMS*, samazinot izmaksas, kas tika maksātas trešo pušu lietojumprogrammu izstrādātājiem.

Šo funkciju vērtību ir apstiprinājusi abu organizāciju vadība. *ISMS* tika prezentēta arī Latvijas informācijas tehnoloģiju drošības kopienai *CERT.LV* seminārā "*Be Secure*" 2023. gadā, kā arī IT drošības pasākumā "*Cyber Commando*", kas 2024. gadā notika Rīgā.

6.1.tabula

Izmantotās *ISMS* iespējas CFLA un IUB

Iespējas	**CFLA**	**IUB**
Izmaksas	Licenču izmaksu samazināšana	Izmaksu samazināšna, nepērkot dokumetu vadības sistēmas auditācijas moduli
Datu avoti	Tiek izmantoti tie paši datu avoti kā RTU gadījumā	Papildu datu avoti, piemēram, tiek pievienoti žurnāli no ārējām uzņēmuma sistēmām, nepalielinot licenču maksu
Organizācijai specifiski moduļi	Tīkla datplūsmas analīzes modulis iekšējā audita vajadzībām, lai pārbaudītu atbilstību	Modulis trešo pušu lietojumprogrammu žurnālu failu analīzei, izmantojot *ISMS*
Automatizēta lietotāju iesaiste	Lietoti iestādes specifiskie komunikācijas kanāli ziņojuma nodošanai	Lietotājiem tiek nosūtītas instrukcijas problēmas novēršanai, īpašos gadījumos tiek iesaistīti sistēmas administratori
Drošības incidentu pārvaldība	Paziņojumu veidi un automatizācijas moduļi ir pielāgoti organizācijas vajadzībām	Paziņojumu veidi un automatizācijas moduļi ir pielāgoti organizācijas vajadzībām

Nobeigumā autors secina, ka ISMS platforma ir līdzvērtīgs risinājums komerciāliem produktiem drošības pārvaldības jomā. Platforma nodrošina organizācijai nepieciešamo kiberdrošības risku identificēšanu un mazināšanu, kā arī apstiprina autora izvirzīto hipotēzi, ka, apvienojot vairākus datu avotus, specializētus draudu identificēšanas modeļus un platformas, tiek iegūta pilnvērtīgāka drošības incidentu identificēšana, salīdzinot ar individuālu šim mērķim paredzētu risinājumu izmantošanu.

Turpmākie pētījumi tiks fokusēti uz papildu moduļu izstrādi pielietojot mākslīgā intelekta metodes, kā arī tiks uzlaboti esošo modeļu rezultatīvie rādītāji.

LITERATŪRAS SARAKSTS

[1] M. o. D. o. Latvia, « Latvian Cybersecurity strategy 2019-2022,» 2019. [Tiešsaiste]. Available: http://tap.mk.gov.lv/doc/2018_11/AiMpamn_221118_PLKS.1220.docx. [Piekļūts 08 06 2020].

[2] «An official website of the European Union,» 14 12 2022. [Tiešsaiste]. Available: https://eur-lex.europa.eu/eli/dir/2022/2555. [Piekļūts 06 09 2023].

[3] CISA, «What is Cybersecurity?,» Cybersecurity & Infrastructure Security agency, 2009. [Tiešsaiste]. Available: https://us-cert.cisa.gov/ncas/tips/ST04-001. [Piekļūts 30 04 2021].

[4] MITRE, «Risk Mitigation Planning, Implementation, and Progress Monitoring,» MITRE Corporation, 2018. [Tiešsaiste]. Available: https://www.mitre.org/publications/systems-engineering-guide/acquisition-systems-engineering/risk-management/risk-mitigation-planning-implementation-and-progress-monitoring. [Piekļūts 03 05 2021].

[5] «SolarWinds hack was 'largest and most sophisticated attack' ever,» Reuters, 21 02 2021. [Tiešsaiste]. Available: https://www.reuters.com/article/us-cyber-solarwinds-microsoft-idUSKBN2AF03R. [Piekļūts 26 04 2021].

[6] «Colonial Pipeline CEO admits to authorizing $4.4 million ransomware payment,» CNN, 19 05 2021. [Tiešsaiste]. Available: https://edition.cnn.com/2021/05/19/politics/colonial-pipeline-ransom/index.html. [Piekļūts 10 06 2021].

[7] «Nedēļas nogalē atvairīti kiberuzbrukumi 70 valsts iestāžu tīmekļa vietnēm,» DIENA, [Tiešsaiste]. Available: https://www.diena.lv/raksts/latvija/zinas/nedelas-nogale-atvairiti-kiberuzbrukumi-70-valsts-iestazu-timekla-vietnem-14280778. [Piekļūts 24 05 2022].

[8] ENISA, «ENISA Threat Landscape 2020 - Botnet,» ENISA, 20 10 2020. [Tiešsaiste]. Available: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2020-botnet. [Piekļūts 30 04 2021].

[9] Saeima, «Informācijas tehnoloģiju drošības likums,» Saeima, 28 10 2010. [Tiešsaiste]. Available: https://likumi.lv/doc.php?id=220962. [Piekļūts 27 04 2021].

[10] Saeima, «Minimālās kiberdrošības prasības,» Saeima, 25 06 2025. [Tiešsaiste]. Available: https://likumi.lv/ta/id/361481-minimalas-kiberdrosibas-prasibas. [Piekļūts 27 11 2025].

[11] E. P. U. PADOME, EIROPAS PARLAMENTA UN PADOMES REGULA (ES) 2016/679 par fizisku personu aizsardzību attiecībā uz personas datu apstrādi un šādu datu brīvu apriti un ar ko atceļ Direktīvu 95/46/EK (Vispārīgā datu aizsardzības regula), Eiropas Savienības Oficiālais Vēstnesis, 2016.

[12] «Marriott reveals its second customer data breach in two years,» CBSnews, 31 03 2020. [Tiešsaiste]. Available: https://www.cbsnews.com/news/marriott-data-breach-2020-5-million/. [Piekļūts 27 04 2021].

[13] «ICO fines Marriott International Inc,» Information Commisioner's office, 30 10 2020. [Tiešsaiste]. Available: https://ico.org.uk/about-the-ico/news-and-events/news-and-blogs/2020/10/ico-fines-marriott-international-inc-184million-for-failing-to-keep-customers-personal-data-secure/. [Piekļūts 27 04 2021].

[14] «"City Bee" pēc Lietuvas klientu datu noplūdes apgalvo, ka lietotāju dati Latvijā ir drošībā,» Apollo ziņas, 17 02 2021. [Tiešsaiste]. Available: https://www.apollo.lv/7182586/city-bee-pec-lietuvas-klientu-datu-nopludes-apgalvo-ka-lietotaju-dati-latvija-ir-drosiba. [Piekļūts 27 04 2021].

[15] «Latvijā, iespējams, notikusi nekustamo īpašumu apsaimniekošanas uzņēmumu klientu datu noplūde,» Ziņu aģentūra LETA, 05 02 2021. [Tiešsaiste]. Available: https://nra.lv/latvija/338501-latvija-iespejams-notikusi-nekustamo-ipasumu-apsaimniekosanas-uznemumu-klientu-datu-noplude.htm. [Piekļūts 27 04 2021].

[16] J. P. Anderson, Computer Security Threat Monitoring and Surveillance, James P. Anderson Co., 1980.

[17] P. Scwab, «The History of Intrusion Detection Systems (IDS) – Part 1,» 09 09 2015. [Tiešsaiste]. Available: https://www.threatstack.com/blog/the-history-of-intrusion-detection-systems-ids-part-1. [Piekļūts 08 06 2020].

[18] ISACA, «State of Cybersecurity 2020,» ISACA, 2020. [Tiešsaiste]. Available: https://www.isaca.org/go/state-of-cybersecurity-2020. [Piekļūts 26 04 2021].

[19] PWC, «24th Annual Global CEO Survey,» PricewaterhouseCoopers, 2021. [Tiešsaiste]. Available: https://www.pwc.com/gx/en/ceo-agenda/ceosurvey/2021.html. [Piekļūts 26 04 2021].

[20] J. Yakencheck, «Increase Automation to Overcome Cyber Resilience Challenges,» [Tiešsaiste]. Available: https://securityintelligence.com/posts/increase-automation-to-overcome-cyber-resilience-challenges/. [Piekļūts 26 03 2020].

[21] e. planet, «Top Cybersecurity Companies,» eSecurity planet, 05 01 2021. [Tiešsaiste]. Available: https://www.esecurityplanet.com/products/top-cybersecurity-companies/. [Piekļūts 27 04 2021].

[22] N. I. o. S. a. Technology, «Framework for Improving Critical Infrastructure Cybersecurity,» 18 04 2018. [Tiešsaiste]. Available: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf. [Piekļūts 08 06 2020].

