TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Markus Kuuse 206049IAAB

SEIRELAHENDUSTE STANDARDISEERIMINE JA ÜHTSUSTAMINE AS EESTI POST NÄITEL

Bakalaureusetöö

Juhendaja: Edmund Laugasson
MSc

Kaasjuhendaja: Martin Rajur
MSc

Tallinn 2024

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Markus Kuuse

04.01.2024

Annotatsioon

Antud lõputöö keskendub IT seire standardisatsiooni uurimisele ning seiresüsteemide analüüsimisele. Töö eesmärk on standardiseerida ja ühtlustada IT infrastruktuuri seiret ettevõtte AS Eesti Post näitel. See hõlmab kasutusel olevate seiretööriistade analüüsimist, et leida nendes võimalikke kitsaskohti, ja seire standardisatsiooni väljatöötamist.

Lõputöös tutvustatakse ettevõtet AS Eesti Post, mainitakse ära mis seadmeid ning teenuseid IT peab jälgima. Samuti antakse ülevaade üleüldise seire taustast, sealhulgas erinevatest seire ning andme tüüpidest, andmeedastusmeetoditest ja süsteemi jälgimiseks mõeldud metoodikatest.

Töös analüüsitakse AS Eesti Postis kasutusel olevaid seiresüsteeme, rõhuasetusega nende haldamisel, konfiguratsioonil, automatiseerimise võimalustel ning dokumentatsiooni ja toe kvaliteedil. Analüüsi alusel hinnatakse seiresüsteeme vastavalt näidisettevõtte IT-alasetele vajadustele, selle tulemil on võimalik hinnata milliseid seiretööriistu peaks ettevõttes asendama.

Seirestandardisatsiooni juhise väljatöötamine on töö viimane etapp, kus autor annab soovitusi ja näpunäiteid standardisatsiooni rakendamiseks ettevõttes. Juhise koostamisel on silmas peetud, et see oleks võimalikult universaalne ning rakendatav olenemata kasutusel olevast seiresüsteemist.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 48 leheküljel, 9 peatükki, 3 joonist, 2 tabelit.

Abstract

Standardization and Unification of Monitoring Solutions Based on the Example of AS Eesti Post

The following bachelors thesis focuses on studying IT monitoring standardization and analysis of monitoring systems. The aim of this thesis is to standardize and unify the IT infrastructure monitoring based on the example of AS Eesti Post. This includes an analysis of monitoring tools in use within the company, to identify potential bottlenecks and the development of a monitoring standard.

The thesis introduces AS Eesti Post, outlining the equipment and services that IT monitors. The text provides an overview of the background of the monitoring process, including the various types of monitoring and data, data transmission methods, and monitoring methodologies for the system.

The analysis focuses on the management, configuration, automation options, and quality of documentation and support of the monitoring systems used in the company. The assessment will determine which monitoring tools are best suited to the IT needs of the example company.

The final stage of this work is the development of a guide for standardization monitoring. The author provides recommendations and examples for implementing standardization in enterprises. The guide aims to be universal and applicable regardless of the system used for monitoring.

The thesis is written in Estonian and is 48 pages long, including 9 chapters, 3 figures and 2 tables.

Lühendite ja mõistete sõnastik

Agent	Rakendus, mis kogub kohalikult süsteemilt andmeid ja saadab need seiresüsteemile
Apache veebiserver	Rakendus, mis võimaldab serveris olevaid faile kuvada veebilehitsejas
API	*Application Programming Interface*, liides, mis võimaldab ühendada erinevaid rakendusi
DNS	*Domain Name System*, hajusandmebaas, mis loob sideme domeeninimede ja IP aadresside vahel
Exporter	Rakendus, mis kogub kohalikult süsteemilt andmeid ja teeb need seiresüsteemile kättesaadavaks
Flag	Rakenduse käivitamisel juurde lisatud konfiguratsiooni atribuut
HTTP	*HyperText Transfer Protocol*, protokoll andmete edastamiseks veebiserveri ja kliendi vahel
Hüsterees	Varasem süsteemi seisund, millel võib olla mõju praegusele või tulevasele seisundile
IaC	*Infrastructure as Code*, protsess, millega on võimalik hallata süsteeme masin-loetavate failide abil
IPMI	*Intelligent Platform Management Interface*, arvutisüsteemide haldamise liides
Kubernetes	Tarkvara konteinerrakenduste orkestreerimiseks
MSSQL	Microsofti poolt arendatud relatsiooniline andmebaas
OID	*Object identifier*, unikaalne identifikaator andmeobjektil
SNMP	*Simple Network Management Protocol*, protokoll suhtlemiseks võrgus olevate seadmetega
SSH	*Secure Socket Shell*, protokoll, mis võimaldab süsteemile turvaliselt ligi pääseda
Syslog	Protokoll logide saatmiseks

Sisukord

Joonistе loetelu

Tabelite loetelu

1. Sissejuhatus

Tänapäeva kiiresti arenevas ärikeskonnas seisavad organisatsioonid oma IT-taristu tõhusal haldamisel silmitsi mitmete väljakutsetega. Süsteemide suurenev keerukus, kasvavad andmemahud ja kliendi ootus saada ööpäevaringset teenust on tekitanud selge vajaduse tõhusate ja töökindlate seirelahenduste järele.

Efektiivse seire puudumine võib kaasa tuua mitmeid tõsiseid tagajärgi ettevõtete süsteemides. Võimalikud seisakud, tervikluse või konfidentsiaalsuse kaod ning finantsiline kahju on vaid mõned näited probleemidest, mis võivad tekkida ilma süsteemide pideva ja tõhusa seiramiseta[1].

Eesoleva lõputöö eesmärk on uurida ettevõtte IT-taristu seire standardiseerimise ja ühtsustamise võimalusi ASi Eesti Post näitel. See hõlmab kasutusel olevate seiretööriistade analüüsimist, et leida nendes võimalikke kitsaskohti, ja seire standardisatsiooni väljatöötamist.

2. Taust

Selles peatükis tutvustatakse ettevõtet, mille näitel lõputöö valmib. Autor annab ülevaate ettevõtte seirevajadustest, sealhulgas sellest, milliseid seadmeid ja teenuseid nad aktiivselt jälgivad. Samuti käsitletakse üldise seire tausta, erinevaid seireandmete tüüpe, andmeedastusmeetodeid ning metoodikaid süsteemide analüüsimiseks.

2.1 Ettevõttest

AS Eesti Post, paremini tuntud turundusnime all Omniva, on rahvusvaheline logistikaettevõte. Omniva pakub nii ettevõtetele kui ka eraklientidele posti-, logistika- ja e-kaubanduse lahendusi[2]. Ettevõtte põhiline turg on Baltimaad, kuid tegutsetakse ka mujal maailmas. IT-süsteemide ühtlase ja tõrgeteta töö tagamiseks on Omnivas juurutatud robustne seirelahendus. Lahendus hõlmab tervet IT-taristut, sealhulgas rakenduste jälgimist. Põhilised jälgitavad seadmed on:

- Pakiautomaadid.
- Võrguseadmed (ruuterid, võrgulülitid, tulemüürid).
- Serverid (GNU/Linux, MS Windows).
- Andmemassiivid.
- Pakisorteerimisliin.

Seiratavate rakenduste alla kuuluvad:

- Andmebaasid.
- Kubernetese instantsid.
- Veebilehed.
- Siserakendused.

Süsteemide jälgimine võimaldab Omnival tuvastada tõrkeid enne, kui need hakkavad kriitilisi äriprotsesse häirima. Lisaks aitab seire käigus kogutud statistika otsuseid langetada, näiteks kui palju peaks süsteemile ressursse juurde andma.

2.2 Seire

Infotehnoloogias tähendab seire pidevat andmete kogumist, et tagada süsteemide ja rakenduste ootuspärane toimimine[3]. Seire käigus kogutud andmetele on võimalik süsteeme ja äriprotsesse põhjalikult analüüsida, teha järeldusi, planeerida infotehnoloogilise taristu skaleerimist ja reageerida kiiresti kriitilistele sündmustele.

2.2.1 Tüübid

Seire jaguneb kaheks põhitüübiks. Need on proaktiivne ja reaktiivne.

Proaktiivne

Proaktiivne seire on üks fundamentaalne osa talitluspidevuse planeerimisest[4]. Selle eesmärk on andmete kogumine selleks, et vältida ootamatuid tõrkeid või lahendada probleemid enne, kui need kahjustavad süsteemi käideldavust[5].

Reaktiivne

Reaktiivne seire keskendub probleemide tuvastamisele ja nende lahendamisele pärast nende tekkimist[6]. See tähendab, et reaktiivse seire käigus oodatakse probleemide ilmnemist ning seejärel reageeritakse vastavalt probleemi kriitilisusele. Reaktiivne seire koosneb tavaliselt kolmest etapist:

1. Probleemi tuvastamine: seiresüsteemis konfigureeritud alarmi reegel ületab lävendit.
2. Alarmi esitamine: seiresüsteem kuvab teadet reegli lävendi ületamisest.
3. Probleemi lahendamine: pärast alarmi märkamist lahendab süsteemiadministraator alarmi põhjuse.

2.2.2 Andmed

Seiretööriistad võivad koguda mitut tüüpi andmeid. Need andmed on tüüpiliselt kahel kujul:

- Logid.
- Mõõdikud.

Logid

Logi on (enamasti teksti kujul) teave, mida tarkvara või süsteem loob. Logid annavad parema arusaama konkreetsest probleemist või käitumisest süsteemis. Nende üks peamisi kasutusjuhte on tõrkeotsing[7].

Logi kirjete haldamise *de-facto* standard on *syslog*, mis pakub raamistikku logikirjete loomiseks, salvestamiseks ja edastamiseks[8, 9]. *Syslog*'i poolt loodud logidel on erinevad tasemed. Tasemeid on kokku seitse[10]:

- *Emergency*: süsteem pole kättesaadav.
- *Alert*: olukord vajab kohest sekkumist.
- *Critical*: kriitiline olukord.
- *Error*: veateade.
- *Warning*: hoiatav olukord.
- *Notice*: tavaline, kuid oluline olukord.
- *Informational*: teabe kirje.
- *Debug*: silumistaseme sõnum.

Logide jälgimise süsteemid koguvad erinevatest allikatest andmeid (näiteks *syslog*'i server), tuvastavad nendes sarnased elemendid (ajatempel, märksõnad jms), indekseerivad ja rühmitavad need. Tuntumad seiresüsteemid, mille põhifunktsioon on logide analüüsimine, on Elastic Stack, LogSplunk ja ArcSight Logger[11].

