TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Karl Rikkonen 206101IAAB

Automatiseeritud intsidentide reageerimisplatvormi juurutamine turbeoperatsioonide keskuses TheHive 5 näitel

Bakalaureusetöö

Juhendaja: Siim Vene
MSc
Kaasjuhendaja: Karl Mendelman
MSc

Tallinn 2023

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: Karl Rikkonen

10.05.2023

Annotatsioon

Käesolev bakalaureusetöö keskendub automatiseeritud turbeintsidentide reageerimisplatvormi (IRP) juurutamisele organisatsiooni turbeoperatsioonide keskuses (SOC). Töö eesmärgiks on asendada aegunud ja ebaefektiivne piletihaldustarkvara organisatsioonile sobivama lahendusega, mis vastab turbeoperatsioonide keskuse juhtkonna poolt sätestatud kriteeriumitele.

Lõputöös tutvustatakse küberturbe valdkonna teoreetilisi alused, pakkudes ülevaadet turbeoperatsioonide keskusest, turbeintsidentide reageerimise protsessist, intsidentide reageerimisplatvormidest ja SOAR (*Security Orchestration, Automation, and Response*) süsteemidest. Töös analüüsitakse erinevaid automatiseeritud intsidendihaldus- ja reageerimisplatvorme ning SOAR süsteeme, mille tulemusena valitakse välja sobiv lahendus organisatsiooni turbeoperatsioonide keskusele. Töö käsitleb valitava platvormi jaoks olulisi integratsioonivõimalusi ja analüüsib ning võrdleb süsteemihaldustööriistu ja -meetodeid platvormi juurutamiseks organisatsioonis.

Praktilises osas luuakse konteineriseeritud platvorm ning virtuaalne testkeskkond. Lisaks rakendatakse projektile automaatne tarneahel, tagamaks platvormile püsiv integreerimis-, juurutus- ning arendusvõimalus asutuse privaatpilves. Praktilise osa lõppfaasis viiakse sisse esmased integratsioonid turbetööriistadega. Töö tulemused aitavad parandada turbeintsidentidele reageerimisprotsessi, suurendada organisatsiooni olukorrateadlikust küberruumis ning tõhustada turbemeeskonna tööd.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 69 leheküljel, 4 peatükki, 5 joonist, 3 tabelit.

Abstract

Deploying an Automated Incident Response Platform in a Security Operations Center (SOC) Using TheHive 5 as an Example

This bachelor's thesis focuses on implementing an automated incident response platform (IRP) in an organization's Security Operations Center (SOC). The study aims to replace outdated and inefficient ticket management software with a more suitable solution that meets the criteria set by SOC management.

The thesis introduces the theoretical foundations of cybersecurity operations, providing an overview of the Security Operations Center, incident response process, incident response platforms, and SOAR (Security Orchestration, Automation, and Response) systems. The study analyzes automated incident response platforms, and SOAR systems, selecting an appropriate solution for the organization's SOC. The work addresses significant integration possibilities for the chosen platform and analyzes and compares system management tools and methods for implementing the platform in the organization.

The author created a containerized platform and a virtual test environment in the practical part of the thesis. In addition, an automatic delivery pipeline is implemented for the project, ensuring a continuous integration and deployment (CI/CD) capability for the platform within the organization's private cloud. In the final phase of the practical part, the author introduces initial integrations with security tools for the chosen platform. The results of the study help improve the incident response process, enhance the organization's situational awareness in cyberspace, and streamline the security team's work.

The thesis is written in Estonian and contains 69 pages of text, 4 chapters, 5 figures, 3 tables.