[23] «*CERT.LV*,» [Tiešsaiste]. Available: https://cert.lv/lv. [Piekļūts 08 06 2021].

[24] «SOC-as-a-Service,» Deltarisk, 2021. [Tiešsaiste]. Available: https://deltarisk.com/soc-as-a-service/. [Piekļūts 04 05 2021].

[25] R. Cybersecurity, «SOC as a Service,» Radar Cybersecurity, 2021. [Tiešsaiste]. Available: https://www.radarcs.com/service-technology/managed-security-services/. [Piekļūts 04 05 2021].

[26] At&T, «SOC as a service,» At&T, 2021. [Tiešsaiste]. Available: https://cybersecurity.att.com/solutions/security-operations-center/soc-as-a-service. [Piekļūts 04 05 2021].

[27] Profico, «SOC-as-a-Service,» Profico, 2021. [Tiešsaiste]. Available: https://www.proficio.com/soc-as-a-service/. [Piekļūts 04 05 2021].

[28] Netsurion, «Security Operations Center (SOC),» Netsurion, 2021. [Tiešsaiste]. Available: https://www.netsurion.com/managed-threat-protection/security-operations-center. [Piekļūts 04 05 2021].

[29] «SOC Report Cost,» TrustNet, 2021. [Tiešsaiste]. Available: https://www.trustnetinc.com/pricing/soc-ssae18-report-cost/. [Piekļūts 05 05 2021].

[30] «Alexa Top million domains,» Alexa, [Tiešsaiste]. Available: http://s3.amazonaws.com/alexa-static/top-1m.csv.zip. [Piekļūts 11 08 2021].

[31] «Alexa Top Sites,» Amazon, 2021. [Tiešsaiste]. Available: https://aws.amazon.com/alexa-top-sites/. [Piekļūts 07 05 2021].

[32] K. Sandkuhl un J. Stirna, Capability Management in Digital Enterprises, Springer International Publishing, 2018, p. 396.

[33] A. Kofod-Petersen, «How to do a Structured Literature Review in computer science,» ResearchGate, 2015.

[34] ISO, «Information security management (ISO/IEC 27001),» ISO, 2013. [Tiešsaiste]. Available: https://www.iso.org/isoiec-27001-information-security.html. [Piekļūts 27 04 2021].

[35] CISA, «Cybersecurity Framework,» CISA, 2021. [Tiešsaiste]. Available: https://us-cert.cisa.gov/resources/cybersecurity-framework. [Piekļūts 27 04 2021].

[36] «CIS Critical Security Controls,» Cisecurity, [Tiešsaiste]. Available: https://www.cisecurity.org/controls. [Piekļūts 27 12 2023].

[37] «SOC for Cybersecurity,» Aicpa&Cima, [Tiešsaiste]. Available: https://www.aicpa-cima.com/topic/audit-assurance/audit-and-assurance-greater-than-soc-for-cybersecurity. [Piekļūts 27 12 2023].

[38] «Cyber Essentials,» NATional Cyber Security Cetre, UK, [Tiešsaiste]. Available: https://www.ncsc.gov.uk/cyberessentials/overview. [Piekļūts 27 12 2023].

[39] *CERT.LV*, «Latvija kopā ar sabiedrotajiem veikusi IKT sistēmu draudu meklēšanas operāciju,» *CERT*, 30 11 2022. [Tiešsaiste]. Available: ka ir ārkārtīgi svarīgi nodrošināt tīkla inventarizāciju un redzamību, operētājsistēmu un izmantotās programmatūras savlaicīgus atjauninājumus, sistēmas drošības notikumu apkopošanu un uzraudzību, kā arī reaģēšanu uz incidentiem. [Piekļūts 07 12 2022].

[40] D.-b. X. Lin Li, «Research on the network security management based on data mining,» %1 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), Chengdu, China, 2010.

[41] «Blacklist Check List,» [Tiešsaiste]. Available: https://whatismyipaddress.com/blacklist-check. [Piekļūts 16 06 2021].

[42] «Analyze suspicious files and URLs to detect types of malware,» VirusTotal, 2021. [Tiešsaiste]. Available: https://www.virustotal.com/gui/home/url. [Piekļūts 07 05 2021].

[43] «Blacklists,» [Tiešsaiste]. Available: https://mxtoolbox.com/blacklists.aspx. [Piekļūts 16 06 2021].

[44] M. Sheng, Machine Learning for Computer and Cyber Security Principles, Algorithms, and Practices, New York: CRC Press Taylor & Francis Group, 2019.

[45] «Intrusion detection system,» Wikipedia, [Tiešsaiste]. Available: https://en.wikipedia.org/wiki/Intrusion_detection_system. [Piekļūts 10 08 2021].

[46] M. A. F. B. S. OUIAZZANE, «A Multi-Agent Model for Network Intrusion Detection,» %1 2019 1st International Conference on Smart Systems and Data Science (ICSSD), 2019.

[47] S. C. M. M. a. D. C. A. Shah, «Building Multiclass Classification Baselines for Anomaly-based Network Intrusion Detection Systems,» %1 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), 759-760.

[48] S. S. Sivatha Sindhu, S. Geetha un A. Kannan, «Decision tree based light weight intrusion detection using a wrapper approach,» Expert Systems with Applications, sēj. 39, nr. 1, pp. 129-141, 2012.

[49] Z. K. J. B. a. R. I. A. Sharma, «Analysis of security data from a large computing organization,» %1 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN), 506-517, 2011.

[50] N. F. A. G. E. M. a. M. P. I. Rose, «Something Is Better Than Everything: A Distributed Approach to Audit Log Anomaly Detection,» %1 IEEE Cybersecurity Development (SecDev), 2017.

[51] R. Gerhards, The Syslog Protocol.

[52] «The World’s First Truly Open Threat Intelligence Community,» Aleanvault, [Tiešsaiste]. Available: https://otx.alienvault.com/. [Piekļūts 18 06 2021].

[53] «IBM Security QRadar,» IBM, [Tiešsaiste]. Available: https://www.ibm.com/security/security-intelligence/qradar. [Piekļūts 18 06 2021].

[54] «Security Information and Event Management (SIEM),» Logrhythm, [Tiešsaiste]. Available: https://logrhythm.com/solutions/security/siem/. [Piekļūts 18 06 2021].

[55] «Splunk Enterprise Security,» Splunk , [Tiešsaiste]. Available: https://www.splunk.com/en_us/software/enterprise-security.html. [Piekļūts 18 06 2021].

[56] R. D. C. A. P. Marcello Cinque, «Contextual filtering and prioritization of computer application logs for security situational awareness,» Future Generation Computer Systems, sēj. Volume 111, nr. ISSN 0167-739X, pp. 668-680, 2020.

[57] W. Z. a. W. Xinyu, «NetFlow Based Intrusion Detection System,» %1 2008 International Conference on MultiMedia and Information Technology, 2008.

[58] «fprobe,» SourceForge, 2016. [Tiešsaiste]. Available: https://sourceforge.net/p/fprobe/wiki/Home/. [Piekļūts 13 05 2021].

[59] «nfdump,» GitHub, 2021. [Tiešsaiste]. Available: https://github.com/phaag/nfdump. [Piekļūts 13 05 2021].

[60] K. Singh, S. C. Guntuku, A. Thakur un C. Hota, Big Data Analytics framework for Peer-to-Peer Botnet detection using Random Forests, Information Sciences, 2014.

[61] «dumpcap - Dump network traffic,» Wireshark, [Tiešsaiste]. Available: https://www.wireshark.org/docs/man-pages/dumpcap.html. [Piekļūts 27 08 2021].

[62] «tshark - Dump and analyze network traffic,» Wireshark, [Tiešsaiste]. Available: https://www.wireshark.org/docs/man-pages/tshark.html. [Piekļūts 27 08 2021].

[63] «Perl,» Perl.org, [Tiešsaiste]. Available: https://www.perl.org. [Piekļūts 27 08 2021].

[64] B. B. Gupta, Machine Learning for Computer and Cyber Security Principles, Algorithms, and Practices, New York: CRC Press Taylor & Francis Group, 2018.

[65] R. D. Muhammet Baykara, «A novel honeypot based security approach for real-time intrusion detection and prevention systems,» Journal of Information Security and Applications, sēj. 41, pp. 103-116, 2018.

[66] «The Significance and Role of Firewall logs,» Exabeam, [Tiešsaiste]. Available: https://www.exabeam.com/siem-guide/siem-concepts/firewall-logs/. [Piekļūts 09 08 2021].