Mõõdikud

Mõõdikud on ressursside mõõtmise käigus saadud arvväärtused, mis annavad ülevaate süsteemi parameetritest nagu protsessori koormus, mälukasutus jms[12]. Mõõdikud pakuvad kolme põhifunktsiooni[13]:

1. Kontroll: mõõdikud võimaldavad kontrollida süsteemi, teenuse toimivust.
2. Suhtlus: mõõdikute abil saab edastada kergesti arusaadavat teavet teenuse toimivusest välistele osapooltele (tootetiimid, juhid).
3. Parendus: mõõdikud aitavad tuvastada lünkasid teenuse kvaliteedis, misjärel saab neid parendada.

Tuntumad seiretööriistad, mis kasutavad mõõdikupõhist lähenemist, on Prometheus, Zabbix ja Nagios.

2.2.3 Andmeedastusmeetodid

Selleks, et seiresüsteemid saaksid andmed jälgitavatelt seadmetelt kätte, on olemas kaks põhilist andmeedastusmeetodit: *push* ja *pull*[14].

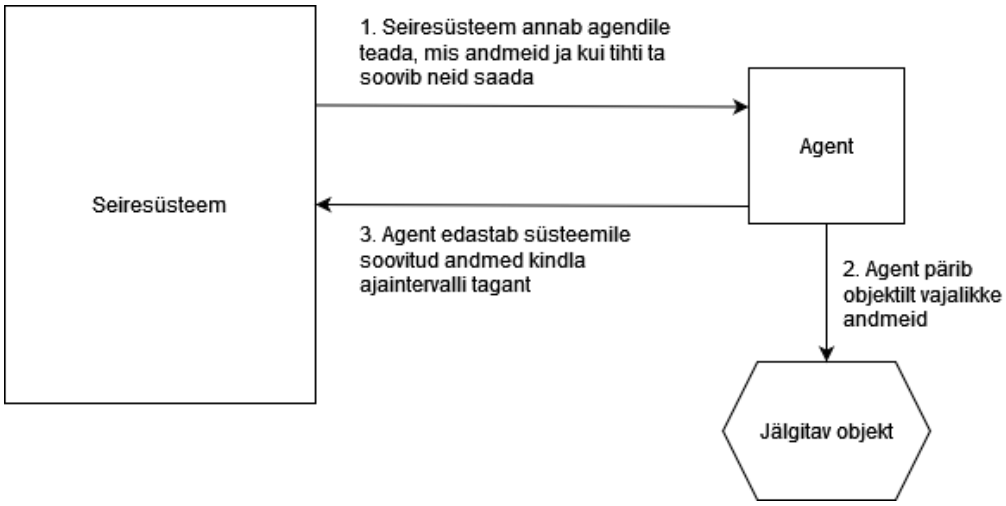
Push-meetod

Push-meetodi kasutamiseks peab jälgitavates süsteemides olema paigaldatud agent, mis saadab perioodiliselt andmeid tsentraalsele serverile[15], hajusa seirelahenduse[1] korral sõlmedele (*nodes*)[16]. Joonis 1 näitab andmeedastusvoogu täpsemalt.

Push-meetod vähendab võrgu liiklust, kuna seiresüsteem peab agendiga ühendust võtma ühe korra ja tellima soovitud *OID* kohta teate. Peale *OID*-teate tellimist, hakkab agent seiresüsteemile andmeid edastama[14]. *Push*-meetodil on kaks edastusviisi[16]:

- Perioodiline edastamine: andmeid saadetakse perioodiliselt (näiteks iga minuti tagant).
- Sündmusepõhine edastamine: andmeid saadetakse, kui ressursi väärtus muutub.

Push-meetodi kasutamise eeliseks on võime reageerida kiiresti andmete muutumise korral ja täita kõrgeid (andmete) tervikluse nõudeid – kuna agent on teadlik igast ressursi väärtuse muutusest, saab ta täpselt määrata, milliseid andmeid ja millal seiresüsteemile edastada[17].



Joonis 1. *Andmevoo liiklus push-meetodiga*

Pull-meetod

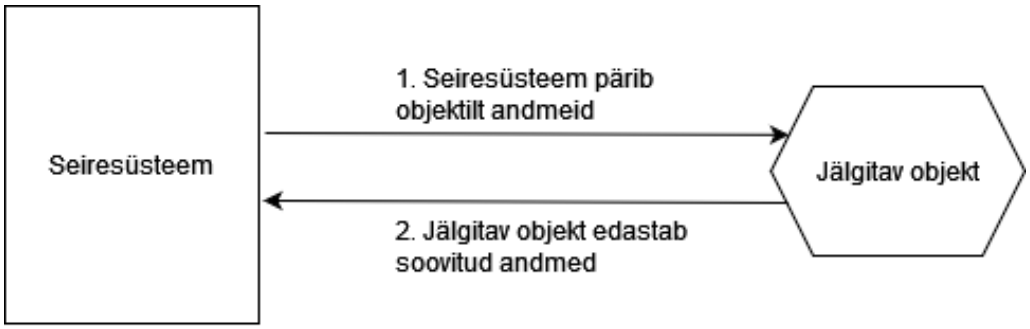
Pull-meetodi kasutamiseks on vaja rakendust, mis seiresüsteemile vajalikud andmed kättesaadavaks teeb. Seejärel pärib seiresüsteem seadmelt andmeid ise. Levinuimad

[1] lahendus, kus puudub tsentraalne server. Selle asemel on mitu üksteisest eraldatud serverit, ehk sõlme

variandid andmete kättesaamiseks *pull*-meetodiga on:

- *SNMP*: seiresüsteem saadab *SNMP*-päringud seadmetele ja need vastavad küsitud andmetega.
- *HTTP*: seiresüsteem pärib seadmelt andmeid, mida on võimalik üle *HTTP* edastada. Sinna alla kuuluvad näiteks rakendusliidesed.

Pull-meetod kasutab rohkem võrguresursse kui *push*, seega suuremate seirelahenduste korral tuleks eelistada viimast[15]. Joonis 2 kuvab meetodi andmeedastusvoogu.



Joonis 2. *Andmevoo liiklus pull-meetodis*

2.3 Metoodikad

Metoodika üldistatud definitsioon on meetodite ja protseduuride kogum, mida kasutatakse projektides[18]. Metoodika tüüpe, mis pakuvad erinevaid lahendusviise süsteemide jälgimiseks, on mitu. Üldlevinud on *USE* ja neli kuldset signaali. Neid metoodikaid saab rakendada mõõdikukogumikel. Mõõdikukogumik on oluline komponent seire standardiseerimisel.

2.3.1 USE metoodika

USE (utilization, saturation and errors) metoodika on mõeldud kasutamiseks süsteemi jõudluse uurimisel, kohe pärast probleemi tekkimist, et kiiresti jälile saada kitsaskohtadele. See põhineb kolmel mõõdikul[19]:

- Koormus: ressursside praegune koormustaseme mõõtmine. Kasutuse analüüs aitab tuvastada ressursid, mis on ülekoormatud või alakasutatud.
- Ülekoormus: keskendub süsteemi jõudluse piiridele. See hõlmab ressursside

koormustaseme jälgimist kriitilistel punktidel, kus ülekoormus võib põhjustada probleeme.
- Vead: keskendub süsteemi vigade esinemisele.

USE metoodika eesmärk on anda hinnang kogu süsteemi jõudlusele, tuvastades sealhulgas võimalikud pudelikaelad ja probleemid. Selle põhjal saavad süsteemiadministraatorid teha paremaid valikuid, et süsteemi efektiivselt hallata ja optimeerida.

2.3.2 Nelja kuldse signaali metoodika

Neli kuldset signaali (*The Four Golden Signals*) on Google SRE tiimide poolt välja töötatud komplekt mõõdikuid. Mõõdikuteks on neli "signaali":

- Peiteaeg: aeg mis kulub päringu teenindamiseks.
- Liiklus: nõudlus teenusele.
- Vead: ebaõnnestunud päringud.
- Ülekoormus: piir, mil ressursi efektiivsus langeb.

Google'i sõnul peaks keskenduma eelmainitud mõõdikute jälgimisele, kui on võimalik jälgida ainult nelja kasutajakeskse teenuse või süsteemi näitajat[20].

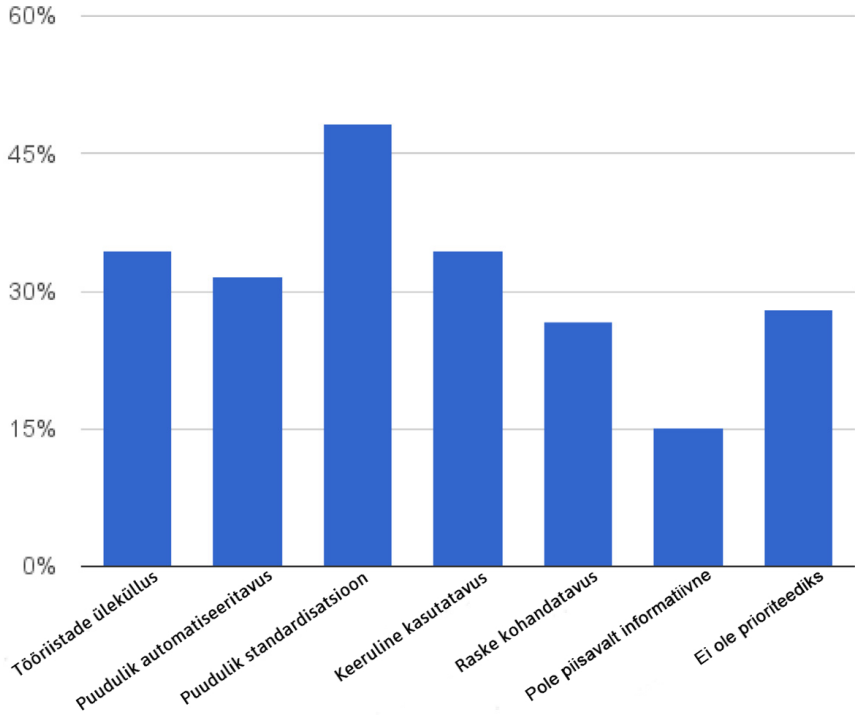
3. Probleem

Omnivas kasutatakse mitut seiretööriista. See raskendab ettevõttes intsidentide korrelatsioonide leidmist ja nendele reageerimist. Ettevõttes puudub seire standardisatsioon; ei ole välja toodud parimaid praktikaid alarmide ja reeglite ülesseadmisel, mistõttu tekib praegustes süsteemides liiga palju alarmidest tingitud müra.