Lühendite ja mõistete sõnastik

Alpine Linux	Turvaline ja väikse jalajäljega *Linuxi* distributsioon
Ansible	Tööriistade komplekt kirjeldamaks infrastruktuuri koodina
Ansible Playbook	*Ansible* tööks vajalik skripti fail, *YAML*-i süntaksiga. Ansible mänguraamat kirjeldamaks taristut koodina
Ansible Roles	*Ansible* taaskasutatavad, modulaarsed konfiguratsioonide kogumid
Ansible Vault	*Ansible* salafraaside ja konfidentsiaalsete andmete krüpteerimise tööriist
API	*Application Programing Interface*, rakenduste vaheline suhtlusliides
AWS	*Amazon Web Services*, Amazoni poolt pakutavad pilvepõhised teenused
Bash	*Bourne-Again Shell*, *Linux* (*Unix*) kest ja käsukeel
CERT	*Computer Emergency Response Team*, arvutiavariide tõrje rühm
CERT Société Générale	Prantsusmaa panga arvutiavariide tõrje rühm
CI/CD	*Continuous Integration, Continuous Delivery/Deployment*, järjepidev integratsioon ja juurutus
Cortex	Automaatanalüüse teostav mootor (platvorm, *TheHive-Project*)
Cortex Neurons	*TheHive Cortex* laiendused - analüsaatorid ja reageerijad
CSIRT	*Computer Security Incident Response Team*, arvutiturbeintsidentidele reageerimise rühm
CSP	*Content Security Policy*, (veebilehtede) sisu turbepoliitika
Cuckoo (Sandbox)	Automaatne pahavaraanalüüsi liivakast
DevOps	*Development and Operations*, Arendusfilosoofia - arendus ja käitus
DNS	*Domain Name System*, domeeninimede süsteem
Docker	Konteinerite virtualiseerimise platvorm
Docker Compose	*Docker*-i konteinerite orkestreerimise tööriist *YAML*-i abil
Docker Compose File	*Docker*-i konteinerite kirjeldus fail *YAML*-i süntaksiga
Docker Stack	*Docker*-i teenuste virna haldustööriist
Docker Swarm	*Docker*-i klastrite ja konteinerite orkestreerimise tööriist

DomainTools	Veebiteenus domeeniinfo pärimiseks
EDR	*Endpoint Detection and Response*, lõppseadme pahavaratuvastuse ja reageerimise süsteem
ENISA	*European Union Agency for Cybersecurity*, Euroopa Liidu võrgu-ja infoturbe agentuur
EVS-ISO	Eesti Standardimis- ja Akrediteerimiskeskus
ext4	Linuxi failisüsteem
FP	*False positive*, väärtuvastus, väärpositiivsus
FQDN	*Fully Qualified Domain Name*, täielik domeeninimi
Freemium	Turundusstrateegia ja sellele põhinev ärimudel pakkumaks tarkvara nii tasuta põhiversioonina kui ka tasulisi lisafunktsioone
Git	Versioonihaldussüsteem
HTTP	*Hypertext Transfer Protocol*, hüperteksti edastusprotokoll
HTTPS	*Hypertext Transfer Protocol Secure*, turvaline hüperteksti edastusprotokoll
IaC	*Infrastructure as Code*, taristu koodina
IoC	*Indicator of Compromise*, turvarikkemärk või ohuindikaator
IPS	*Intrusion Prevention System*, sissetungitõrje süsteem
IRM	*Incident response methodology*, intsidentidele reageerimise metoodika
IRP	*Incident Response Platform*, intsidentide reageerimise platvorm
ISIRT	*Information Security Incident Response Team*, infoturbeintsidentide tõrje rühm
iTop	*IT Operations Portal*, IT-teenuste haldamise (ITSM) tööriist
ITSM	*IT Service Management*, infotehnoloogiateenuste haldus
JA3 (Fingerprint)	*SSL/TLS* sertifikaadi digitaalnesõrmejälg
Jenkins	Järjepideva integratsiooni ja juurutamise tööriist (tarneahelate loomise tööriist)
Jenkinsfile	*Jenkins*-i töökirjelduse skripti fail
KPI	*Key Performance Indicator*, keskne soorituse indikaator/mõõdik
Kubernetes	Konteinerite orkestreerimise platvorm
LVM	*Logical Volume Manager*, loogiliste kettajagude ja mäluseadmete haldamise tarkvara, *Linux* operatsioonisüsteemis
MISP	*Malware Information Sharing Platform*, ohuteabe jagamise platvorm
NFS	*Network File System*, võrgufailisüsteem (protokollistik)