[67] S. K. Hajar Esmaeil As-Suhbani, «Classification of Firewall Logs Using Supervised Machine Learning Algorithms,» %1 International Journal of Computer Sciences and Engineering Vol.7, Issue.8, 2019.

[68] D. W. V. a. J. P. Sharma, «Optimized Classification of Firewall Log Data using Heterogeneous Ensemble Techniques,» %1 2021 International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2021, 2021.

[69] F. F. Daniel Plohmann, U. o. B. Khaled Yakdan, D. Michael Klatt, J. Bader un F. F. Elmar Gerhards-Padilla, «A Comprehensive Measurement Study of Domain Generating Malware,» %1 25th USENIX Security Symposium, Austin, TX, 2016.

[70] Zonefiles, «Compromised domain lis,» Zonefiles, 05 2021. [Tiešsaiste]. Available: https://zonefiles.io/compromised-domain-list/. [Piekļūts 03 05 2021].

[71] A. Ahluwalia, I. Traore, K. Ganame un N. Agarwal, «Detecting Broad Length Algorithmically Generated Domains,» Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, pp. 19-34, 2017.

[72] D.-T. Truong un G. Cheng, «Detecting domain-flux botnet based on DNS traffic features in managed network,» Security and Communication Networks, sēj. 9, nr. 14, pp. 2338-2347, 2016.

[73] E.-O. Jose Selvi, Ricardo J.Rodríguez, «Detection of algorithmically generated malicious domain names using masked N-grams,» Elsevier, sēj. Volume 124, pp. Pages 156-163, 2019.

[74] H. S. A. A. A. D. G. Jonathan Woodbridge, Predicting Domain Generation Algorithms with Long Short-Term Memory Networks, Arlington,VA 22201: Endgame, Inc, 2016, p. 13.

[75] K. R. S. Barbosa, E. Souto, E. Feitosa un K. El-Khatib, «Identifying and Classifying Suspicious Network Behavior Using Passive DNS Analysis,» 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, pp. 160-167, 2015.

[76] M. Mowbray un J. Hagen, «Finding Domain-Generation Algorithms by Looking at Length Distribution,» %1 2014 IEEE International Symposium on Software Reliability Engineering Workshops, 2014.

[77] J. Peck, C. Nie, R. Sivaguru, C. Grumer, F. Olumofin, B. Yu, A. Nascimento un M. D. Cock, «CharBot: A Simple and Effective Method for Evading *DGA* Classifiers,» IEEE Access, pp. 91759-91771, 2019.

[78] Cloudfare, «DNS over TLS,» Cloudfare, 2021. [Tiešsaiste]. Available: https://developers.cloudflare.com/1.1.1.1/dns-over-tls. [Piekļūts 03 05 2021].

[79] Mozilla.org, «About DNS-over-HTTPS,» Mozilla.org, 07 01 2020. [Tiešsaiste]. Available: https://support.mozilla.org/en-US/kb/firefox-dns-over-https#w_about-dns-over-https. [Piekļūts 07 01 2020].

[80] «An open DNS recursive service for free security and high privacy,» Quad9, 2021. [Tiešsaiste]. Available: https://quad9.com/. [Piekļūts 07 05 2021].

[81] Mcafee, «10 key functions performed by the SOC,» Mcafee, 2021. [Tiešsaiste]. Available: https://www.mcafee.com/enterprise/en-us/security-awareness/operations/what-is-soc.html#definition. [Piekļūts 03 05 2021].

[82] M. R. Long, «A Small Business Guide to the Security Operations Center,» Montley Fool, 18 11 2020. [Tiešsaiste]. Available: https://www.fool.com/the-blueprint/soc/. [Piekļūts 04 05 2021].

[83] J. Y. P. L. a. R. F. E. C. Zhong, «"Learning From Experts' Experience: Toward Automated Cyber Security Data Triage",» IEEE Systems Journal, sēj. 13, pp. 603-614, 2019.

[84] Y. Korff, «How much does it cost to build a 24x7 SOC?,» 28 02 2018. [Tiešsaiste]. Available: https://expel.io/blog/how-much-does-it-cost-to-build-a-24x7-soc/. [Piekļūts 05 05 2021].

[85] «SOAR defined,» Microsoft, [Tiešsaiste]. Available: https://www.microsoft.com/en-us/security/business/security-101/what-is-soar. [Piekļūts 23 09 2023].

[86] S. Shea, «SOAR (security orchestration, automation and response),» TechTarget, [Tiešsaiste]. Available: https://www.techtarget.com/searchsecurity/definition/SOAR. [Piekļūts 26 09 2023].

[87] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed un M. Xu, «A Survey on Machine Learning Techniques for Cyber Security in the Last Decade,» IEEE Access, sēj. 8, pp. 222310-222354, 2020.

[88] G. F. T. H. S. N. W.-S. Y. Jared Lee Lewis, «IP Reputation Analysis of Public Databases and Machine Learning Techniques,» %1 International Conference on Computing, Networking and Communications, ICNC 2020. IEEE, 2020.

[89] «AbuseIPDB,» [Tiešsaiste]. Available: https://www.abuseipdb.com/. [Piekļūts 11 08 2021].

[90] F. a. M. J. P. Magalhaes, «Adopting machine learning to support the detection of malicious domain names,» 7th International Conference on Internet of Things: Systems, Management and Security, IOTSMS 2020, 2020.

[91] abuse.ch, [Tiešsaiste]. Available: https://urlhaus.abuse.ch/downloads/text/. [Piekļūts 11 08 2021].

[92] «*DGA* Collection,» [Tiešsaiste]. Available: https://github.com/pchaigno/dga-collection. [Piekļūts 11 08 2021].

[93] «Firefox DNS-over-HTTPS,» Firefox, [Tiešsaiste]. Available: https://support.mozilla.org/en-US/kb/firefox-dns-over-https. [Piekļūts 03 05 2021].

[94] M. Vale, «Google Public DNS now supports DNS-over-TLS,» Google, 2019. [Tiešsaiste]. Available: https://security.googleblog.com/2019/01/google-public-dns-now-supports-dns-over.html. [Piekļūts 03 05 2021].

[95] J. Y. Z. W. H. L. X. Sun, «HGDom: Heterogeneous Graph Convolutional Networks for Malicious Domain Detection,» %1 NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium, 2020.

[96] Y. S. Y. H. Y. L. J. L. Wei Wang, «BotMark: Automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors,» Information Sciences, sēj. Volume 511, pp. 284-296, 2020.

[97] H. B. S. Nõmm, «Unsupervised Anomaly Based Botnet Detection in IoT Networks,» %1 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018.

[98] Z. T. R. Z. L. L. J. Liu, «A Distance-Based Method for Building an Encrypted Malware Traffic Identification Framework,» %1 IEEE Access.

[99] «English Dictionary,» Cambridge, [Tiešsaiste]. Available: https://dictionary.cambridge.org/dictionary/english/. [Piekļūts 11 08 2021].

[100] «Vulnerabilities, Exploits, and Threats at a Glance,» Rapid7, [Tiešsaiste]. Available: https://www.rapid7.com/fundamentals/vulnerabilities-exploits-threats/. [Piekļūts 11 08 2021].

[101] «The OWASP Testing Project,» OWASP, [Tiešsaiste]. Available: https://owasp.org/www-project-web-security-testing-guide/latest/2-Introduction/README. [Piekļūts 11 08 2021].

[102] «CVE details,» MITRE Corporation, [Tiešsaiste]. Available: https://www.cvedetails.com/index.php. [Piekļūts 11 08 2021].

[103] «OpenVas by Greenbone,» Greenbone, [Tiešsaiste]. Available: https://openvas.org. [Piekļūts 11 08 2021].

[104] «The Nessus Family,» Tenable, 2021. [Tiešsaiste]. Available: https://www.tenable.com/products/nessus. [Piekļūts 05 05 2021].

[105] «One platform one agent one view,» Qualys, [Tiešsaiste]. Available: https://www.qualys.com. [Piekļūts 11 08 2021].

[106] Z. Chkirbene, S. Eltanbouly, M. Bashendy, N. Alnaimi un A. Erbad, «Hybrid Machine Learning for Network Anomaly Intrusion Detection,» 2020 IEEE

International Conference on Informatics, IoT, and Enabling Technologies, ICIoT 2020, pp. 163-170, 2020.

[107] C. I. f. Cybersecurity, «Publiski pieejams datu avots NSL-KDD,» 2009. [Tiešsaiste]. Available: https://www.unb.ca/cic/datasets/nsl.html. [Piekļūts 30 04 2021].

[108] R. M. Elbasiony, E. A. Sallam, T. E. Eltobely un M. M. Fahmy, «A hybrid network intrusion detection framework based on random forests and weighted k-means,» Ain Shams Engineering Journal, sēj. 4, pp. 753-762, 2013.