Näide: *iga päev kindlal ajavahemikul saadetakse klientidele teavitusi. Seetõttu tõuseb ressursside kasutus süsteemis. Kuna seiresüsteem ei tea, et see on normaalne, siis saadab ta administraatorile teavituse. Lisaks on paljud jälgitavad seadmed mitmes seiresüsteemis korraga esindatud, mistõttu tekivad duplikaatalarmid.*

3.1 Probleemi ulatus

Omniva pole ainuke ettevõte, kus esineb selliseid probleeme. 2020. aastal läbi viidud uuring "*Cloud applications monitoring: An industrial study*"[21], heidab valgust IT-taristu seire tavadele ettevõtetes. Uuring leidis, et kõige suurem väljakutse, mida tajutakse seire rakendamisel, on standardisatsiooni puudumine ja seiretööriistade üleküllus (Joonis 3). Küsitluses osales üle 140 inimese enam kui 70 erinevast organisatsioonist.



Joonis 3. *Kõige kriitilisemad probleemid seire rakendamisel[21]*

3.2 Soovituslik lahendus

Ülaltoodud nõrkuste adresseerimiseks ja seire parandamiseks soovitab autor kasutusele võtta uue seiresüsteemi. Autor koostab üleüldise juhendi seire standardiseerimiseks. Uus süsteem peab vastama ettevõtte IT-vajadustele, mis on kirjeldatud 6.1 peatükis.

Eesmärk on ettevõttes olevaid seiresüsteeme vähendada ja katta seirega ära rohkem seadmeid ning teenuseid. Standardiseerimisjuhend aitab ettevõtte seireprotsessi muuta läbipaistvamaks ja efektiivsemaks, lihtsustades seire rakendamist IT-taristule.

4. Hindamismetoodika

Käesolevas peatükis tutvustab autor hindamismudelit ning analüüsitavaid teemasid. Vastavalt teemale on kasutatud erinevaid uurimistöid, mis aitavad hinnata teema üleüldist olulisust ning mõnel juhul ka selle struktuuri. Seiresüsteeme analüüsitakse 5. peatükis.

Parima seiresüsteemi väljavalimiseks koostatakse hindamismudel, millega saaks hinnata seiresüsteemi vastavust kriteeriumitele. Hindamismudelit rakendatakse 6.1 peatükis.

4.1 Analüüsitavad teemad

Analüüsitavaid teemasid on kokku viis:

1. Konfiguratsioon ja haldamine.
2. Automatiseerimine.
3. Alarmeerimine.
4. Andmete visualiseerimine ja talletamine.
5. Dokumentatsioon ja tugi.

4.1.1 Konfiguratsioon ja haldamine

Efektiivne konfiguratsioon ja haldus aitavad tagada süsteemi töökindluse ja vähendada inimtekkelisi vigu. Mitmete uuringute kohaselt on konfiguratsioonivead ühed levinumad tarkvararikke põhjustajad[22, 23, 24]. 2016. aastal avaldatud aruanne leidis, et faili-põhine konfiguratsioon põhjustab süsteemiadministraatoritel kognitiivseid raskusi[25]. Seetõttu on oluline, et konfiguratsiooni parameetrid oleksid arusaadavad ja kergelt kohandatavad.

Võrdluse käigus saab selgemalt näha, kuidas iga süsteem võimaldab neid protsesse ellu viia. Näiteks võib üks süsteem pakkuda halduseks kasutajaliidest, mis võimaldab kiiresti seireparameetreid määrata ja teavituste seadistusi hallata. Teisalt võib teine süsteem lasta konfiguratsiooni muuta ainult failidega ja selle haldamine võib olla aeganõudvam.

4.1.2 Automatiseerimine

Automatiseerimine on üks tähtsamaid ülesandeid süsteemiadministraatori töös. Süsteemihaldust automatiseerivaid skripte luues on süsteemiadministraatoritel rohkem aega, et

juurutada uusi lahendusi ja lahendada kriitilisemaid probleeme[26, 27, 28].

Automatiseerimist käsitlevas osas toob autor välja võimalused süsteemi töövoo automatiseerimiseks. Antud võimaluste alla kuuluvad liidesed, mis aitavad vähendada korduvaid tegevusi (seadmete lisamine seiresüsteemi, graafikute tegemine jms).

4.1.3 Alarmeerimine

Alarmeerimine võimaldab koheselt teavitada probleemist süsteemis, misjärel saab läbi viia automaatseid või manuaalseid toiminguid selle kõrvaldamiseks.

Selles alampeatükis selgitab autor välja, kas süsteem võimaldab alarme seadistada ja neid saata erinevate kanalite kaudu (näiteks e-post, SMS või MS Teams). Lisaks uurib autor, mis võimalusi seiresüsteem alarmeerimise osas (teavituste rotatsioon, alarmide teadvustamine, tasemete määramine jm) veel pakub.

4.1.4 Andmete visualiseerimine ja talletamine

Andmete visualiseerimine aitab seire käigus kogutud andmeid analüüsida, valideerida, ning struktureerida[29]. Andmete talletamine määrab ära, kui kaua on andmed kättesaadavad ja visualiseeritavad, enne kui need kustutatakse.

Võrdluse käigus toob autor välja andmete visualiseerimise ja talletamise võimalused süsteemis.

4.1.5 Dokumentatsioon ja tugi

Enamus kasutajaid vajab süsteemide kohta teavet, et olla võimelised neid efektiivselt haldama ja käsitsema. Dokumentatsioonil on oluline roll kasutaja rahulolu säilitamisel (kuna see pakub juhiseid keerukamate toimingute lahendamiseks) ja kasutaja teadmiste täiendamisel[30].

Dokumentatsiooni ja toe alapeatükis hindab autor süsteemi dokumentatsiooni struktuuri, lähtudes Sommerville'i välja pakutud viiest osast[31]:

- Funktsionaalne kirjeldus: süsteeminõuded ja olemasolevad funktsioonid.
- Paigaldusjuhend: detailsed juhised süsteemi paigaldamiseks konkreetsesse keskkonda.
- Sissejuhatav peatükk: süsteemi tutvustus ja standardfunktsioonide kirjeldus.

- Mõistete register: funktsioonide ja veateadete kirjeldus.
- Süsteemiadministraatori juhend: üldine teave süsteemi haldamiseks.

Kui süsteemil on rakendusliides (*API*), hindab autor selle dokumentatsiooni eraldi, kasutades selleks 2015. aastal läbi viidud uuringut *"How API Documentation Fails"*[32], mis analüüsis rakendusliideste dokumentatsiooni kvaliteeti. Uuringus osales enam kui 300 tarkvaraspetsialisti.

Lisaks toob autor välja, kui tihti seiresüsteeme hooldatakse ning uuendusi ja vea paikamisi pakutakse. Autor selgitab välja tegevuskava olemasolu, kui kaugele tulevikku see on planeeritud ning kas seda hoitakse ajakohasena.

4.2 Hindamismudel

Süsteemide hindamiseks annab autor igale süsteemile kriteeriumidele vastamise eest punkte (kirjeldatud peatükis 6.1). Punktide jaotamise struktuur on järgmine:

- Kriteerium on täidetud: 1 punkt.
- Kriteerium on osaliselt täidetud (lahendus on puudulik või peab selle eesmärkide saavutamiseks kasutama kolmanda osapoole tarkvara): 0,5 p.
- Kriteerium pole täidetud: 0 p.

Punktid korrutatakse kriteeriumi olulisuse tasemega. Olulisuse tasemed on: madal (x1), keskmine (x2), kõrge (x3).

5. Kasutusel olevate süsteemide analüüs

Eesolevas peatükis tutvustab autor Omnivas kasutusel olevaid seiresüsteeme ja analüüsib neid. Analüüsi käigus võrreldakse seiretööriistade võimekusi ja kitsaskohti, mille tulemina saab süsteemi vajalikkust ettevõttes hinnata.

5.1 Ülevaade

Ülevaates vaadeldakse nelja seiresüsteemi: Nagios Core, Cacti, Prometheus ja Zabbix. Tutvustatakse nende laiemat kasutusala ning kuidas Omnivas neid kasutatakse.

5.1.1 Nagios Core

Nagios Core on vabavaraline seiresüsteem, mis loodi 1999. aastal NetSaint nime all[33]. 14 aastat hiljem ilmus Nagios Core 4, mis on tänaseni Nagios Core üldversioon. Süsteem on mõeldud serveri ressursside (protsessori koormus, mälu kasutus jms), võrguseadmete, avalike teenuste (HTTP, SSH) ja andmebaasi päringute seiramiseks. Nagios Core'i on võimalik kasutada agendiga või ilma (*agentless*). Agenti kasutades on võimalik jälgida masina teenuseid ja omadusi, mis pole üle võrgu kättesaadavad[34].

Omnivas on Nagios Core kasutusel põhiliselt Windowsi-põhiste serverite seiramiseks: jälgitakse teenuste olekut, kettaruumi ja protsessori koormust, kuid ka andmebaasi päringuid. Nende parameetrite kättesaamiseks on masinatele paigaldatud NRPE-agent. Andmete kogumine toimub pistikprogrammide abil, mis paigaldatakse NRPE-agendiga samasse masinasse.

5.1.2 Cacti

Cacti on avatud lähtekoodiga seiretööriist, mis kasutab RRDTooli andmete salvestamiseks ja graafikute loomiseks[35]. Cacti loodi 2001. aastal ja seda hooldab The Cacti Group Inc. Cacti on mõeldud põhiliselt võrguseadmete seireks, kuna sellel pole agenti ja see kasutab andmete kogumiseks põhiliselt *SNMP*-protokolli[36].

Omniva kasutab Cactit võrguseadmete, sealhulgas ruuterite ja kommutaatorite seireks. *SNMP* abil jälgitakse nende võrguliideste kaudu toimuvat andmevahetust ja protsessori töö koormust.

5.1.3 Prometheus

Prometheus on vabavaraline seire ja häire tööriistakomplekt, mis töötati välja 2012. aastal SoundCloudis ettevõttesiseseks kasutamiseks. Komplekti kuuluvad:

- PromQL: päringukeel mõõdikute pärimiseks ja agregeerimiseks.
- TSDB: kronoloogiline andmebaas mõõdikute hoidmiseks.

Agentide asemel on Prometheusel *exporter*id, mis koguvad kliendi masinas teatud teenuse kohta infot ja avalikustavad selle info HTTP kaudu. Seejärel saab Prometheus, *pull*-meetodit kasutades andmed kätte. Komplekt on arendatud eelkõige pilve- ja konteinerkeskkonna seiramiseks[37].

Prometheust kasutatakse Omnivas koos visualiseerimistarkvaraga Grafana. Koos moodustavad need kaks süsteemi lahenduse, mis toetab nii seiret kui ka andmete graafilist esitamist. Antud lahenduse abil jälgitakse Omnivas mitmeid seadmeid ja teenuseid:

- Andmebaase.
- Servereid (GNU/Linux).
- Kubernetese instantse.
- Siserakendusi.
- Pakiautomaate.