NIPS	*Network Intrusion Prevention System*, võrgupõhine *IPS*
NIST	*National Institute of Standards and Technology*, USA Riiklik Standardi- ja Tehnikainstituut
NTP	*Network Time Protocol*, võrguaja protokoll, arvutikellade sünkroniseerimiseks
OWASP ZAP	*Open Web Application Security Project Zed Attack Proxy*, nõrkuste otsimise tööriist
Playbook	*Ansible* tööks vajalik skripti fail, *YAML*-i süntaksiga. *Ansible* mänguraamat kirjeldamaks taristut koodina
Powershell	*Microsoft*-i poolt loodud skriptimiskeel
PPT	*People, Process, Technology*, inimesi, protsesse ja tehnoloogid siduv raamistik
Python	Kõrgtaseme programmerimiskeel
S3 (AWS S3)	*Amazon Simple Storage Service*, Amazoni lihtne andmete salvestamise teenus. Objektsalvestusteenus, *Amazon S3* või teenus, mis pakub sarnast funktsionaalsust
SANS	*SysAdmin, Audit, Network and Security Institute*, Juhtiv infoturbekoolituste ja -sertifitseerimise instituut
SIEM	*Security Information and Event Management*, turbeteabe ja -sündmuste haldus (süsteem)
SNMP	*Simple Network Management Protocol*, lihtne võrguhalduse protokoll
SOAR	*Security Orchestration, Automation, and Response*, turbe orkestreerimine, automatiseerimine ja reageerimine (süsteem, platvorm, strateegia)
SOC	*Security Operations Center*, (küber) turbeoperatsioonide keskus, lühidalt turbekeskus
SOCaaS	*Security Operations Center as a Service*, *SOC* teenusena
SSH	*Secure Shell*, turvaline kest ehk turvaline (krüpteeritud) kaugpöördus protokoll
SSL	*Secure Sockets Layer*, turvasoklikihi protokoll, *TLS* eelkäija
TCP/IP	*Transmission Control Protocol/Internet Protocol*, võrguprotokollide perekond (edastusohje protokoll ja interneti protokoll)
TLP	*Traffic Light Protocol*, valgusfooriprotokoll – teabe tundlikkuse määramine värvikoodide (reeglite) järgi
TLS	*Transport Layer Security*, transpordikihi krüpteerimis protokoll. *SSL*-i edasiarendus
TP	*True Positive*, tõene tuvastus

TTP	*Tactics, Techniques, and Procedures*, taktikad, tehnikad ja protseduurid – ohusubjekti käitumismustrid
Ubuntu	*Linuxi* distributsioon (*Debian*-põhine)
URL	*Uniform Resource Locator*, ühtne ressursilokaator (lihtsustatult veebiaadress)
Vagrant	*HashiCorp*-i loodud virtuaalmasinate haldus tööriist
Vagrantfile	*Vagrant*-i konfiguratsiooni fail kirjeldamaks virtuaalmasinate automatiseeritud käitamist ja haldust
VirusTotal	Teenus, mis analüüsib faile ja *URL*-e viiruste ja pahavara suhtes
XSS	*Cross-Site Scripting*, skriptisüst, murdskriptimine (ründemeetod)
X-XSS-Protection	Brauseri kaitsefunktsioon *XSS* rünnete vastu
YAML	*YAML Ain't Markup Language*, *YAML* pole märgistuskeel – inimloetav andmete jadastuse keel

Sisukord

Jooniste loetelu

Tabelite loetelu

Sissejuhatus

Tänapäeval on info- ja küberruum kiiresti muutuv keskkond, kus organisatsioonide ning ettevõtete jaoks on turvaliste infosüsteemide tagamine ja ümbritsevate ohtude kiire kaardistamine kriitilise tähtsusega. Küberturvalisus on tänapäeval suurt kõlapinda saav valdkond ning kasvav küberohtude tase ja küberkurjategijate leidlikkus tähendab, et kaasaegsed lahendused on hädavajalikud organisatsioonide kaitsmiseks küberkuritegevuse ja -rünnete eest.

Ettevõtetes ja organisatsioonides on tavaks saanud osta teenusena sisse või luua oma küberturbe operatsioonide keskus, lühidalt turbekeskus ehk SOC (*Security Operations Center*). Turbekeskus on spetsialiseerunud turbe- ja infosüsteemide monitooringule, turbeintsidentide tuvastamisele ning neile reageerimisele ja küberohtude leevendamisele. SOC kui teenus hõlmab endas mitmeid protsesse, mille järgi turbemeeskond toimetab. Üheks olulisemaks turbekeskuse protsessiks on turbeintsidentidele reageerimisprotsess, mis aitab tagada, et kõik turvarikked saavad tuvastatud ja lahendatud ning organisatsioon saab turvalisuse tagamiseks vajaliku teabe ja ressursid õigeaegselt. Kokkuvõtvalt tagab turbekeskus, et organisatsiooni infosüsteemides oleks tagatud andmete käideldavus, konfidentsiaalsus ja terviklus.