[109] «Kārtība, kādā tiek nodrošināta informācijas un komunikācijas tehnoloģiju sistēmu atbilstība minimālajām drošības prasībām,» likumi.lv, [Tiešsaiste]. Available: https://likumi.lv/ta/id/275671-kartiba-kada-tiek-nodrosinata-informacijas-un-komunikacijas-tehnologiju-sistemu-atbilstiba-minimalajam-drosibas-prasibam. [Piekļūts 12 08 2021].

[110] S. H. Mousavi, M. Khansari un R. Rahmani, «A fully scalable big data framework for Botnet detection based on network traffic analysis,» Information Sciences, nr. 512, pp. 629 - 640, 2020.

[111] R. Alguliyev un Y. Imamverdiyev, «Big Data: Big Promises for Information Security,» %1 2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT), 2014.

[112] «Apache Spark™ is a unified analytics engine for large-scale data processing.,» Apache, 2021. [Tiešsaiste]. Available: https://spark.apache.org/. [Piekļūts 06 05 2021].

[113] «Umbrella Popularity List,» Cisco, 09 2021. [Tiešsaiste]. Available: http://s3-us-west-1.amazonaws.com/umbrella-static/index.html. [Piekļūts 21 09 2021].

[114] API Explorer, «API Explorer,» Aruba, [Tiešsaiste]. Available: https://www.arubanetworks.com/techdocs/ClearPass/6.7/Guest/Content/AdministrationTasks1/API_Explorer.htm. [Piekļūts 12 08 2021].

[115] «clearpass-api-python,» Aruba, [Tiešsaiste]. Available: https://github.com/aruba/clearpass-api-python. [Piekļūts 12 08 2021].

[116] «Getting started with the REST API,» Hewlett Packard, [Tiešsaiste]. Available: https://developers.hp.com/hp-proactive-management/getting-started-rest-api. [Piekļūts 12 08 2021].

[117] «REST API Guide,» Juinper, [Tiešsaiste]. Available: https://www.juniper.net/documentation/us/en/software/junos/rest-api/index.html. [Piekļūts 12 08 2021].

[118] «PAN-OS REST API,» Palo Alto, [Tiešsaiste]. Available: https://docs.paloaltonetworks.com/pan-os/9-0/pan-os-panorama-api/get-started-with-the-pan-os-rest-api/pan-os-rest-api.html. [Piekļūts 12 08 2021].

[119] M. D. H. Garg, «Securing IoT Devices and SecurelyConnecting the Dots Using REST API and Middleware,» %1 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), 2019.

[120] S. Cass, «The Top Programming Languages 2023,» IEEE, [Tiešsaiste]. Available: https://spectrum.ieee.org/the-top-programming-languages-2023. [Piekļūts 04 01 2024].

[121] «Apache Kafka,» Apache, 2021. [Tiešsaiste]. Available: https://kafka.apache.org/. [Piekļūts 06 05 2021].

[122] «Suricata Open Source IDS / IPS / NSM engine,» The Open Information Security Foundation., [Tiešsaiste]. Available: https://suricata-ids.org/. [Piekļūts 20 08 2019].

[123] «Intelligent Management Software,» Hewlett Packard Enterprise, 2021. [Tiešsaiste]. Available: https://buy.hpe.com/us/en/software/intelligent-management-software/c/c001014. [Piekļūts 05 05 2021].

[124] «Downloading Nmap,» Nmap.org, 2021. [Tiešsaiste]. Available: https://nmap.org/download.html. [Piekļūts 05 05 20201].

[125] «Build on InfluxDB,» InfluxData, 2021. [Tiešsaiste]. Available: https://www.influxdata.com/. [Piekļūts 06 05 2021].

[126] «Your observability wherever you need it,» GrafanaLabs, 2021. [Tiešsaiste]. Available: https://grafana.com/. [Piekļūts 06 05 2021].

[127] «Secure Network Access Control for Modern IT,» Aruba, 2021. [Tiešsaiste]. Available: Secure Network Access Control for Modern IT. [Piekļūts 06 05 2021].

[128] «ICANN root zone,» ICANN, 2021. [Tiešsaiste]. Available: http://stats.research.icann.org/dns/tld_report/archive/index.html. [Piekļūts 07 05 2021].

[129] «Gartner Magic Quadrant,» Gartner, [Tiešsaiste]. Available: https://www.gartner.com/en/research/methodologies/magic-quadrants-research. [Piekļūts 04 06 2021].

[130] «Introducing 1.1.1.1 for Families,» Cloudfare, [Tiešsaiste]. Available: https://blog.cloudflare.com/introducing-1-1-1-1-for-families/. [Piekļūts 07 06 2021].

[131] «Norton ConnectSafe,» Norton, [Tiešsaiste]. Available: https://en.wikipedia.org/wiki/Norton_ConnectSafe. [Piekļūts 07 06 2021].

[132] K. P. Shung, «Accuracy, Precision, Recall or F1?,» 15 05 2015. [Tiešsaiste]. Available: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9. [Piekļūts 25 03 2021].

[133] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin un J. Aguilar, «Towards the Deployment of Machine Learning Solutions in Network Traffic Classification: A Systematic Survey,» 2019.

[134] «BlackBerry Protect,» BlackBerry, [Tiešsaiste]. Available: https://www.blackberry.com/us/en/products/unified-endpoint-security/blackberry-protect. [Piekļūts 10 06 2021].

[135] «McAfee Endpoint Security,» McAfee, [Tiešsaiste]. Available: https://www.mcafee.com/enterprise/en-us/products/endpoint-security.html. [Piekļūts 10 06 2021].

[136] «Prevent endpoint breaches,» Broadcom, [Tiešsaiste]. Available: https://www.broadcom.com/products/cyber-security/endpoint/end-user. [Piekļūts 10 06 2021].

[137] P. H. Z. Hu,L. Zhu,J. Heidemann,A. Mankin,D. Wessels, Specification for DNS over Transport Layer Security (TLS), IETF Tools, 2016, p. 18.

[138] «IPv4 - Packet Structure,» TutorialsPoint, [Tiešsaiste]. Available: https://www.tutorialspoint.com/ipv4/ipv4_packet_structure.htm. [Piekļūts 16 06 2021].

[139] «Suricata User Guide,» Suricata, [Tiešsaiste]. Available: https://suricata.readthedocs.io/en/suricata-6.0.2/. [Piekļūts 16 06 2021].

[140] E. S. I. K. I. S. A. Privalov, «Graph-based evaluation of probability of disclosing the network structure by targeted attacks,» %1 NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium, 2020.

PIELIKUMI

A.1 Izmantotie termini un saīsinājumi

A1. 1. tabulā apkopoti darbā izmantotie termini un to skaidrojums, A1.2. tabulā ietverto darbā izmantotie saīsinājumi un to atšifrējums.

A1.1. tabula

Darbā izmantotie termini

Termins/saīsinājums	**Skaidrojums**
Informācijas tehnoloģija	Zināšanu, metožu, paņēmienu un tehniskā aprīkojuma kopums, kas ar datoru un sakaru līdzekļu starpniecību nodrošina jebkuras informācijas iegūšanu, glabāšanu un izplatīšanu.
Informācijas sistēma	Iekārtu, procedūru, lietotojumprogrammu, tās komponentu un integrāciju kopums, kas paredzēts, ir izveidots, strādā un tiek uzturēts, lai vāktu, uzkrātu, apstrādātu, uzglabātu un izmantotu informāciju.
Ielaušanās noteikšanas sistēma	ierīce vai programmatūras kopums, kas uzrauga tīklu vai sistēmu attiecībā uz ļaunprātīgām darbībām vai drošības politikas pārkāpumiem, brīdinot atbildīgo personu.
Ielaušanās novēršanas sistēma	ierīce vai programmatūras kopums, kas uzrauga tīklu vai sistēmu attiecībā uz ļaunprātīgām darbībām vai drošības politikas pārkāpumiem, brīdinot atbildīgo personu un veicot aktīvas darbības, kā ierīces atslēgšana no tīkla.
vairākdatoru sistēma	Divu vai vairāku datoru saslēgums, kurā katrs dators izpilda kādu uzdevuma daļu, specifisku funkciju vai funkciju kombināciju.
Lielie dati	datu uzkrāšana milzīgā apjomā, kas sarežģī to apstrādi, izmantojot tradicionālos datu bāžu pārvaldības rīkus.
EDR sistēmas	EDR tiek izmantots, lai noteiktu, vai galapunkta ierīcē ir uzinstalēta ļaunprātīga programmatūra (piemēram APT), un atrastu veidus, kā reaģēt uz šāda veida draudiem (sniegtu atbildi). Bieži vien EDR risinājumi izmanto aģentus, kas instalēti galaiekārtā, lai savāktu datus no