5.1.4 Zabbix

Zabbix on 2001. aastal loodud avatud lähtekoodiga seiretarkvara, mis on mõeldud IT-taristu jälgimiseks. Seda arendab Zabbix LLC. Zabbix toetab nii agendiga kui ka agendita seiret, kasutades selleks *SNMP*, *SSH*, *IPMI* jm protokolle[38].

Zabbixit hetkel aktiivselt Omnivas ei kasutata, süsteemis on hetkel mõni üksik võrguseade, mida jälgitakse *SNMP* abil. Ühtegi teist seadmetüüpi või teenust Zabbixis ei hallata.

5.2 Võrdlus võimekustest ja kitsaskohtadest

Käesolevas peatükis viib autor läbi põhjaliku võrdluse eelkirjeldatud süsteemidele, mis on Omnivas aktiivselt kasutusel. Süsteemide analüüs aitab aru saada, milline jälgimissüsteem on alakasutatud ning milline vajaks asendamist. Lisaks annab võrdlus lugejale ülevaate süsteemide ajakohasusest.

Võrdlus hõlmab olulisi aspekte nagu konfiguratsioon, haldamine, automatiseerimine, dokumentatsioon ja tugi. Selle analüüsi käigus kogutud teave moodustab olulise aluse järgnevateks etappideks.

5.2.1 Nagios Core

Konfiguratsioon ja haldamine

Nagios Core'i haldamine on keerukas protsess, kuna Nagiose konfigureerimine käib mitme teksti-põhise failiga[39, 40]. Kasutajaliideses ja käsureal see funktsionaalsus puudub, mistõttu on isegi paari seadme lisamine Nagiosse ajakulukas. Konfiguratsiooni failide muutmiseks peab oluliselt toetuma Nagiose dokumentatsioonile, kuna iga väiksemgi viga võib Nagiose viia veaseisundisse. Vigu failides aitab leida validaator[41].

Seadmete lihtsamaks haldamiseks on Nagiosel olemas mallid, mis võimaldavad mitmete seadmete ja teenuste vahel konfiguratsioone[42] kopeerida. See eeldab, et seadmetel on vajalikud pistikprogrammid ja agent eelnevalt paigaldatud.

Automatiseerimine

Nagios Core'i automatiseerimise võimalused on oluliselt piiratud. Puudub rakendusliidese tugi, mille kaudu saaks skriptimise abil süsteemi haldamist automatiseerida. Seetõttu sõltub konfiguratsiooni automatiseerimine kolmanda osapoolte programmidest nagu Ansible, ja nende piirangutest. Nagiose konfiguratsioonifailid ei kasuta üldtuntud formaate nagu *YAML*, *JSON* või *INI*, mis raskendab automatiseerimist veelgi.

Suuremahulise IT-taristu jaoks puudub Nagios Core'il automaatne seadmetuvastus-funktsioon, mis võimaldaks värskelt üles seatud seadmeid seiresüsteemi lisada. Seda puudust saab osaliselt katta kasutades *IaC* tööriistu. Populaarsematest *IaC* tööriistadest on Nagios Core'i haldamise tugi ainult Chef'il[43]. Seadmeid, mis kasutavad suhtluseks *SNMP*-protokolli, automaatselt lisada ei saa.

Alarmeerimine

Nagios Core toetab alarmide konfigureerimist ning teadvustamist, kuid ei toeta teavituste saatmist. Teavituste saatmiseks on vaja kasutada kolmandate osapoolte tehtud pistik-rakendusi[44]. Teavituste saatmiseks on Nagiosel 18 pistikrakendust, nende hulgas tugi teadete saatmiseks MS Teamsi sõnumi, SMSi kui ka e-postina[45].

Nagios Core'il on võimalik seada neli alarmi tähtsuse taset[46]:

- *OK*: korras (lahenenud).
- *Warning*: hoiatus.
- *Unknown*: teadmata.
- *Critical*: kriitiline.

Andmete visualiseerimine ja talletamine

Andmete visualiseerimiseks läheb Nagiosel vaja lisarakendust PNP4Nagios või MRTG[47]. Andmete talletamiseks kasutab Nagios lisarakendust NDOUTILS[48]. Salvestusperiood on kõikide andmete kohta üks, spetsiifilise teenuse või seadme salvestusperioodi määrata ei saa.

Dokumentatsioon ja tugi

Nagios Core'i dokumentatsiooni[49] struktuur vastab näiliselt Sommerville väljapakutule, puuduseks on ainult mõistete register. Dokumentatsiooni lugedes nähtub, et süsteeminõuetes on ainsateks tingimusteks Linuxi-põhine operatsioonsüsteem ja võrguühendus. Täpsustamata on nõuded süsteemi andememahu, vahemälu ja protsessori võimsuse suhtes, mistõttu on keeruline hinnata Nagios Core'i skaleeritavust ja ressursside kasutust. Lisaks on dokumentatsioonis viiteid aegunud agendile[50], mille arendus lõpetati 2020. aastal[51].

Tootjapoolne tugi on väga aeglane ja versiooniuuendused irregulaarsed: süsteemile pole üle viie aasta lisatud ühtegi täiendust[52]. Selle aja vältel on avaldatud 13 uut versiooni, kus on kokku parandatud 83 viga.

Nagios Core'il on avalik tegevuskava[53], mis kirjeldab plaane 5. versiooni loomiseks. Pole teada, millal uus versioon ilmavalgust näha võiks. Samuti on see uuendamata vähemalt aastast 2020[54].

5.2.2 Cacti

Konfiguratsioon ja haldamine

Cactit saab konfigureerida ja hallata nii veebipõhise kasutajaliidese kui ka käsurea abil. Seega on kasutajal võimalik valida endale sobivam viis süsteemi haldamiseks[40].

Veebipõhise kasutajaliidese kaudu saab seadistada ja hallata Cacti töökeskkonda. Selle liidese abil saab lihtsasti lisada uusi seadmeid, jälgida seire andmeid ja hallata graafikuid.

Cacti käsurea liidese kasutamine pakub suuremat paindlikkust ning annab administraa-

toritele täpsema kontrolli süsteemi konfiguratsiooni üle. Võrreldes kasutajaliidesega võimaldab käsurida oluliselt kiiremini lisada uusi seadmeid, koostada graafikuid ja defineerida malle.

Automatiseerimine

Cacti dokumentatsioonis[55] on automatiseerimise kohta eraldi peatükk, mis kirjeldab automatiseerimise võimalusi kasutajaliideses. Lisaks saab olemasoleva käsurea liidese kaudu automatiseerida teatud ülesandeid (näiteks lisada seadmeid ja teha graafikuid). Nii nagu Nagiosel, pole ka Cactil rakendusliidese tuge.

Cacti võimaldab automaatselt jälgida seadmeid, mis on tuvastatud võrgu skaneerimise käigus. Selle jaoks peab lisama Cactisse alamvõrgu aadressi ja *SNMP*-protokolli andmed.

Alarmeerimine

Alarmide ja reeglite ülesseadmise jaoks on Cactil pistikprogramm, millega saab saata teateid e-posti, *Syslog*'i või *SNMP* kaudu[56]. Muud lisafunktsionaalsust pistikprogramm ei paku.

Andmete visualiseerimine ja talletamine

Cactil on sisseehitatud graafikute tugi. Lisaks on graafikutel mallide tugi, mis võimaldab tuvastatud seadmetele automaatselt graafikuid lisada[36].

Andmete talletamise perioodi saab seada iga malli kohta eraldi[36].

Dokumentatsioon ja tugi

Cacti dokumentatsioon[55] on üpris mahukas, see katab ära pea kõik Sommerville'i poolt pakutud osad. Nagu Nagiosel, pole ka Cactil süsteemiressursi nõudeid, vaid on välja toodud ainult tarkvaralised sõltuvused. Lisaks kirjalikule materjalile pakub Cacti erinevaid juhendeid videote näol. Mitmete uuringute põhjal on video väärtuslik abivahend juhendamisel ja õppimisel[57, 58].

Cacti viimane suurim versioonimuutus toimus pea viis aastat tagasi[59] (v2.1.38 -> v2.2.0). Uuendused on viimastel aastatel aeglustunud: kui varem toimus kaheksa uuendust aastas, siis viimase kolme aasta vältel on välja lastud kolm uuendust aastas.

Cactil oli avalik tegevuskava kuni 2017. aasta juunikuuni[60]. Pole teada, miks see eemaldati.

5.2.3 Prometheus

Konfiguratsioon ja haldamine

Konfiguratsioonifaile on Prometheusel üks, ning see kasutab üldtuntud *YAML*-formaati. Prometheusega kaasa tuleva rakenduse Promtool abil saab konfiguratsioonifaili enne selle süsteemi salvestamist valideerida. Selle kasutamine aitab tuvastada failides tekkivaid konfiguratsioonivigu. Lisaks konfiguratsioonifailile saab *flag*'e kasutades seada mõningaid parameetreid Prometheuse käivitamisel.

Prometheuse haldamine käib läbi sama konfiguratsioonifaili, kuhu saab lisada töid ja alarmireegleid. Töö annab Prometheusele teada, millise veebiaadressi kaudu seadme andmeid koguda ja kui pika ajaintervallidega seda teha. See sisaldab endas unikaalset nime ja jälgitavas seadmes asuva eksporteri asukohta (tavaliselt formaadis *IP:Port* või *DNS:Port*).

Automatiseerimine

Kasutatav *YAML*-formaat võimaldab Prometheust automaatselt hallata *IaC* tööriistadega, nagu Ansible, Puppet, Chef. Neil kõigil on olemas ka tugi Prometheuse jaoks[37]. Nende tööriistadega saab automaatselt lisada konfiguratsioonifaili töid ja alarmi reegleid.

Prometheusel on sisseehitatud teenuse avastamise tugi, mis võimaldab automaatselt seadmetelt mõõdikuid koguda. Avastamine saab toimuda kolmel viisil[37]:

- Lugedes *IaC*-tööriistaga täidetud faile.
- Pärides rakendusliidest.
- Pärides *DNS*-kirjeid.

Kõik kolm viisi peavad enne olema kirjeldatud konfiguratsioonifailis (näiteks millist rakendusliidest pärida või millist faili lugeda).

Prometheusel on olemas kaks rakendusliidest: haldusliides ja päringuliides. Haldusliidesese abil saab teha lihtsamaid toiminguid, näiteks kontrollida Prometheuse olekut (aktiivne või mitteaktiivne), uuesti laadida konfiguratsioonifaili ja väljuda programmist. Päringu rakendusliidese kaudu saab Prometheuse andmebaasis olevaid mõõdikuid pärida.

Alarmeerimine

Prometheus laseb alarmi reegleid defineerida eraldi failis, kuid teadete saatmiseks läheb vaja eraldiseisvat tarkvara nagu Alertmanager[37] või Grafana[61]. Grafanal on sisse-ehitatud tugi saata teavitusi emailile, SMSile ja MS Teamsi. Samuti võimaldab Grafana alarme teadvustada, kasutades selleks pistikrakendust "OnCall"[62].