Turbeintsidentide reageerimisprotsessi toetab suuresti turbeintsidentide haldus- ja reageerimistarkvara. Tegemist on tarkvaratüübiga, mis vastutab kogu turbeintsidendi elutsükli eest, alates intsidendi loomisest või tuvastamisest, kuni intsidendi pileti sulgemise ja arhiveerimiseni. Turbeintsidentide arhiveerimine võimaldab luua retrospektiivset statistikat ning korreleerida erinevaid sündmuste atribuute, näiteks: tuvastada turbeintsidendis osalejad, milliseid tegevusi tehti, millal neid tehti ja kuidas intsident lahendati, pakkudes organisatsioonile olulist ajaloolist mälu turbeintsidentide kohta. Vältimaks inimestest tingituid vigu ning tõhustamaks intsidentide reageerimisprotsessi, kasutavad organisatsioonid üha enam automatiseerituid lahendusi, mis suudavad suurema osa reageerimisprotsessist automatiseerida. Sellised automatiseeritud intsidendihaldus ja -reageerimis platvorme nimetatakse ka SOAR

(*Security Orchestration, Automation and Response*) süsteemideks. Kaasaegsed automatiseeritud reageerimisplatvormid suurendavad turbekeskuse operatiivsust ja reageerimiskiirust.

Käesolev bakalaureusetöö keskendub automatiseeritud turbeintsidentide reageerimisplatvormi, lühendatult IRP (*Incident Response Platform*) juurutamisel organisatsiooni turbekeskuses. Uue IRP juurutamise ajendiks on asendada aegunud ja ebaefektiivne piletihaldustarkvara. Töö eesmärgiks on valida välja ning juurutada turbeintsidentide reageerimistarkvara, mis vastab SOC juhtkonna poolt sätestatud nõuetele. Põhirõhk on töös suunatud erinevatele haldustööriistade ja –meetoditele, mis aitavad IRP-d konfigureerida, üles ehitada ning juurutada. Lisaks käsitletakse erinevaid võimalusi IRP integreerimiseks teiste turbetööriistadega, saavutamaks automatiseeritud intsidentide reageerimisplatvorm ning liikuda lähemale SOAR võimekusele. Käesoleva töö skoopi ei kuulu turbeintsidentide lahendamine, platvormi sisu haldus (kasutajad, piletimallid, armatuurlauad), platvormi varunduse tagamine ja avariitaasteplaani loomine, platvormi skaleerimine ja kõrgkäideldavuse tagamine, organisatsiooni privaatpilve taristu haldus ja privaatpilves virtuaalmasinate loomine, põhjalikku platvormi veebilehe testimist, koolitus ja treening keskkonnas platvormi juurutamist.

1 Töö taust

Turbeoperatsioonikeskuse üheks peamiseks tarkvaralahenduseks on intsidendihaldus- ja reageerimisplatvorm, mis moodustab turbemeeskonna informatsioonituumiku, kuna sinna kogutakse kogu vajalik teave turbeintsidentide kohta. Intsidendihaldus- ja reageerimisplatvorm on SOC-i jaoks kriitilise tähtsusega, tagades kaitstava organisatsiooni põhjaliku olukorrateadlikkuse. Olukorrateadlikkuse loomist toetavad IRP-sse integreeritud statistilised andmed ning andmete põhjal koostatavad mõõdikud. Tagamaks SOC-i madalat reageerimisaega tuleks automatiseerida võimalikult suur osa intsidentide reageerimisprotsessist ning tagada erinevate turbetööriistade ja -tehnoloogiate sidusus IRP-ga. See võimaldab rikastada turbeintsidentidega seotud andmeid ning rakendada rikastatud andmete põhjal automaatset reageerimist.

Nagu ka sissejuhatuses mainitud, siis tihtipeale nimetatakse automatiseeritud intsidendihaldus ja reageerimisplatvorme ka SOAR-süsteemideks. SOAR-süsteemid kujutvad endast komplekssеid, mitmetest tarkvaraosisest koosnevaid platvorme eesmärgiga automatiseerida turbemeeskonna töövooge. SOAR-süsteemid suudavad vastu võtta sisendeid erinevatest turbetarkvaradest, näiteks tulemüüridest, logihaldussüsteemidest ja pahavaratõrje süsteemidest ning seejärel viia kogutud andmete põhjal läbi automaatanalüüsi, korreleerides kogutuid andmeid. Korrelatsioonide tulemusel suudavad SOAR süsteemid rakendada intsidendilahendusplaanis erinevaid automatiseerituid tegevusi, näiteks blokeerida tulemüürist lõppseadme võrguliiklus, kui see on pahavaraga nakatunud[1], [2]. NIST-i (*National Institute of Standards and Technology*) raporti IR 8434[3] ja Palo Alto Networks 2020 raporti[2] põhjal võib SOAR süsteemi kirjeldada ka kui viimastel aastatel välja arenenud küberturbestrateegia, mille aluseks on intsidendireageerimise protsessi automatiseerimine.