	daudziem dažāda veida datu avotiem attiecībā uz šo galiekārtu un saglabātu tos centrālajā datu bāzē. Starp šiem datiem, bet ne tikai, ir dati no šādiem avotiem: • ARP • DNS • Tīkla dati • Sistēmas reģistrs • Sistēmas atmiņas stāvoklis • IP adreses • Uzstādītā aparatūra EDR tiek uzskatīts par papildinājumu tradicionālajiem aizsardzības līdzekļiem, piemēram, uz parakstiem balstītiem rīkiem vai SIEM. Tie visi nodrošina informācijas paneļus vai ziņojumus, un tiek veikta datu analīze. EDR risinājumi pašlaik atbalsta galvenokārt Windows OS, tikai sāk atbalstīt citas platformas, piemēram, Linux, Unix, iOS vai Android.
SIEM sistēmas	SIEM tiek izmantots, lai nodrošinātu vienotu centrālo vietu datu glabāšanai un analīzei no dažādiem žurnālu avotiem. Tas nekādā veidā neaprobežojas tikai ar galaiekārtām un nodrošina līdzekļus informācijas klasificēšanai, lai redzētu un analizētu visus datus reāllaikā un spētu atbilstoši rīkoties. Visa informācija ir sniegta viegli saprotamā veidā un nav saistīta ar konkrētu produktu. SIEM izmanto dažādus lietojuma gadījumus, kas nav atkarīgi tikai no viena veida sistēmas, bet gan no daudziem dažādiem žurnālu avotiem, piemēram, ugunsmūriem, serveriem, IPS, starpniekserveriem utt. SIEM atbalsta daudzas dažādas platformas un var izmantot datu korelācijai, žurnālu pārvaldībai un kriminālistikai. SIEM var tikt pielietots dažādās jomās, piemēram, IT operācijās, IT drošībā, finanšu vai medicīnisku datu apstrādei un citur

"Meduspods"	Programmatūras kopums, kura uzdevums ir pievilināt uzbrucējus, atstājot pieejamus dažādus portus, kā arī imitējot ievainojamu sistēmu atbildes
Precision - Precizitāte	Mašīnmācīšanās alogoritmu novērtēšanas mērs, bieži izmantot termins angļu valodā, kurš nosaka algoritma darbības precizitāti
Accuracy - Ticamība	Mašīnmācīšanās alogoritmu novērtēšanas mērs, bieži izmantot termins angļu valodā, kurš nosaka algoritma darbības precizitāti
Recall -Pārklājums	Mašīnmācīšanās alogoritmu novērtēšanas mērs, bieži izmantot termins angļu valodā, kurš nosaka algoritma darbības precizitāti – pārklājums
F1-Score – F1-mērs	Mašīnmācīšanās alogoritmu novērtēšanas mērs, bieži izmantot termins angļu valodā, kurš nosaka algoritma darbības precizitāti – F1-mērs

A1.2. tabula

Darbā izmantotie saīsinājumi un termini

Saīsinājums	**Atšifrējums**
ARP	Sakaru protokols, kuru izmanto, lai atklātu saites slāņa adresi, piemēram, MAC adresi, kas saistīta ar noteiktu interneta slāņa adresi, parasti IPv4 adresi
APIDS	Uz OSI lietojumslāni balstīta ielaušanās noteikšanas sistēma
APT	Slepena ļaunprātīga darbība, parasti programma, kura ir nacionāla valsts sponsorēta grupa, kuras izmantošanas rezultātā tiek iegūsta nesankcionēta piekļuve datortīklam un tā ilgstoši paliek neatklāta
Apache Kafka	Reāla laika ziņojuma datu apstrādes sistēma

API	Aplikāciju programmēšnas interfeiss, lai sadarbotos ar kādu produktu, piemēram, izmantojot API ir iespējams bloķēt inficētai ierīcei piekļuvi datortīklam.
BIG DATA	Lielie dati
C&C	Robottīkla komadcentri, kuri tiek izmantoti robotu tīkla darbības nodrošināšanai, komunicējot ar atsevišķie botiem (inficētām ierīcēm) un uzdodot šiem botiem uzdevumus
C&C	robotu tīklu komandas un kontroles centri, kuri tiek izmantoti, lai vadītu inficētās ierīces un dotu tam komandas
CDD	Spējā izstrādes pieeja (Capability driven developement)
CentOS	Uz atvērto kodu bāzēta serveru bāzēta operētājsistēma (https://www.centos.org/)
CISO	Informācijas drošības pārvaldnieks
Cron	ir uz laiku balstīts darba plānotājs Unix līdzīgās datoru operētājsistēmās
CVSS	Ievainojamības bīstamības pakāpe intervālā no 0-10, kur 0 - nav bīstama, bet 10 ļoti bīstama ievainojamība
DIDS	Izkliedēta ielaušanas noteikšanas sistēma
DGA	Domēnu ģenerēšanas algoritms
DHCP	Dinamiskais IP adreses piešķiršanas mehānisms
DMZ	Demilitarizētā zona - koncepts, kad ierīces šajā zonā tiek atdalītas ar ugunsmūri gan no interneta, gan no lietotājiem
DTC	Lēmumu koku klasifikators
ECN	Datu paketē informācija par sastrēgumiem definēts ar *RFC* 3168 (2001). ECN ļauj paziņot par tīkla pārslodzi, nenometot paketes
FTP	Failu pārsūtīšanas protokols. FTP ir veidots uz klienta-servera modeļa arhitektūras bāzes un izmanto atsevišķus vadības un datu savienojumus starp klientu un serveri

GDPR	Vispārējā datu aizsardzības regula
HADOOP	Atvērtā pirmkoda programmatūras kopums, kas var izmantot daudzus datorus tīklā risinot problēmas, kas saistītas ar lielajiem datiem
HDFS	HADOOP ir mērogojama un izplatītā failu sistēma, kas rakstīta Java HADOOP ietvaram
HIDS	Ielaušanās noteikšanas sistēma, kura tiek izvietota uz klienta darbstacijas un uzrauga komunikāciju starp klientu un internetu
HONEYPOT	Datora drošības mehānisms, kas izveidots, lai atklātu, novirzītu vai neitralizētu informācijas sistēmu nesankcionētas izmantošanas mēģinājumus
HTTP	Hiperteksta pārsūtīšanas protokols (HTTP) ir lietojumprogrammu slāņa protokols interneta protokola komplekta modelī
IDS	Ielaušanās noteikšanas sistēma
IPS	Ielaušanās novēršanas sistēma
IRC	(Internet Relay Chat) teksta tērzēšanas sistēma. Tā nodrošina tērzēšanu starp jebkuru dalībnieku skaitu tā sauktajos sarunu kanālos, kā arī diskusijas tikai starp diviem partneriem, piemēram, jautājumu un atbilžu dialogos. Jebkurš dalībnieks var atvērt jaunu sarunu kanālu, un viens datora lietotājs var piedalīties arī vairākos šādos vienlaikus kanālos.
IS	Informācijas sistēma
ISMS	IS drošības pārvaldības platforma
ISO27002	Informācijas drošības standarts
JSON	JSON jeb JavaScript Object Notation ir datu strukturēšanas veids, jeb veids kā var tikt strukturēti dati objektā. JSON formāts ļauj specifiskā struktūrā pārsūtīt datus no klienta uz serveri konkrētā formātā
JSON	JavaScript Object Notation ir atvērts standarta faila un datu apmaiņas formāts, kas izmanto cilvēkiem lasāmu tekstu, lai uzglabātu un pārsūtītu datu objektus, kas sastāv no atribūtu un vērtību pāriem, kā arī masīviem (vai citām sērijveidojamām vērtībām).