Andmete visualiseerimine ja talletamine

Kuigi Prometheus on võimeline andmeid visualiseerima, on siiski soovituslik kasutada visualiseerimistarkvara Grafana, mis võimaldab Prometheuse andmebaasist andmeid pärida ja nende põhjal näidikpaneele genereerida[61].

Prometheus soovitab pikaajaliseks andmete talletamiseks kasutada eraldiseisvat lahendust[63]. Mõned neist, näiteks VictoriaMetrics[64], võimaldavad talletada mõõdikuid erinevate säilimistähtaegadega.

Dokumentatsioon ja tugi

Nii nagu eelnevad süsteemid, ei vasta Prometheuse dokumentatsioon[65] täielikult Sommerville'i struktuurile: sellel puuduvad nii mõistete register kui ka täismahus süsteeminõuded. Olemasolevad süsteeminõuded pole koondatud ühte peatükki, vaid on hajutatud dokumentatsiooni erinevate osade vahel.

Prometheuse rakendusliideste dokumentatsioonidel pole ühtegi kriitilist puudust, mis on välja toodud uuringus *"How API Documentation Fails"*. Mõlema rakendusliidese dokumentatsioonid on ajakohased ja sisaldavad näiteid.

Prometheusel algab iga kuue nädala järel uus väljalasketsükkel, pärast kuut nädalat ei saa need üldjuhul enam veaparandusi. Lisaks tavaversioonidele, mis lastakse välja iga kuue nädala järel, on Prometheusel ka *LTS* (*Long Term Support*) versioonid, mis saavad turva- ja veaparandusi ühe aasta vältel[66].

Prometheusel on olemas avalik tegevuskava[67]. Sarnaselt Nagiosele ei sisalda see ühtegi kuupäeva, mil konkreetne eesmärk võiks täidetud olla. Lisaks on seda viimati uuendatud aastal 2021[68].

5.2.4 Zabbix

Konfiguratsioon ja haldamine

Zabbixit saab konfigureerida veebipõhise kasutajaliidese ja rakendusliidese kaudu, välja arvatud jõudluse häälestamisel, millisel juhul tuleb parameetreid määrata failis.

Zabbixis on seadmete haldamiseks mallide tugi. Malle kasutatakse sageli konkreetsete mõõdikute grupeerimiseks, seejärel rakendatakse mallid süsteemidele, kus soovitakse jälgida samu parameetreid. Nende kasutamine muudab Zabbixi haldamise efektiivsemaks, eriti suurte ja keerukate IT-taristute puhul.

Automatiseerimine

Zabbixi rakendusliides võimaldab automatiseerida konfigureerimise ja haldamisega seotud tegevusi[69] nagu mõõdikute lisamine, gruppide loomine jm. Lisaks saab rakendusliidese abil integreerida Zabbixit erinevate süsteemidega.

Zabbixi automaatseks ülesseadmiseks ja agentide paigaldamiseks saab kasutada *IaC* tööriistu Puppet või Ansible, neil mõlemal on ametlik Zabbixi tugi[70, 71].

Zabbix toetab seadmete automaatset tuvastamist, seda nii agenti kasutatavate kui ka *SNMP* abil jälgitavate seadmete puhul. Agendipõhistel seadmetel tuleb konfigureerida aadress, mille kaudu süsteemiga suheldakse. *SNMP*-seadmetel tuleb Zabbixis määrata *SNMP*-parameetrid ja alamvõrgu aadress. Peale automaatset tuvastamist saab Zabbix seadmed süsteemi lisada, neid grupeerida, lisada neile külge malle, jooksutada skripte jpm.

Alarmeerimine

Zabbixil on sisseehitatud tugi alarmide seadistamiseks, teadvustamiseks ning saatmiseks[69]. Teavitusi saab saata nii e-posti, SMSi kui ka MS Teamsi sõnumina.

Zabbixis on alarmile võimalik määrata kuus erinevat tähtsuse taset[69]:

1. *Not classified*: määramata.
2. *Information*: teave.
3. *Warning*: hoiatus.
4. *Average*: keskmine.
5. *High*: kõrge.
6. *Disaster*: katastroof.

Andmete visualiseerimine ja talletamine

Zabbixis on sisseehitatud andmete visualiseerimise võimekus, mis võimaldab luua näidikpaneele ning erinevaid graafikuid.

Talletamisaega saab Zabbixis määrata globaalselt malli-, seadme- või mõõdikupõhiselt[69].

Dokumentatsioon ja tugi

Zabbixi dokumentatsiooni[72] struktuur vastab täielikult Sommerville'i väljapakutud struktuurile, sellel on olemas kõik viis osa. Sisu poolest on dokumentatsioon väga mahukas ja detailne. Nii nagu Cacti, pakub Zabbix ka videomaterjale.

Zabbixi rakendusliidese dokumentatsiooni kvaliteet on väga hea: kõikidel funktsioonidel on olemas detailsed kirjeldused ja kasutusnäited, samuti hoitakse neid ajakohastena.

Zabbixi meeskond uuendab süsteemi iga kuue kuu järel ning pakub täismahulist tuge (üldiste, kriitiliste, ning turvavigade paikamine) senisele versioonile kuni uue versiooni ilmumiseni[73].

Zabbixil on olemas avalik tegevuskava[74]. Seal on kirjeldatud, milliseid funktsioone võib järgmistes versioonides oodata ja nende arenduse hetkeseisu (valmis, töös).

6. Uue seiresüsteemi valik

Seiresüsteeme hinnatakse peatükis 4.2 välja toodud hindamismudeliga. Hinnangu põhjal soovitatakse ettevõttele seiresüsteemi, millega saab asendada kriteeriume mitte täitvad süsteemid.

6.1 Süsteemide hindamine

Seiresüsteemi valimisel on oluline määratleda kriteeriumid, mis vastaksid ettevõtte vajadustele. Süsteemile antud punktid on kuvatud tabelis 1 (kõrgem punktisumma on parem). Süsteemi kriteeriumid jagunevad funktsionaalseteks (F) ja mittefunktsionaalseteks (MF):

- F1: pakub haldamiseks erinevaid võimalusi.
- F2: on olemas rakendusliidese tugi.
- F3: seadmete automaatne avastamine.
- F4: võimalus alarme seadistada.
- F5: võimekus saata teavitusi SMSi, e-post ja MS Teamsi teel.
- F6: võimalus andmeid visualiseerida erinevate graafikute abil.
- F7: võimalus perioodiliselt andmeid säilitada.
- MF1: kvaliteetne dokumentatsioon.
- MF2: aktiivne arendus tootja poolt.
- MF3: avalik tegevuskava.
- MF4: ametlik tugi erinevate teenuste ning seadmetüüpide jälgimiseks.

Tabel 1. *Seiresüsteemi hinnang*

Kriteerium	Olulisuse tase	Nagios Core	Cacti	Prometheus	Zabbix
F1	Madal	0	1	1.5	1
F2	Kõrge	0	0	1.5	3
F3	Kõrge	3	3	3	3
F4	Kõrge	3	1.5	3	3
F5	Kõrge	1.5	0	1.5	3
F6	Kõrge	0	0	1.5	3
F7	Keskmine	0	2	1	2
MF1	Madal	0	0.5	0.5	1
MF2	Kõrge	0	1.5	3	3
MF3	Madal	0.5	0	0.5	1
MF4	Kõrge	3	0	3	3
KOKKU		11	9.5	17	26

6.2 Süsteemi valik

Seatud kriteeriumidele vastas kõige paremini Zabbix, mis kogus maksimumpunktid (26), sellele järgnes Prometheus 17 punktiga. Viimasteks jäid Nagios Core ja Cacti, mis said vastavalt 11 ja 9.5 punkti.

Hindamise käigus selgus, et Cacti ja Nagios Core ei vasta suurele osale funktsionaalsetest ja mittefunktsionaalsetest nõuetest. Prometheus võimaldab enamiku nõuetest täita kolmanda osapoole rakendustega. Zabbixil pole aga vaja lisatarkvara, et vastata seatud kriteeriumitele

Nende tulemuste põhjal soovitab autor välja vahetada kaks kõige madalama punktisummaga seiretööriista - Nagios Core ja Cacti - ning asendada need täismahus Zabbixiga. Tööriistad keskenduvad põhiliselt serverite ja võrguseadmete seirele, mida võimaldab ka Zabbix.

Prometheusel on ettevõttetes tihipeale veidi erinev rakendus. See on eelkõige mõeldud konteiner ja pilvekeskkonna seireks. Omniva kasutab seda ka siserakenduste jälgimiseks. Kuna Prometheuse ja Zabbixi arhitektuurid on erinevad, vajab siserakenduste seire üleviimine ettevõttelt suuri investeeringuid. Võttes arvesse, et Prometheus saab täita kõiki kriteeriume, ei pea autor Prometheuse täismahus asendamist vajalikuks.

7. Seire standardiseerimine

Standardiseerimise eesmärk on tagada, et seadmeid jälgitaks järjepidevalt ja ühtlustatud viisil. See tähendab, et olenemata seiresüsteemist käib andmete kogumine kindlate kriteeriumite ja piirmäärade järgi ning mõõdikute ja alarmide konfiguratsioon on ühtne. See teeb tõrkeotsimise ning juurpõhjuse leidmise efektiivsemaks. Standardisatsiooni rakendamisel on lihtsam reageerida muutustele, mis võivad tekkida ettevõtte või seire arenemisel.

Lisaks sellele aitab seire standardiseerimine tõsta seireprotsessi efektiivsust ja vähendada selle keerukust. Ühtse standardi väljatöötamine lihtsustab seiresüsteemide haldamist ja skaleerimist ning suurendab kaetavust jälgitavate süsteemide ja teenuste osas.

7.1 Mõõdikukogumikud ja seiregrupid

Mõõdikukogumikud võimaldavad kohandada andmeseiret vastavalt organisatsiooni konkreetsetele vajadustele. See võib hõlmata erinevate teenuste või süsteemi mõõdikute ning alarmide seadmist vastavalt vajadustele. Seiregrupid aitavad luua selge struktuuri, kus sama eesmärki täitvad seireelemendid (seadmed, teenused, rakendused) on ühendatud. Nende alusel saab määratleda mõõdikukogumikud vastavalt iga grupi konkreetsetele vajadustele.