Käesolev lõputöö käsitleb intsidentide reageerimisplatvormi juurutusetappe organisatsiooni turbeoperatsioonide keskuses TheHive 5 platvormi näitel. Töö käigus võrreldakse erinevaid intsidentide reageerimisplatvorme ning põhjendatakse, miks just TheHive 5 platvorm vastab kõige paremini SOC juhtkonna poolt sätestatud nõuetele. Esimese etapina seadistatakse valitud platvormi juurutuse katsetamiseks virtuaalne

testkeskkond, millele paigaldatakse konteineriseeritud IRP. Pärast esmast testfaasi luuakse organisatsiooni privaatpilves konteineriseeritud IRP tarbeks virtuaalserverid. Järgmise etapina lisatakse koodihoidla ja privaatpilve vaheline tarneahel, tagamaks platvormile järjepideva integratsiooni-, juurutus- ja arendusvõime. Seejärel lisatakse juurutatud IRP-le esmased integratsioonid andmerikastustööriistadega, luues aluse organisatsiooni SOAR süsteemi valmimisele. Viimases etapis analüüsitakse töös IRP ressursikasutust ning IRP tõhusust SOC meeskonna vaatest, et hinnata juurutuse mõju ning lisandväärtusi organisatsiooni turvalisusele ja intsidendihalduse protsessile.

1.1 Probleemi kirjeldus ja hetkeolukord

Organisatsiooni SOC on hetkeseisuga kasutanud turbeintsidentide haldamiseks ning reageerimiseks põhiliselt austusesiseseid suhtluskanaleid ja iTop (*IT Operations Portal*) tarkvara. Antud tarkvara miinuseks on vähene integreerimisvõimalus teiste toodetega, näiteks tulemüüride, logihaldussüsteemide ja pahavaratõrje süsteemidega. Samuti ei anna konkreetne platvorm head ülevaadet intsidendihalduse statistikast.

iTop[4] on avatud lähtekoodiga ITSM (*Information Technology Service Management*) tarkvara ning on ennekõike mõeldud organisatsiooni IT-haldus tarkvaraks, millega saab jälgida IT-teenuseid, -vahendeid ja -ressursse. iTop-i on ka sisseehitatud piletisüsteem, mille abil on võimalik organisatsioonis läbi viia turbeintsidentide haldust, kuid puuduvad intsidentide reageerimisvõimekusele orienteeritud automaatika.

Nagu eelnevalt mainitud, kasutab SOC intsidentidele reageerimiseks organisatsioonisiseselt suhtlusplatvorme. Kuigi suhtlusplatvormidele on integreeritud erinevaid tööriistu ja veebihaake, mis suhtlevad teiste süsteemidega ja lihtsustavad intsidendihaldurite ning -reageerijate tööd, ei ole suhtlusplatvormide kasutamine turbeintsidentide info talletamiseks parim lahendus. Esiteks on informatsioon killustatud ja täielik teave turbeintsidendi kohta ei pruugi jõuda ühte intsidendipiletisse kokku. Teiseks on keeruline piirata informatsiooni ligipääsetavust erinevatele kasutajatele ja partneritele, järgides kübervaldkonnas kehtivat fooriprotokolli (*Traffic Light Protocol*, TLP). Kolmandaks on raske luua usaldusväärset statistikat intsidendihalduse kohta, kui andmed on hajutatud erinevates suhtlusplatvormides. EVS-ISO/IEC 27035:2012 standardi[5] alusel tuleks talletada kogu intsidendipileti informatsioon kesksesse

infoturbesündmuste, -intsidentide ja -nõrkuste andmebaasi, mida haldab ISIRT (*Information Security Incident Response Team*) ehk infoturbeintsidentide tõrje rühm.

Tuginedes eelnevalt kirjeldatud vajadustele on oluline valida ja juurutada platvorm, mis toetaks SOC-i intsidendihaldurite ning -reageerijate tööd. Töökindel platvorm, mis hõlbustaks turbeintsidentidega seotud informatsiooni talletamist ning võimaldaks integreerida andmerikastustööriistu ja turbesüsteeme.