KIP	Konfidencialitāte, integritāte un pieejamība
KNN	K-tuvāko kaimiņu klasifikācijas metode (k-Nearest Neighbors)
MAC adrese	Unikāla, ražotāja piešķirta adrese tīkla interfeisam, piemēram: AA:BB:CC:00:11:22
MapReduce	MapReduce ir programmēšanas modelis un ar to saistīta īstenošana lielu datu kopu apstrādei un ģenerēšanai izmantojot paralēlu, izkliedētu algoritmu klasterī
MD5	Jaucējfunkcijas algoritms, kas izmanto 128 bitu jaucējvērtību
MITM	Man-in-the-middle uzbrukums, kur ļaunprātīgais lietotājs ir starp leģitīmo lietotāju un viņa lietoto servisu
MK442	Ministru kabineta 2015.gada 28.jūlija noteikumi Nr.442 "Kārtība, kādā tiek nodrošināta informācijas un komunikācijas tehnoloģiju sistēmu atbilstība minimālajām drošības prasībām"
MS Forms	Microsoft Office 365 funkcionalitāte, kas ļauj iegūt atgriezenisko skaiti no lietotājiem
Nessus	Ievainojamību skaneris
NETFLOW	Tīkla metadati
nfdump	Nfdump ir rīku komplekts, lai uzkrātu un apstrādātu netflow un sflow datus. Rīks ir spējīgs apstrādāt un uzkrāt netflow v1, v5/v7,v9,IPFIX un SFLOW.
NIDS	Ielaušanās noteikšanas sistēma, kura tiek izvietota tīklā un uzrauga komunikāciju starp klientu un internetu
NMAP	Portu skanēšanas rīks
NNC	Neironu tīklu klasifikators
M365	Office 365. Microsoft mākoņpakalpojuma produkts, kas iekļau e-pastu, failu apmaiņu, dokumentu izstrādi un citus.
PCI DSS	Maksājumu karšu industrijas drošības standarts

PCI DSS	Maksājumu karšu nozares datu drošības standarts ir informācijas drošības standarts organizācijām, kuras apstrādā kredītkartes datus
PIDS	Uz protokolu bāzēta ielaušanās noteikšanas sistēma
PYTHON	Augsta līmeņa programmēšanas valoda *Python*
INFLUX	Atvērtā koda laikrindu datubāze, kuru izstrādājusi kompānija InfluxData
OSI	Atvērto sistēmu sadarbības bāzes etalonmodelis (OSI modelis)
RFC	Lēmumu mežu klasifikators
SHA1	Jaucējfunkcijas algoritms, kas izmanto 160 bitu jaucējvērtību
SIEM	Sistēmu uzraudzības rīks
SMTP	Simple mail transfering protocol - protokols, e-pasta pārsūtīšanai starp serveriem
SOAR	Drošības incidentu automatizēta reaģēšanas sistēma
SOC	Sistēmu drošības operāciju centrs
SQL	Strukturēta pieprasījumu valoda darbam ar datubāzes ierakstiem
SSH	Secure socket shell konsoles vide, lai piekļūtu un administrētu informācijas sistēmu
SVC	Support Vector Machine klasifikators
syslog	Syslog ir ziņojumu reģistrēšanas standarts. Katrs ziņojums ir apzīmēts ar objekta kodu, norādot programmatūras tipu, kas ģenerē ziņojumu, un tam tiek piešķirta kritiskuma pakāpe
TCP	Viens no galvenajiem protokoliem Interneta protokolu saimē. Tas tika izstrādāts sākotnējā tīkla ieviešanas stadijā, kurā tas papildināja interneta protokolu (IP). Tāpēc visu komplektu parasti sauc par *TCP*/IP
TTL	Time to live - ir mehānisms, kas ierobežo datora vai tīkla datu paketes dzīves ilgumu, lai pakete mūžīgi neceļotu pa datortīklu
UDP	Izmantojot *UDP*, lietojumprogrammas var nosūtīt ziņojumus, kas šajā gadījumā tiek saukti par datagrammām, citiem resursdatoriem interneta

	protokola (IP) tīklā. *UDP* izmanto vienkāršu bezsaistes komunikācijas modeli ar minimālu protokola mehānismu. Atšķirībā to *TCP*, *UDP* nav datu paketes kontrolsummas pārbaudes, tāpēc to parasti izmanto multimēdiju datu pārraudei
URL	Interneta tīmekļa adrese (URL) ir atsauce uz internetā pieejamu resursu

A.2 Izmantotie skripti

1) *Python* skripts, kurš tiek izmantots *ISMS* platformā, lai identificētu rupja spēka paroles minēšanas mēģinājumus SSH servisā.

```
#!/usr/bin/python3


import subprocess as sub

import time,os

from datetime import datetime

from collections import Counter

papild = ""

start = time.time()

pasts = time.time() - 20

ipadrese = ""

ports = '22'

no = ["0.0.0.0"]

uz = []

all = []

def ierakstit(info):

 with open("/var/log/suricata/fast.log", 'a') as file:

   file.write(info + "\n")

p = sub.Popen(('tcpdump', '-i', 'p1p1', '-nn', 'tcp and dst port ', ports), stdout=sub.PIPE)

for row in iter(p.stdout.readline, b''):

 a = row.rstrip()   # process here

 try:

  src = a[a.find(" IP ")+4:a.find(" > ")]

  dst = a[a.find(" > ")+3:a.find(":",a.find(" > "))]

  s = src.split('.')

  d = dst.split('.')

  if len(s) > 3:

   src = s[0] + "." + s[1] + "." + s[2] + "." + s[3]

   sport = s[4]

  if len(d) > 3:
```

```
   dst = d[0] + "." + d[1] + "." + d[2] + "." + d[3]
   dport = d[4]
  laiks = a[:a.find(" ")]
 # ************************************************
  if (time.time() - start) > 10:
   xx = Counter(no).most_common()
   if (xx[0][1] > 6):
      for i, line in enumerate(all):
       if all[i][1] == xx[0][0]:
        ipadrese = all[i][2]
      now = datetime.now()
     konsole = str(xx[0][0])
     if konsole != "0.0.0.0":
       kom = 'echo "^' + konsole + '" > /dev/udp/???.???.???.???/514'
       os.system(kom)
   del no[:]
   del all[:]
   del uz [:]
   start = time.time()
  if len(dst) > 0:
   if dst not in uz:
    uz.append(dst)
    no.append(src)
    all.append([laiks,src,dst])
 except: pass
```

2) *Python* skripts, kurš tiek izmantots *ISMS* platformā, lai uzkrātu DNS pieprasījumus turpmākai to apstrādei.

```
#!/usr/local/bin/python3
import subprocess as sub
import time,json
lastsrc = ""
lastdst = ""
```

```
lastdns = ""

def ierakstit(info):

 with open("/root/Python/visi_dns.log", 'a') as file:

   file.write(info + "\n")

from kafka import KafkaProducer

from kafka.errors import KafkaError

# To produce messages

producer = KafkaProducer(bootstrap_servers=['??.??.??.??:9092'])

p = sub.Popen(('tcpdump', '-i', 'p1p1', '-n', 'udp', 'port 53'), stdout=sub.PIPE)

for row in iter(p.stdout.readline, b''):

    a = row.rstrip()   # process here

    a = str(a)

    if (' A? ' in a) and ('AAAA?' not in a):

     src = a[a.find(" IP ")+4:a.find(" > ")]

     dst = a[a.find(" > ")+3:a.find(":",a.find(" > "))]

     s = src.split('.')

     d = dst.split('.')

     src = s[0] + "." + s[1] + "." + s[2] + "." + s[3]

     sport = s[4]

     dst = d[0] + "." + d[1] + "." + d[2] + "." + d[3]

     dport = d[4]

     time = a[+2:a.find(" ")]

     req =  a[a.find(" A? ")+4:a.find("(",a.find(" A? "))-2]

     if (lastsrc != src) and (lastdst !=dst) and (lastdns != req):

      #  kopa = (time + " " + src + ">" + dst + " " + req)

       if ("," not in req) and ("^" not in req) and ('"' not in req):

        kopa = '{"time":"' + time + '","src":"' + src + '","dst":"' + dst + '","dns":"' + req
+ '"}'

        try:

         future = producer.send("dns", kopa.encode('utf-8'))

         record_metadata = future.get(timeout=10)

        except KafkaError:

         pass
```

```
        ierakstit(time + " " + src + ">" + dst + " " + req)
        lastsrc = src
        lastdst = dst
        lastdns = req
```

3) *Python* skripts, kurš tiek izmantots *ISMS* platformā, lai identificētu RDP skanēšanu.

```
#!/usr/bin/python
import subprocess as sub
import time,ipaddr,os
from datetime import datetime
from collections import Counter
papild = ""
start = time.time()
pasts = time.time() - 20
ipadrese = ""
ports = '3389'
no = ["0.0.0.0"]
uz = []
all = []
def ierakstit(info):
 with open("/var/log/suricata/fast.log", 'a') as file:
   file.write(info + "\n")
p = sub.Popen(('tcpdump', '-i', 'p1p1', '-nn', 'tcp and dst port ', ports), stdout=sub.PIPE)
for row in iter(p.stdout.readline, b''):
 a = row.rstrip()   # process here
 try:
  src = a[a.find(" IP ")+4:a.find(" > ")]
  dst = a[a.find(" > ")+3:a.find(":",a.find(" > "))]
  s = src.split('.')
  d = dst.split('.')
  if len(s) > 3:
```

```
   src = s[0] + "." + s[1] + "." + s[2] + "." + s[3]
   sport = s[4]
  if len(d) > 3:
   dst = d[0] + "." + d[1] + "." + d[2] + "." + d[3]
   dport = d[4]
  laiks = a[:a.find(" ")]
 # ************************************************
  if (time.time() - start) > 3:
   xx = Counter(no).most_common()

   if (xx[0][1] > 4):
    a = ipaddr.IPAddress(xx[0][0])
    if (ipaddr.IPNetwork('192.168.0.0/16').Contains(a))\
       or (ipaddr.IPNetwork('10.0.0.0/8').Contains(a)) or
(ipaddr.IPNetwork('172.16.0.0/12').Contains(a)):
     v = 1
    else:
      for i, line in enumerate(all):
       if all[i][1] == xx[0][0]:
        ipadrese = all[i][2]
      now = datetime.now()
      konsole = str(xx[0][0])
      if konsole != "0.0.0.0":
       kom = 'echo "^' + konsole + '" > /dev/udp/??.??.??.??/514'
       os.system(kom)
   del no[:]
   del all[:]
   del uz [:]
   start = time.time()
  if len(dst) > 0:
   if dst not in uz:
    uz.append(dst)
    no.append(src)
```

```
    all.append([laiks,src,dst])
 except: pass
```