Näide: *ettevõttes on neli serverit ja üks võrgulüüs. Servereid on kahte tüüpi: üks Windowsi-põhine ja kolm Linuxi-põhist. Windowsi serveris jookseb andmebaas MSSQL, mis teenindab Linuxi servereid. Linuxi serverites jookseb Apache'i veebiserver, mis on klientidele kättesaadav. Ettevõttel oleks vaja jälgida nii seadmete jõudlust kui ka andmebaasis olevate tabelite suurust*

Tulenevalt näitest moodustub kolm seiregruppi, mis on koostatud vastavalt jälgimisvajadustele:

1. Seadmed: siia kuuluvad kõik neli serverit ja võrgulüüs.
2. Kliendikesksed teenused: siia kuuluvad kõik neli serverit.
3. Andmebaasid: siia kuulub andmebaas.

Andmete ühtsuse tagamiseks tuleks iga elemenditüübi kohta defineerida mõõdikukogumik, mida jälgida. Mõõdikukogumiku loomisel saab võtta põhjaks *USE* metoodika,

kliendikeskse elemendi puhul lisaks *USE*-metoodikale ka kolm mõõdikut nelja kuldse signaali metoodikast: peiteaeg, liiklus ja teenuse vead. Nende mõõdikute kogumine aitab elementidel analüüsida jõudlust ja kvaliteeti.

Näite põhjal jaguneksid mõõdikud nii: Seiregruppi "Seadmed", kuuluvad mõõdikud mis on kõikidel viiel seadmel ühised:

- Protsessori koormus.
- Mälukasutus.
- Võrguliidese läbilaskemaht.

Seiregrupi "Kliendikesksed teenused" mõõdikud:

- Aeg, mis serveril kulus kliendile vastamiseks.
- Aktiivsete ühenduste arv.
- Ebaõnnestunud päringute hulk.

Seiregrupp "Andmebaasid", mis koosneb andmebaasi mõõdikutest:

- Tabelite suurus andmebaasis.

Sellist struktuuri saab kergelt rakendada uute seadmete puhul. Näiteks kui IT-taristule lisatakse juurde üks andmebaasi server, on selge, et see kuulub kõikidesse ülalmainitud gruppidesse, misjärel saab sellele kohaldada vastavad mõõdikukogumikud.

7.2 Alarmid

Alarm on kõrvalekalle normaalsest süsteemi toimimisest, mis nõuab operaatori sekkumist[75, 76]. Operaatori teavitamine alarmi tekkest võimaldab kiiremini reageerida potentsiaalsetele intsidentidele.

Standardisatsiooni puudumine toob endaga tihtipeale kaasa valesti seadistatud alarmid, mis omakorda tähendab rohkem müra. Müra esineb tavaliselt kahel kujul:

- Arusaamatud alarmid, kui alarmi teates ei ole piisavalt kasulikku teavet[75].
- Alarmide üleliigsus, mille põhjustajateks on tihtipeale erinevate reeglite ja piirväärtustega koostatud alarmid[77].

Probleemsem neist kahest on alarmide üleliigsus. Suurtele kogustele alarmidele ei jõuta

õigeaegselt reageerida, mis põhjustab alarmide ignoreerimist ja nende vaigistamist. Seda nähtust on eelkõige uuritud meditsiinis[78, 79, 80, 81], mis on vaieldamatult kõige suurema vastutusega tööala, kuna kaalul on inimelud.

Probleemi adresseerimiseks tuleks alarmile seada võimalikult täpsed piirmäärad ja vältima hüstereesi. Hüstereesi vastu meetmete rakendamine aitab vähendada alarme, mis muudavad lühikese aja jooksul oma staatust[82, 83, 84].

Eri keskkondade (test, *prelive* jm) alarmid peavad olema kuvatud üksteisest eraldi, näiteks erinevatel näidikpaneelidel. See aitab kiiremini mõista probleemi kriitilisust, mille tulemina saab seada konkreetseid prioriteete. Alarmi koostamisel peab silmas pidama järgnevaid küsimusi[75]:

- Kas alarm on tingitud ebatavalisest olukorrast süsteemis? Oluline on, et alarm ei kajastaks süsteemi tavapärast käitumist.
- Kas alarmiga kaasas olev teave annab selge ülevaate toimuvast? Süsteemi-administraator peab saama info põhjal oletada alarmi tekke põhjuse.
- Kas alarmi põhjus võib mõjutada teenust või kliente? Alarm peab olema suunatud olulistele sündmustele, mis võivad mõjutada ettevõtte põhitegevust või kliendi-rahulolu.

Kui ühele küsimusele on vastus eitav, siis peaks alarmi vajalikkust uuesti hindama ja vajadusel seadma sellele madalam prioriteet. Alarmis olev info peab omakorda vastama minimaalselt küsimustele "Mis?, Kus?, Millal?":

- "Mis?" ehk milline ressurss häire püstitas.
- "Kus?" ehk millises süsteemis või teenuses ressurss asub.
- "Millal?" ehk ajahetk, mil alarm aktiveerus.

Need küsimused aitavad piirata alarmide hulka seiresüsteemis ja vähendada juurpõhjuse otsimise aega.

7.3 Tulemused

Autori loodud standardiseerimisjuhendit kasutati uue Zabbixi süsteemi juurutamisel. Seadmed jaotati gruppidesse ning nendele kohaldati mõõdikukogumikud (nähtavad Lisa 2). Alarmide seadistamisel pööratati tähelepanu üleliigsete alarmide vähendamisele, võttes arvesse hüstereesi ja rakendades sellele vastumeetmeid.

Tulemuse hindamiseks võrreldi vanas ja uues Zabbixi süsteemis nelja võrgulüüsi kuue kuu vältel. Võrgulüüsid olid süsteemide vahel samad ning nende puhul jälgiti samu parameetreid. Alarmide kogus, mis tekkis süsteemides, on kuvatud tabelis 2:

Tabel 2. *Alarmide kogus Zabbixi süsteemides*

Tase	Vana	Uus
Warning	38	11
High	1	3
Kokku	39	14

Vana süsteemi alarmide koostamisel ei olnud peatükis 7.2 olevaid alarmide sätestamist puudutavaid küsimusi ja soovitusi arvesse võetud. Sellest tulenevalt esines seal ka suurem kogus hoiatusalarme. Probleem seisnes peamiselt selles, et puudusid hüstereesivastased meetmed, mis oleksid takistanud alarmide korduvat aktiveerimist. See viis olukorrani, kus lühikese aja jooksul tekkis suur hulk sarnase sisuga alarme. Vanas süsteemis leidus ka valesti ülesseatud alarme, mistõttu ei aktiveerunud mõned kõrge tasemega alarmid. Selle (kitsa) võrdluse alusel saab väita, et uues süsteemis langes alarmide esinemissagedus ligi 65%.

Uue seiresüsteemi kasutuselevõtuga on ettevõttes paranenud seireelementide kaetavus. Olemasolevad seiregrupid ja mõõdikukogumikud võimaldavad seiresüsteemi lisada elemente kiiremini ning teha muudatusi seireprotsessis sujuvamalt.

8. Edasised tegevused

Lõputöö keskendus ettevõtte IT-taristu seire standardiseerimisele ja ühtsustamisele, kuid avas ukse mitmetele edasistele uurimissuundadele. Eesolevas peatükis vaatleb autor võimalusi analüüsi laiendamiseks ja täiendamiseks, pakkudes välja kuus teemat, mis jäid töö ulatusest välja:

1. **IT-turbe standardisatsiooni uurimine** IT-turbe standardisatsiooni analüüsi puhul on võimalik uurida erinevaid turberaamistikke, näiteks ISO27000, COBIT jm. Samuti saab analüüsida tööriistu, mis on mõeldud IT-turbe seireks ning pakkuda välja ettevõttele sobilik lahendus.
2. **Seiremetoodikate uurimine ja hindamine**
   Seiremetoodikaid antud töö ei hinnanud, vaid autor tõi välja kaks enimlevinud metoodikat, mis keskenduvad IT-taristu seires eri eesmärkidele. Edasine uuring võiks hõlmata seiremetoodikate rühmitamist ning nende põhjalikku hindamist.
3. **Alarmide üleküllus**
   Standardisatsiooni koostamisel selgus, et alarmide üleküllust on uuritud eelkõige meditsiinivaldkonnas. Kuigi sellest valdkonnast saadud teadmised on väärtuslikud, oleks IT-seiresüsteemi disainimisel vaja uurida probleemi spetsiifiliselt IT-seire vaatepunktist. See aitaks seiresüsteemide arendajatel lisada rohkem funktsioone, mis on mõeldud just valepositiivsete alarmide vähendamiseks.
4. **Standardisatsioonijuhendi põhjalikum testimine**
   Standardisatsioonijuhendi testimiseks läbi viidud võrdlus põhines viie võrgulüüsi jälgimisel, mis on liiga väike arv, et teha kindlaid järeldusi. Tulevikus peaks testimist laiendama, kaasates rohkem seadmetüüpe, näiteks servereid, ruutereid ja tulemüüre.
5. **Tehisintellekti rakendamine ettevõtte töövoo automatiseerimiseks**
   Tehisintellekti kasutuselevõtt ettevõtetes on kasvav suundumus, seda rakendadakse erinevate ülesannete lahendamisel, näiteks juturobotid. Tehisintellektist võiks ka abi olla seires, nimelt alarmi lugemisel ja selle põhjuse lahendamisel. Tehisintellekti kasutamine selles kontekstis võib oluliselt suurendada efektiivsust ja vähendada vajadust operaatori sekkumiseks.
6. **Majanduslik kasu**
   Majandusliku kasu teada saamiseks, oleks vaja arvutada ettevõtte keskmine intsidendi maksuvus ning mõõta intsidentide reageerimisaega uues seirelahenduses. Sellega saaks kindlaks teha kas reageerimisaeg on uue lahenduse tõttu paranenud ja kui suurt mõju see majanduslikult omab.

9. Kokkuvõte

Käesoleva lõputöö eesmärk oli standardiseerida ja ühtsustada ettevõtte IT-taristu seiret, selle raames koostati seire standardisatsiooni juhis ning valiti ettevõttele välja seiresüsteem, millega olemasolevaid süsteeme vähendada. Näitena kasutati rahvusvahelist logistika-ettevõtet AS Eesti Post

Töö käigus anti ülevaade ettevõttest AS Eesti Post - millega nad tegelevad ning mis süsteeme jälgivad. Lisaks tutvustati üldist seire tausta; leitavaid seiretüüpe, andmetüüpe, andmeedastusmeetodeid ja jälgimismetoodikaid.

Seire põhitüüpe on kaks: proaktiivne ja reaktiivne. Proaktiivne seire on osa talitluspidevuse planeerimisest, mille eesmärk on vältida ootamatuid tõrkeid süsteemis. Reaktiivse seire korral lahendatakse probleemid pärast nende tekkimist.

Jälgitavatelt seadmetelt andmete kättesaamiseks on kaks põhilist andmeedastusmeetodit, *push* ja *pull*. *Push*-meetodi kasutamiseks peab jälgitavates süsteemides olema agent, mis saadab perioodiliselt andmeid serverile. *Pull*-meetodi kasutamiseks on vaja rakendust, mis seiresüsteemile vajalikud andmed kättesaadavaks teeb. Seejärel pärib seiresüsteem seadmelt andmeid ise.