1.2 Ülesande püstitus ja eesmärk

Kuna organisatsiooni SOC meeskond areneb pidevalt, täiustudes uute liikmete ja tööriistadega ning vana intsidendihaldustarkvara ja -protsess oli ajale jalgu jäänud, tuli SOC juhtkonna poolt otsus välja vahetada intsidendihaldus platvorm. Varasemalt kasutatud intsidendihaldustarkvara ja -protsess on hakanud piirama SOC meeskonna ja tööriistade potentsiaali, intsidentide lahendamise kiirust ning ka intsidentide informatsiooni talletamis võimet. Täpsemalt on töö eesmärk lahendada järgmised ülesanded:

- Valida välja sobiv IRP lähtudes SOC juhtkonna poolt sätestatud kriteeriumitele (vt. ptk.1.5 lk.19).
- Valida välja sobivad haldustööriistad ja -meetodid, mis võimaldaks IRP juurutada SOC-i poolt ette nähtud taristutele.
- Lähtudes *The Twelve-Factor App* metoodikast[6], kasutada IRP juurutamisel ühte koodibaasi ning seeläbi tagada eraldiseisvad, kuid homogeensed arendus-, test- ja tootmiskeskkond.
- Juurutada valitud platvorm organisatsiooni privaatpilves.
- Luua esmased IRP ja andmerikastustööriistade vahelised integratsioonid.

Kokkuvõtteks on käesoleva lõputöö eesmärgiks juurutada intsidendihaldus ja reageerimistarkvara organisatsiooni SOC-is, mis võimaldaks integreerida erinevaid turbe-, andmerikastustööriistu ja süsteeme, kuid samal ajal järgides SOC juhtkonna poolt kehtestatud kriteeriume (vt. ptk.1.5 lk.19). Töös keskendutakse peamiselt TheHive 5 IRP ehitusele ja juurutusele SOC-is. Tulemusena saavutatakse oluline samm turbevaldkonna jätkuvas arengus, mis aitab tagada organisatsioonile tõhusama turvalisuse ja turbestrateegia.

1.3 Töö aktuaalsus

Intsidentide lahendamise- ja reageerimisplatvormid on väga olulised, sest need toetavad turbemeeskonna kiiret ja tõhusat reageerimist ning aitavad minimeerida rünnete võimalikku kahju. Automatiseeritud platvormid võimaldavad turbeintsidentidele reageerida kiiremini, tagades järjepideva ja täpse intsidendireageerimise protsessi täitmise. Automatiseeritud platvormide kasutamine on laialt levinud nii era- kui ka avalikus sektoris ning nõudlus nende järele kasvab pidevalt.

Ebasobiv platvormivalik võib põhjustada suurt ressursikulu olemasolevate turbesüsteemidega integreerimisel, mis tähendab, et platvormi maksumus ja integratsioonidele kuluv ressurss on ebaoptimaalne. See eeldab suuremat meeskonna panust automatiseeritud süsteemi käitamisel ja arendamisel, mis omakorda tähendab täiendavaid eesmärgipäratuid kohustusi turbemeeskonnale, tõstes sellega oluliselt turberiski organisatsioonis.

Seetõttu on oluline uurida ja analüüsida erinevaid intsidentide reageerimisplatvorme, et leida tõhus ja automatiseeritud lahendus turbekeskusele. Platvorm peaks olema võimalikult suurel määral integreeritav olemasolevate süsteemidega, piiratud inimtööjõu ja rahalise ressursi tingimustes, vastama SOC juhtkonna poolt seatud kriteeriumitele ning olema kõige suurema kasuteguriga meeskonna ülesannete vaatest.

1.4 Kasutatud metoodika

Lõputöö läbiviimisel kasutatakse kvalitatiivset uurimismeetodit, kus uurimisobjektiks on automatiseeritud intsidentide reageerimisplatvorm, kaasates võrdlusesse erinevaid reageerimisplatvorme. Juurutamisel analüüsitakse erinevaid haldusvahendeid ja -meetodeid, millega oleks sobilik reageerimisplatvormi SOC-is rakendada. Järeldustes analüüsitakse ja hinnatakse juurutatud keskkonna tõhusust, võttes arvesse nii platvormi ressursikasutust kui ka efektiivsust intsidentide lahendamisel. Kui saadud tulemused on ebarahuldavad, uuritakse, millised põhjused viisid selleni, et juurutatud keskkond ei osutunud kasulikuks või efektiivseks.