4) *Python* skripts, kurš tiek izmantots *ISMS* platformā, lai veidotu "meduspodu".

```
import time
import socket


def getInput():
        motd = input('MOTD: ')
        host = input('IP Address: ')
        while True:
                try:
                        port = int(input('Port: '))
                except TypeError:
                        print ('Error: Invalid port number.')
                        continue
                else:
                        if (port < 1) or (port > 65535):
                                print ('Error: Invalid port number.')
                                continue
                        else:
                                return (host, port, motd)


def writeLog(client, data=''):
        separator = '='*50
        fopen = open('/root/Documents/Honeypot/honey.log', 'a')
        fopen.write('Port 3389 Time: %s\nIP: %s\nPort: %d\nData: %s\n%s\n\n\n'%(time.ctime(),
client[0], client[1], data, separator))
        fopen.close()


def main(host, port, motd):
```

```
        print ('Starting honeypot!')
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.bind((host, port))
        s.listen(100)
        while True:
                (insock, address) = s.accept()
                print ("Connected 3389 " + str(time.ctime()) + "   " + str(address))
                try:
                        insock.send('%s'%(motd))
                        data = insock.recv(1024)
                        insock.close()
                except socket.error as e:
                        writeLog(address)
                else:
                        writeLog(address, data)


if __name__=='__main__':
        try:
                main("10.100.XX.XX",3389,"")
        except KeyboardInterrupt:
                print ('Bye!')
                exit(0)
        except BaseException as e:
                print('Error: {0}'.format(str(e)))
exit(1)
```

5) *Python* Skripts, lai identificētu vai lietotāja parole ir nozagta

```
#!/usr/local/bin/python3.8

# -*- coding: utf-8 -*-

from collections import Counter

import socket,subprocess

import time,os,sys

from datetime import datetime,timedelta

import json,ipaddr

from influxdb import InfluxDBClient

user = '*******'

password = '********'

host = '********'

port = 8086

dbname = 'A*****'

client = InfluxDBClient(host, port, user, password, dbname)

tagad = datetime.now()

sakuma_datums = tagad - timedelta(minutes=1440,hours=3)



microsoft = ['52.96.0.0/12','52.112.0.0/14','40.112.0.0/13', '40.80.0.0/12',\

                 '40.124.0.0/16', '40.96.0.0/12', '40.120.0.0/14', '40.74.0.0/15',
'40.76.0.0/14', '40.125.0.0/17',\

                '35.224.0.0/12', '35.208.0.0/12', '35.240.0.0/13',\

                '3.128.0.0/9','209.85.128.0/17']


specifiski_useragenti = ['ZOOM_CALENDARING (ExchangeServicesClient/0.0.0.0)','Go-http-
client/1.1']



def izpildit_komandu(komanda):

 status, output = subprocess.getstatusoutput(komanda)

 if (status == 0):

  return output



def iegut_html(dati):


 full_code = '<html><div><br>Security system\

          identified that you logged on into M365 from different countries in 24 hours time.
If the person who logged on from these \
```

```
         countries is not you, please change your M365 password
immediately!<br><br></div><head><style> th, td { padding: 1px; text-align: left;} \

        </style> </head> <body> <table
style="width:300px"><tr><td>Time</td><td>Date</td><td></td><td>IP
Address</td><td>Country</td></tr><tr>'



 kopejais_kods = '<html><div><br>Dro&#353;&#299;bas sist&#275;ma ir identific&#275;jusi
autentifik&#257;ciju \

         M365 no da&#382;&#257;d&#257;m valst&#299;m 24 stundu laik&#257;.\

         <div> <div>Ja persona, kura piesl&#275;dz&#257;s M365 no &#353;&#299;m valst&#299;m
nebij&#257;t J&#363;s, \

         l&#363;dzam nekav&#275;joties nomain&#299;t M365 paroli!<br><br></div><head><style>
th, td { padding: 1px; text-align: left;}\

        </style> </head> <body> <table
style="width:300px"><tr><td>Time</td><td>Date</td><td></td><td>IP
Address</td><td>Country</td></tr><tr>'



 for x in dati:

   kopejais_kods = kopejais_kods + '<td>' +  x[0].strftime("%H:%M:%S") + '</td><td>' +
x[0].strftime("%d.%m") + '</td><td></td><td>' + x[1] + '</td><td>' + x[2] + '</td></tr><tr>'

   full_code = full_code +  '<td>' +  x[0].strftime("%H:%M:%S") + '</td><td>' +
x[0].strftime("%d.%m") + '</td><td></td><td>' + x[1] + '</td><td>' + x[2] + '</td></tr><tr>'




 kopejais_kods = kopejais_kods + '</tr></table></body></html>'

 full_code = full_code + '</tr></table></body></html>'

 return(kopejais_kods,full_code)




def ieliekam_datus_sql(Vards,CommentLV,CommentEN):

   import pypyodbc

   conn =
pypyodbc.connect('Driver=FreeTDS;Server=*****;port=1433;uid=*******;pwd=********;database=****
***')

   selekts = """SELECT TOP 1 * FROM [Radius].dbo.Notification  WHERE Username = '%s' """ %
(Vards)

   selekts_ieliekam = """INSERT INTO
[******].[dbo].[Notification]([UserName],[CommentLV],[CommentEN]) VALUES ('%s','%s','%s')
""" % (Vards,CommentLV,CommentEN)

   selekts_update = """UPDATE [*****].[dbo].[Notification] SET CommentLV = '%s', CommentEN =
'%s' WHERE UserName = '%s' """ % (CommentLV,CommentEN,Vards)

   if conn.cursor().execute(selekts).fetchall():

    cur = conn.cursor()

    conn.cursor().execute(selekts_update)

    conn.commit()

   # print conn.cursor().execute(selekts).fetchall()
```