Metoodikatüüpe, mis pakuvad erinevaid lahendusviise süsteemide jälgimiseks, on mitu. Üldlevinud on *USE* ja neli kuldset signaali. *USE* metoodika on mõeldud kasutamiseks süsteemi jõudluse uurimisel. Neli kuldset signaali keskendub teenuse kvaliteedi mõõtmisele.

Ettevõtte probleem seire rakendamisel tuleneb standardi puudumisest ja seiresüsteemide üleküllusest. Seejuures on antud probleem globaalselt levinud. 2020. aastal läbi viidud uuring leidis, et seire efektiivset rakendamist raskendavad needsamad asjaolud.

Ettevõtte seiresüsteemidele sai tehtud põhjalik analüüs. Analüüsitavaid süsteeme oli neli: Nagios Core, Cacti, Prometheus ja Zabbix. Analüüs hõlmas süsteemide konfigureerimise, haldamise, automatiseerimise, andmete visualiseerimise ja talletamise ning dokumentatsiooni ja toe vaatlemist. Analüüsi tulemusi hinnati ning selgitati välja, milliste süsteemide kasutamist peaks ettevõttes jätkama ning millised vajaks asendamist.

Seire standardiseerimise eesmärk on tagada, et seadmeid jälgitaks järjepidevalt ja ühtlus-

tatud viisil. See tähendab, et olenemata seiresüsteemist käib andmete kogumine kindlate kriteeriumide ja piirmäärade järgi ning mõõdikute ja alarmide konfiguratsioon on ühtne. Seire standardisatsiooni juhendis on kirjeldatud seiregruppide ning mõõdikukogumike eesmärk ning näide nende koostamisest. Alarmeerimise osas on välja toodud soovitusi alarmis sisalduva teabe täiendamiseks ning alarmi ülekülluse vähendamiseks.

Lõputöö täiendamiseks tõi autor välja kuus teemat, mis jäid töö ulatusest välja: IT-turbe standardisatsiooni uurimine, mille puhul saab uurida erinevaid turberaamistikke ja turbele suunatud seiresüsteeme; seiremetoodikate uurimine ja hindamine, mille käigus saab uurida erinevaid seiremetoodikaid, neid rühmitada ning hinnata; alarmide ülekülluse uurimine IT-seire kontekstis, mis aitaks seiresüsteemide arendajatel lisada rohkem funktsioone valepositiivsete alarmide vähendamiseks; standardisatsiooni juhendi põhjalikum testimine, kaasates sinna rohkem seadmetüüpe, näiteks servereid, tulemüüre jm; tehisintellekti rakendamine ettevõtte töövoo automatiseerimiseks, millega oleks võimalik alarmi põhjust automaatselt lahendada; majandusliku kasu uurimine, arvutades välja ettevõtte keskmise instidendi maksuvuse

Kokkuvõtlikult on lõputöö eesmärk saavutatud: koostatud sai praktiline standardisatsiooni juhend ning esitati konkreetseid lahendusi parema seire saavutamiseks ning valitud süsteemide tõhusaks kasutamiseks. ASis Eesti Post võeti kasutusele autori soovitatud seiresüsteem Zabbix. Järgides selle juurutamisel standardisatsioonijuhendit on alarmide sagedus vähenenud 65%. Uue süsteemi kasutuselevõtuga on suurenenud seireelementide kaetavus. Süsteemi aitab efektiivsemalt hallata standardisatsioonijuhendi järgi loodud seiregrupid ja mõõdikukogumikud.

Kasutatud kirjandus

[1] Sihyung Lee, Kyriaki Levanti, and Hyong S. Kim. *Network monitoring: Present and future*. [Kasutatud: 30.09.2023]. URL: https://www.sciencedirect.com/science/article/abs/pii/S138912861400111X.

[2] AS Eesti Post. *Lahendused e-kaupmehele*. [Kasutatud: 11.04.2023]. URL: https://www.omniva.ee/ari/pakk/internetimuuk.

[3] Leanne M Kelly. *Baselines and monitoring: More than a means to measure the end*. [Kasutatud: 23.04.2023]. URL: https://journals.sagepub.com/doi/10.1177/1035719X20977522.

[4] Stanislava Šimonová and Ondřej Šprync. *Proactive IT / IS monitoring for business continuity planning*. [Kasutatud: 18.09.2023]. URL: https://otik.uk.zcu.cz/handle/11025/17404.

[5] Antoine Trad, Damir Kalpic, and Kresimir Fertalj. *Proactive Monitoring of the Information System Risk and Quality*. [Kasutatud: 11.09.2023]. URL: https://ieeexplore.ieee.org/abstract/document/1024687.

[6] Mark Dilman and Danny Raz. *Efficient Reactive Monitoring*. [Kasutatud: 11.09.2023]. URL: https://ieeexplore.ieee.org/abstract/document/1003034.

[7] Wei Xu et al. *Detecting Large-Scale System Problems by Mining Console Logs*. [Kasutatud: 09.05.2023]. URL: https://dl.acm.org/doi/10.1145/1629575.1629587.

[8] P. K. Sahoo et al. *Syslog a Promising Solution to Log Management*. [Kasutatud: 20.09.2023]. URL: https://www.researchgate.net/publication/332780312_Syslog_a_Promising_Solution_to_Log_Management.

[9] Muhammad Naeem Ahmed Khan. *Multi-agent Based Forensic Analysis Framework for Infrastructures Involving Storage Networks*. [Kasutatud: 20.09.2023]. URL: https://link.springer.com/article/10.1007/s40010-017-0473-3.

[10] R. Gerhards. *RFC 5424: The Syslog Protocol*. [Kasutatud: 20.09.2023]. URL: https://dl.acm.org/doi/abs/10.17487/RFC5424.

[11] Dileepa Jayathilake. *Towards structured log analysis*. [Kasutatud: 10.05.2023]. URL: https://ieeexplore.ieee.org/abstract/document/6261962.

[12] Lisa Ehrlinger and Wolfram Wöß. *A Survey of Data Quality Measurement and Monitoring Tools*. [Kasutatud: 20.05.2023]. URL: https://www.researchgate.net/publication/359636546_A_Survey_of_Data_Quality_Measurement_and_Monitoring_Tools.

[13] Steven A Melnyk, Douglas M Stewart, and Morgan Swink. *Metrics and performance measurement in operations management: dealing with the metrics maze*. [Kasutatud: 21.09.2023]. URL: https://www.sciencedirect.com/science/article/pii/S0272696304000105.

[14] J.P Martin-Flatin. *Push vs. pull in Web-based network management*. [Kasutatud: 27.05.2023]. URL: https://ieeexplore.ieee.org/abstract/document/770671.

[15] Ales Komarek et al. *Metric Based Cloud Infrastructure Monitoring*. [Kasutatud: 11.05.2023]. URL: https://link.springer.com/chapter/10.1007/978-3-319-69835-9_37.

[16] Mingwei Lin et al. *A hybrid push protocol for resource monitoring in cloud computing platforms*. [Kasutatud: 24.09.2023]. URL: https://www.sciencedirect.com/science/article/abs/pii/S0030402615017271.

[17] M. Bhide et al. *Adaptive push-pull: disseminating dynamic Web data*. [Kasutatud: 26.09.2023]. URL: https://ieeexplore.ieee.org/abstract/document/1009150.

[18] Richard Veryard. *Quantification of alarm chatter based on run length distributions*. [Kasutatud: 17.10.2023]. URL: https://www.sciencedirect.com/science/article/abs/pii/0011684X85902618.

[19] Brendan Gregg. *Thinking Methodically about Performance: The USE method addresses shortcomings in other commonly used methodologies.* [Kasutatud: 10.06.2023]. URL: https://dl.acm.org/doi/10.1145/2405116.2413037.

[20] Rob Ewaschuk. "Monitoring Distributed Systems". In: *Site Reliability Engineering: How Google Runs Production Systems*. 2016.

[21] Marco Miglierina Damian A. Tamburri and Elisabetta Di Nitto. *Cloud applications monitoring: An industrial study*. [Kasutatud: 11.05.2023]. URL: https://www.sciencedirect.com/science/article/pii/S0950584920301452.

[22] M. SAYAGH et al. *Software Configuration Engineering in Practice Interviews, Survey, and Systematic Literature Review*. [Kasutatud: 12.08.2023]. URL: https://ieeexplore.ieee.org/document/8451922.

[23] Kiran Nagaraja et al. *Understanding and Dealing with Operator Mistakes in Internet Services*. [Kasutatud: 03.10.2023]. URL: https://www.usenix.org/conference/osdi-04/understanding-and-dealing-operator-mistakes-internet-services.

[24] David Oppenheimer, Archana Ganapathi, and David A. Patterson. *Why Do Internet Services Fail, and What Can Be Done About It?* [Kasutatud: 03.10.2023]. URL: https://www.usenix.org/conference/osdi-04/understanding-and-dealing-operator-mistakes-internet-services.

[25] Tianyin Xu, Vineet Pandey, and Scott Klemmer. *An HCI View of Configuration Problems*. [Kasutatud: 05.10.2023]. URL: https://arxiv.org/abs/1601.01747.

[26] David Both. "Automate Everything". In: *The Linux Philosophy for SysAdmins*. 2018.

[27] Geoff Halprin. *The Work Flow of System Administration*. [Kasutatud: 03.10.2023]. URL: https://citeseerx.ist.psu.edu/doc_view/pid/0390b99c6276ea8e2957dfe54b747e71d8f0d242.

[28] Æleen Frisch. *Essential System Administration: Tools and Techniques for Linux and Unix Administration*. O'Reilly Media, Inc., 2002.

[29] Antony Unwin. *Why is Data Visualization Important? What is Important in Data Visualization?* [Kasutatud: 21.10.2023]. URL: https://assets.pubpub.org/5igceo8e/d0ffd9c5-f57e-4643-b4d5-9599c395ac63.pdf.

[30] Gholamreza Torkzadeh and William J. Doll. *The place and value of documentation in end-user computing*. [Kasutatud: 14.08.2023]. URL: https://www.sciencedirect.com/science/article/abs/pii/037872069390063Y?via%3Dihub.

[31] Johannes Sametinger. "Software Documentation". In: *Software Engineering with Reusable Components*. 1997.

[32] Gias Uddin and Martin P. Robillard. *How API Documentation Fails*. [Kasutatud: 08.04.2023]. URL: https://ieeexplore.ieee.org/abstract/document/7140676.

[33] Nagios Enterprises. *History of Nagios*. [Kasutatud: 12.03.2023]. URL: https://www.nagios.org/about/history/.