1.5 Eeldused, tingimused ja piirangud

Antud töös käsitletava platvormi juurutamisel on SOC meeskonna poolt antud täpsed kriteeriumid, millest peab kinni pidama. Lisaks peab eeldama, et SOC meeskonnal on olemas vajalik infrastruktuur, tööriistad ja teadmised platvormi juurutamiseks ning integreerimiseks teiste turbetööriistadega. Järgnevalt on välja toodud SOC juhtkonna poolt kehtestatud kriteeriumid intsidendihaldus- ja reageerimisplatvormile ning selle juururamisele.

Funktsionaalsed nõuded platvormile:

- Platvormi peab olema võimalik üle API (*Application Programing Interface*) integreerida MISP-iga (*Malware Information Sharing Platform*).
- Platvormi peab olema võimalik üle API integreerida andmerikastustööriistadega ja andmebaasidega, nagu näiteks Cortex, VirusTotal, DomainTools, Cuckoo liivakast jpt.
- Platvorm peab toetama ülesannete määramist meeskonnaliimetele intsidendipileti sees. Lisaks lubama mõõta juhtumite ja ülesannete arvu teatud aja vältel ning juhtumi lahendamisele kulunud aega.
- Platvormil loodud juhtumitele ning ülesannetele peab olema võimalik lisada TTP (*Tactics, Techniques, and Procedures*) klassifikaatoreid ja ohuindikaatoreid (turvarikkemärk , *Indicator of Compromise*, IoC).
- Platvorm peab võimaldama koguda statistikat juhtumite, häirete ja ülesannete kohta ning suutma luua kogutud statistika põhjal armatuurlauale diagramme.
- Platvorm peab suutma mõõta ja kuvada turbeintsidentide reageerimisprotsessi keskse soorituse indikaatoreid (*Key Performance Indicators*, KPI).
- Platvorm peab suutma luua juhtumeid MISP-i sündmuste põhjal.
- Platvorm peab lubama avada lahendatud ja arhiveeritud juhtumeid vigade parandamiseks ja andmete muutmiseks.
- Platvormi peab olema võimalik integreerida turbetööriistadega üle API liidese.
- Platvormis peab olema võimalik luua uusi andmevälju ja silte ning loodud andmevälju ja silte peab olema võimalik lisada juhtumitele, häiretele, ülesannetele või API päringutele.

Mittefunktsionaalsed nõuded platvormile:

- Platvorm peab olema kiiresti juurutatav uude infosüsteemi, paigaldus peab käima vaid mõne käsurea sisestusega.
- Platvormi peab olema võimalik paigaldada õhkeraldatud (*Air-gapped*) süsteemina ja platvorm peab töötama ilma interneti ühenduseta.
- Platvormi peab olema võimalik paigaldada lühikese ajajooksul korduvalt (erinevad instantsid) küberõppuste ning meeskonna koolituste tarbeks.
- Eelistatud on platvorm, mis toetab konteineriseerimist ja konteinerkeskkonnas töötamist, näiteks Docker või Kubernetes tehnoloogial.
- Tarkvaral peab olema lühike ja agiilne arendustsükkel ning tootjapoolne tugiteenus.
- Eelistatakse platvormi, mis on avatud lähtekoodiga ning pakub ka tugiteenust.
- Platvormi valikul eelistatakse finantsiliselt vähemkulukat platvormi, mida on tööjõu abil võimalik lihtsamalt juurutada, integreerida ja käitada.
- Platvormi valikul eelistatakse tarkvara, mis ei nõua organisatsioonipoolseid lisaarendusi ehk tarkvara, mis võimaldab üle avaliku API integreerida teisi turbetööriistu, kirjutamata ümber algset tarkvara koodi.
- Eelistatakse platvormi, millega on SOC meeskond varasemalt kokku puutunud, soodustades efektiivsemat kasutamist ja sujuvamat üleminekut juurutatud platvormile.

2 Töö teoreetilised alused

Käesolev peatükk keskendub turbeintsidentide haldamise ja reageerimise põhimõtete, parimate praktikate ning turbevaldkonnas kasutatavate tööriistade ja platvormide selgitamisele. Peatüki eesmärk on luua teoreetiline alus, millele toetub käesoleva lõputöö praktiline osa. Teisisõnu pakub peatükk vajalikku teoreetilist tausta küberturbe valdkonna aspektidele, mida käesoleva lõputöö praktilises osas kasutatakse.