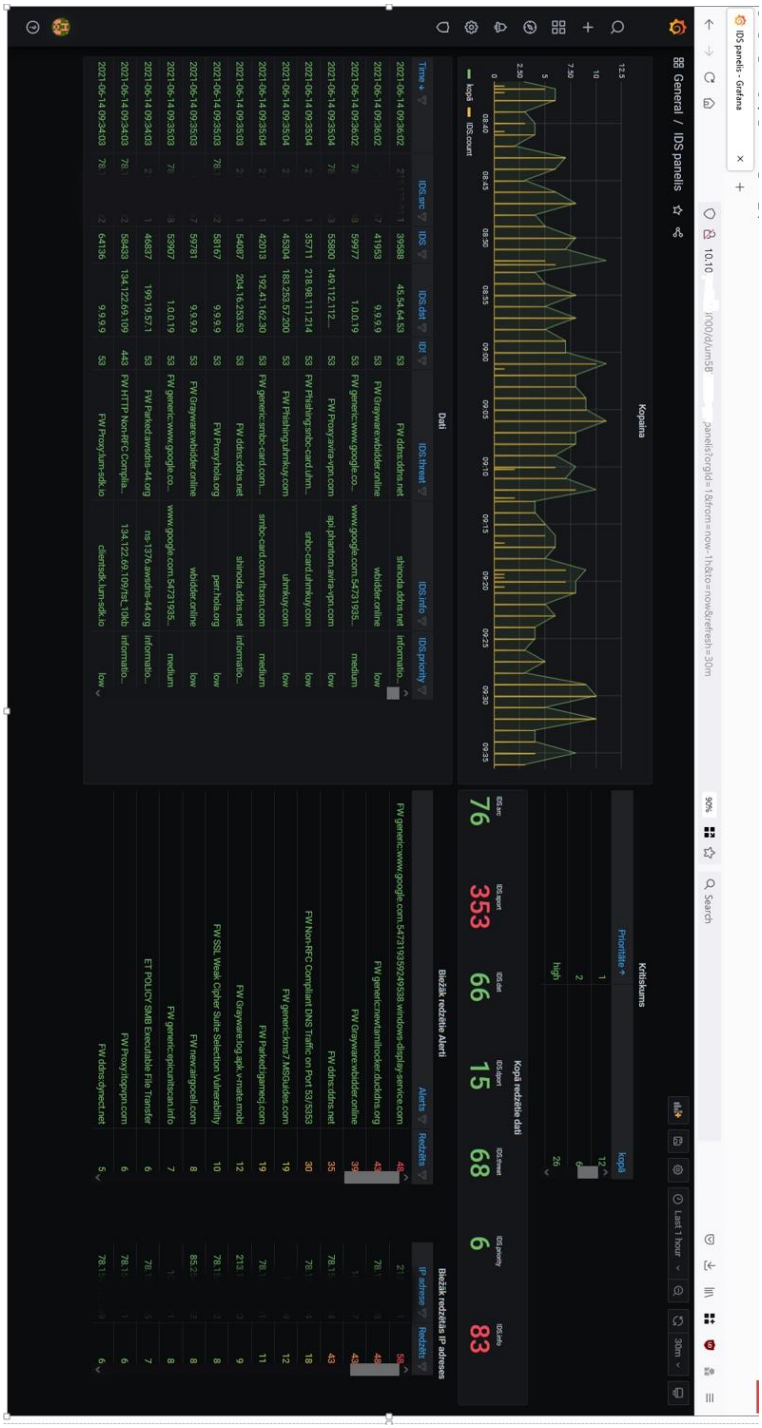
```
  else:
#  print "Nav"
   cur = conn.cursor()
   conn.cursor().execute(selekts_ieliekam)
   conn.commit()
  conn.close()
def novacam_punktus(dati):
  i = dati.split(".")
  return(i[len(i)-2]+'.'+i[len(i)-1])


def izpildit_komandu(komanda):
 status, output = subprocess.getstatusoutput(komanda)
 if (status == 0):
  return output
def mail_send_with_details(relay, sender, subject, to, text, xmailer=None, followup_to=None,
dry=True):
    import smtplib
    from email.mime.text import MIMEText
    import email.utils
    msg = MIMEText(text, _charset='UTF-8')
    msg['Subject'] = subject
    msg['Message-ID'] = email.utils.make_msgid()
    msg['Date'] = email.utils.formatdate(localtime=1)
    msg['From'] = sender
    msg['To'] = to
    if followup_to:
        msg['Mail-Followup-To'] = followup_to
    if xmailer:
        msg.add_header('X-Mailer', xmailer)
    msg.add_header('Precedence', 'bulk')
    s = smtplib.SMTP(relay)
    s.sendmail(msg['From'], {msg['To'], sender }, msg.as_string())
    s.quit()
a = client.query("select time_generated,user_name,ip,user_agent,country,operation,status from
AUTH WHERE time >= '" + str(sakuma_datums) + "' and  time <= now()")
x = (list(a.get_points()))


dati = {}
for i in x:
  try:
```

```
   dati[i['user_name']] = dati[i['user_name']] + "^" + i['time_generated'] + "`" + i['ip'] +
"`" + i['user_agent'] + "`" + i['status'] + "`" + i['country'] + "`" + i['operation']

  except:

   dati[i['user_name']] = i['time_generated'] + "`" + i['ip'] + "`" + i['user_agent'] + "`" +
i['status'] + "`" + i['country'] + "`" + i['operation']

final = []

for i,k in dati.items():

 if 'Not Available' not in str(i):

   x = k.split("^")

   cnt = 0

   iepr = ""

   valstis = ""

   agent = ""

   ipadr = ""

   timestamp = ""

   status = ""

   operation = ""

   ms_ip = ""

   known_user_agent = ""

   for m in x:

    y = m.split("`")

    if iepr != y[1]:

     #Tikai izfiltetie gadijumii

     ir_ms = 0

     mmm = ipaddr.IPAddress(y[1])

     for ms in microsoft:

       if ipaddr.IPNetwork(ms).Contains(mmm):

         ir_ms = 1

     ir_uag = 0

     for uag in specifiski_useragenti:

       if uag in y[2]:

         ir_uag = 1

     if y[3] == "Success" and "UserLoggedIn" in y[5]:

      if y[4] != "Unregistered":

       cnt = cnt + 1

       a  = datetime.strptime(y[0], "%d.%m.%Y %H:%M:%S")

       y[0] = a + timedelta(hours=2)

       y[0] = y[0].strftime("%d.%m.%Y %H:%M:%S")

       timestamp = timestamp + "`" + y[0]

       operation = operation + "`" + y[5]
```

```
        valstis = valstis + "`" + y[4]
        status = status + "`" + y[3]
        agent = agent + "`" + y[2]
        ipadr = ipadr + "`" + y[1]
        ms_ip = ms_ip + "`" + str(ir_ms)
        known_user_agent = known_user_agent + "`" + str(ir_uag)
        iepr = y[1]
    timestamp = timestamp[1:]
    operation = operation[1:]
    valstis = valstis[1:]
    agent = agent[1:]
    ipadr = ipadr[1:]
    status = status[1:]
    ms_ip = ms_ip[1:]
    known_user_agent = known_user_agent[1:]
    final.append([i,cnt,timestamp,valstis,ipadr,agent,status,operation,ms_ip,known_user_agent])
for i in final:
 try:
  # Cik dazadas IP adreses interese
  if 'rtu.lv' in i[0]:
        x1 = i[2].split("`")
        x2 = i[3].split("`")
        x3 = i[4].split("`")
        x4 = i[5].split("`")
        x5 = i[6].split("`")
        x6 = i[7].split("`")
        x7 = i[8].split("`")
        x8 = i[9].split("`")
        valstis = len(Counter(x2))
        unikalie = len(Counter(x4))
        u_ip_adreses = len(Counter(x3))
        #Cik unikali user agenti
        if unikalie > 1:
         #Cik unikalas valstis
         if valstis > 1:
          #Cik unikalas IP
          if u_ip_adreses > 1:
           uag = []
           ip_adreses = []
           valstis = []
```

```
          kop_rez_dic = {}
          for n in range(0,i[1]):
           if x7[n] == '0':
             if x8[n] == '0':
               kop_rez_dic[x2[n]] = x1[n] + "`" + x3[n] + "`" + x4[n] + "`" + x5[n] + "`" +
x6[n]
               uag.append(x4[n])
               ip_adreses.append(x3[n])
               valstis.append(x2[n])
          valstis = len(Counter(valstis))
          unikalie = len(Counter(uag))
          u_ip_adreses = len(Counter(ip_adreses))
          if int(valstis) > 1:
           kopejais_rez = []
           for k,v in kop_rez_dic.items():
            x = v.split("`")
            datetime_object = datetime.strptime(x[0], '%d.%m.%Y %H:%M:%S')
            kopejais_rez.append([datetime_object,x[1],k])
           username = i[0][:i[0].find("@")]
           rezultats =
izpildit_komandu("/root/Documents/Autentifikacijas_logi/datubaze_users.py " + str(i[0]) + "
0")
           if rezultats == '1' or rezultats == '2':
             kopejais_kods,full_code = (iegut_html(kopejais_rez))
             ieliekam_datus_sql(username,kopejais_kods,full_code)
             with open ('/root/Documents/Autentifikacijas_logi/auditacijas_pieraksti.log','a')
as f:
               f.write(str(datetime.now()) + " " + str(i[0]) +  " Valstis:" + str(valstis) + "
UAGent:" + str(unikalie) + " IP:" + str(u_ip_adreses) + " " + str(kop_rez_dic) + "\n")
             f.close()
             if int(valstis) > 3:
              mail_send_with_details('??.??.??.??','kafka@rtu.lv','Possible password
steal','vladislavs.minkevics@rtu.lv',str(i[0]) + "\n" + str(kop_rez_dic))
 except Exception as e:
  print(e)
 except:
   pass
```

6) ISMS platformas lietotāja saskarnes grafiskā komponente.



Vladislavs Minkevičs dzimis 1978. gadā Daugavpilī. Rīgas Tehniskajā universitātē ieguvis bakalaura un maģistra grādus informācijas tehnoloģijā, maģistra grādu ieguvis 2003. gadā. Patlaban strādā Centrālajā finanšu un līgumu aģentūrā Informācijas drošības vadītāja amatā. Zinātniskās intereses saistītas ar kiberdrošību, drošības operāciju centru automatizāciju un mākslīgā intelekta lietošanu kiberdrošībā.