[34] Nagios Enterprises. *Monitoring Publicly Available Services*. [Kasutatud: 12.03.2023]. URL: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/monitoring-publicservices.html.

[35] Dinangkur Kundu and S. M. Ibrahim Lavlu. *Cacti 0.8 Network Monitoring*. Packt Publishing Ltd, 2009.

[36] The Cacti Group Inc. *Features*. [Kasutatud 18.03.2023]. URL: https://www.cacti.net/info/features.

[37] James Turnbull. *Monitoring With Prometheus*. Turnbull Press, 2018.

[38] Zabbix SIA. *Table of Contents*. [Kasutatud: 08.04.2023]. URL: https://www.zabbix.com/documentation/current/en/manual/introduction/about.

[39] Gorka Gallardo Josune Hernantes and Nicolás Serrano. *IT Infrastructure-Monitoring Tools*. [Kasutatud: 31.03.2023]. URL: https://ieeexplore.ieee.org/document/7140697.

[40] Syed Ali. "Monitoring with Nagios and Trend Analysis with Cacti". In: *Practical Linux Infrastructure*. 2015.

[41] Nagios Enterprises. *Verifying Your Configuration*. [Kasutatud: 28.10.2023]. URL: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/verifyconfig.html.

[42] Nagios Enterprises. *Time-Saving Tricks For Object Definitions*. [Kasutatud: 05.10.2023]. URL: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/objecttricks.html.

[43] Sous Chefs. *nagios cookbook*. [Kasutatud: 04.09.2023]. URL: https://github.com/sous-chefs/nagios.

[44] Nagios Enterprises. *Notifications*. [Kasutatud: 23.10.2023]. URL: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/notifications.html.

[45] Nagios Enterprises. *Category: Notifications*. [Kasutatud: 26.10.2023]. URL: https://exchange.nagios.org/directory/Plugins/Notifications.

[46] Nagios Enterprises. *Service Checks*. [Kasutatud: 01.11.2023]. URL: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/servicechecks.html.

[47] Nagios Enterprises. *Visualizations Articles*. [Kasutatud: 23.10.2023]. URL: https://support.nagios.com/kb/category.php?id=153.

[48] Ethan Galstad. *NDOUTILS Documentation Version 2x*. [Kasutatud: 24.10.2023]. URL: https://assets.nagios.com/downloads/nagioscore/docs/ndoutils/NDOUtils.pdf.

[49] Nagios Enterprises. *Table of Contents*. [Kasutatud: 08.04.2023]. URL: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/toc.html.

[50] Nagios Enterprises. *Nagios Core Addons*. [Kasutatud: 08.04.2023]. URL: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/addons.html#nrpe.

[51] Nagios Enterprises. *NRPE Github commits*. [Kasutatud: 08.04.2023]. URL: https://github.com/NagiosEnterprises/nrpe/commit/e5cdd3b641b36ca4f56b1c5ca7286dafb85ab56f.

[52] Nagios Enterprises. *Nagios Core 4.x Version History*. [Kasutatud: 05.10.2023]. URL: https://www.nagios.org/projects/nagios-core/4x/.

[53] Nagios Enterprises. *Nagios Product Roadmaps*. [Kasutatud: 12.04.2023]. URL: https://www.nagios.com/roadmaps/.

[54] Nagios Enterprises. *Arhiiv: Nagios Product Roadmaps*. [Kasutatud: 12.04.2023]. URL: https://web.archive.org/web/20200810113647/https://www.nagios.com/roadmaps/.

[55] The Cacti Group Inc. *Cacti Documentation*. [Kasutatud: 01.11.2023]. URL: https://docs.cacti.net/.

[56] The Cacti Group Inc. *thold*. [Kasutatud: 23.10.2023]. URL: https://github.com/Cacti/plugin_thold.

[57] Denis P. Rudd II. *The value of video in online instruction*. [Kasutatud: 20.08.2023]. URL: https://eddl.tru.ca/wp-content/uploads/2019/08/EDDL5101_W5_Rudd_and_Rudd_2014.pdf.

[58] José Couto Marques et al. *The use of video clips in engineering education*. [Kasutatud: 20.08.2023]. URL: https://www.researchgate.net/publication/261164909_The_use_of_video_clips_in_engineering_education.

[59] The Cacti Group Inc. *Changelog*. [Kasutatud: 05.10.2023]. URL: https://www.cacti.net/info/changelog.

[60] The Cacti Group Inc. *Roadmap*. [Kasutatud: 20.08.2023]. URL: https://web.archive.org/web/20170602090745/http://www.cacti.net/roadmap.php.

[61] Mainak Chakraborty and Ajit Pratap Kundan. “Chapter 6: Grafana”. In: *Monitoring Cloud-Native Applications*. Apress, 2021.

[62] Grafana Labs. *Grafana OnCall OSS*. [Kasutatud: 02.11.2023]. URL: https://grafana.com/oss/oncall/.

[63] Prometheus Authors. *Remote Endpoints and Storage*. [Kasutatud: 23.10.2023]. URL: https://prometheus.io/docs/operating/integrations/#remote-endpoints-and-storage.

[64] VictoriaMetrics. *Multiple retentions*. [Kasutatud: 23.10.2023]. URL: https://docs.victoriametrics.com/Single-server-VictoriaMetrics.html#retention.

[65] Prometheus Authors. *Overview*. [Kasutatud: 01.11.2023]. URL: https://prometheus.io/docs/introduction/overview/.

[66] Prometheus Authors. *Long Term Support*. [Kasutatud: 17.10.2023]. URL: https://prometheus.io/docs/introduction/release-cycle/.

[67] Prometheus Authors. *Roadmap*. [Kasutatud: 08.09.2023]. URL: https://prometheus.io/docs/introduction/roadmap/.

[68] Prometheus Authors. *Prometheus Roadmap Github*. [Kasutatud: 08.09.2023]. URL: https://github.com/prometheus/docs/blob/main/content/docs/introduction/roadmap.md.

[69] Nathan Liefting and Brian van Baekel. *Zabbix 6 IT Infrastructure Monitoring Cookbook*. Packt, 2022.

[70] Vox Pupuli. *puppet-zabbix*. [Kasutatud: 08.10.2023]. URL: https://forge.puppet.com/modules/puppet/zabbix/readme.

[71] Zabbix SIA. *Zabbix Ansible Collection*. [Kasutatud: 08.10.2023]. URL: https://github.com/zabbix/ansible-collection.

[72] Zabbix SIA. *Zabbix documentation*. [Kasutatud: 01.11.2023]. URL: https://www.zabbix.com/documentation/current/en/.

[73] Zabbix SIA. *Zabbix Life Cycle Release Policy*. [Kasutatud: 09.10.2023]. URL: https://www.zabbix.com/life_cycle_and_release_policy.

[74] Zabbix SIA. *Roadmap*. [Kasutatud: 08.10.2023]. URL: https://www.zabbix.com/roadmap.

[75] Stefan Wallin. *Chasing a Definition of "Alarm"*. [Kasutatud: 09.04.2023]. URL: https://link.springer.com/article/10.1007/s10922-009-9127-3.

[76] Stefan Wallin et al. *The semantics of alarm definitions: enabling systematic reasoning about alarms*. [Kasutatud: 11.10.2023]. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.800.

[77] Guofei Jiang et al. *Ranking the Importance of Alerts for Problem Determination in Large Computer Systems*. [Kasutatud: 11.10.2023]. URL: https://dl.acm.org/doi/abs/10.1145/1555228.1555232.

[78] Maria Cvach. *Monitor Alarm Fatigue: An Integrative Review*. [Kasutatud: 11.10.2023]. URL: https://array.aami.org/doi/full/10.2345/0899-8205-46.4.268.

[79] Azizeh Khaled Sowan et al. *Nurses' Perceptions and Practices Toward Clinical Alarms in a Transplant Cardiac Intensive Care Unit: Exploring Key Issues Leading to Alarm Fatigue*. [Kasutatud: 13.10.2023]. URL: https://humanfactors.jmir.org/2015/1/e3/.

[80] Siobhán Casey, Gloria Avalos, and Maura Dowling. *Critical care nurses' knowledge of alarm fatigue and practices towards alarms: A multicentre study. Intensive Critical Care Nursing*. [Kasutatud: 13.10.2023]. URL: https://doi.org/10.1016/j.iccn.2018.05.004.

[81] Bradford D. Winters et al. *Technologic Distractions (Part 2): A Summary of Approaches to Manage Clinical Alarms With Intent to Reduce Alarm Fatigue*. [Kasutatud: 13.10.2023]. URL: https://doi.org/10.1016/j.iccn.2018.05.004.

[82] Jens Folmer and Birgit Vogel-Heuser. *Computing dependent industrial alarms for alarm flood reduction*. [Kasutatud: 15.10.2023]. URL: https://ieeexplore.ieee.org/abstract/document/6198008.

[83] Alan J. Hugo. *Estimation of Alarm Deadbands*. [Kasutatud: 15.10.2023]. URL: https://www.sciencedirect.com/science/article/pii/S1474667016358530.

[84] Sandeep R. Kondaveeti et al. *Quantification of alarm chatter based on run length distributions*. [Kasutatud: 15.10.2023]. URL: https://www.sciencedirect.com/science/article/abs/pii/S0263876213001779.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

Mina, Markus Kuuse

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Seirelahenduste standardiseerimine ja ühtsustamine AS Eesti Post näitel", mille juhendaja on Edmund Laugasson and Martin Rajur
   1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
   1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

04.01.2024

[1] Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 - Zabbixis olevad seadmegrupid ja mõõdikukogumikud

Zabbixis on 4 seadmetüüpi ja 2 rakendust Seadmed:

- Võrgulüüsid
- Ruuterid
- Tulemüürid
- MS Windows serverid

Rakendused:

- Andmebaasid
- Veebilehed

Algne seadmegrupp ja mõõdikukogumik - kuhu kuuluvad kõik seadmed "Taristu seadmed"

- ICMP ping
- Võrgukaardi läbilaske maht
- Võrgukaardi poolt tulnud vead
- Protsessori kasutus
- Vahemälu kasutus

"Võrguseadmed" - Siia kuuluvad kõik võrgulüüsid, ruuterid ja tulemüürid

- SNMP toimivus

"Tulemüürid" - Siia kuuluvad Tulemüürid

- Aktiivsed sessioonid

"MS Windows serverid" - Siia kuuluvad kõik Windows serverid

- Agendi toimivus
- Kogu kettamaht
- Vaba kettamaht

"Andmebaasid" - Siia kuuluvad kõik andmebaasid

- Aeg, mil toimus viimane varundamine
- Jooksev päringute arv
- Luku ooteaeg

"Veebilehed" - Siia kuuluvad veebilehed

- HTTPSi sertifikaadi aegumine
- Veebilehe kättesaadavus üle HTTPSi