Esimene alampeatükk annab ülevaate SOC olemusest ning selle rollist organisatsiooni turvalisuse tagamisel. Järgnevalt käsitletakse turbeintsidentide haldamise ja reageerimise protsesse ning tutvustatakse IRP ja SOAR süsteeme. IRP ja SOAR süsteemidele tehakse võrdlus ja analüüs, mille tulemusel valitakse välja juurutatav IRP. Peatükk jätkub StrangeBee TheHive 5 IRP tutvustusega, kaasates sellese erinevaid vajalikke lahendusi, mis aitasid antud platvormi valida ning on igapäevases kasutuses SOC-i töös. Seejärel käsitletakse TheHive 5 juurutamiseks vajalikke süsteemihaldustööriistu ja -meetodeid.

2.1 Turbeoperatsioonide keskus ehk SOC

Turbeoperatsioonikeskus või turbekeskus (*Security Operations Center*, lühendatult SOC) on üksus, mis tagab turbeoperatsioonide võimekuse organisatsioonis. Tavaliselt ei käsitleta SOC-i ühe üksusena vaid pigem keeruka struktuurina, mille eesmärgiks on hallata ja kaitsta organisatsiooni üldist turbetaset[7]. SOC koosneb kolmest alustalast, nendeks on inimesed, protsessid ja tehnoloogiad. Sellist mudelit nimetatakse ka PPT (*People, Process, Technology*) raamistikuks. Turbevaldkonnas laialt kasutatav PPT raamistiku ideeks on leida tasakaal inimeste, protsesside ning tehnoloogia vahel[8]. Intsidendihaldus ja -reageerimisplatvormid ning SOAR süsteemid seovad need alustalad omavahel paremini kokku, pakkudes turbemeeskondadele standardiseeritud ja automatiseeritud tegevuskavasid ning protsessiahelaid, mille järgi turbeintsidente lahendada.

SOC-i meeskondades töötavad infoturbespetsialistid teostavad organisatsiooni võrguliikluse analüüsi, süsteemide- ja turbemonitooringut ning tegelevad

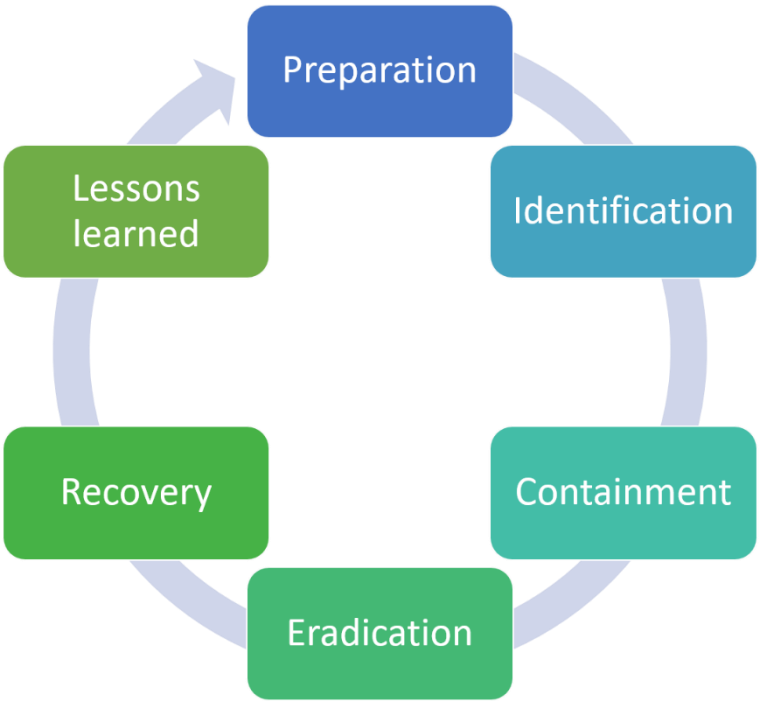
turbeintsidentide tuvastamise, reageerimise ja ohtude maandamisega. Järjepidev analüüs ja monitooring võimaldab SOC-il avastada võimalikke küberrünnakuid ning sekkuda kiiresti, kui midagi ebatavalist peaks juhtuma organisatsiooni infosüsteemides. Meeskondade eesmärk SOC-is on tagada organisatsiooni infosüsteemide ja andmete terviklus, konfidentsiaalsus, käideldavus ning äritegevuse järjepidevus. SOC võib olla osa organisatsiooni struktuurist või sisseostutuna teenusena ehk SOCaaS (*Security Operations Center as a Service*)[9].

2.2 Turbeintsidendile reageerimise protsessi parimad tavad

Iga organisatsiooni SOC lahendab ning reageerib intsidentidele lähtudes SOC-i väljakujunenud parimatest praktikatest, ehk teisisõnu teeb seda iga organisatsioon natuke isemoodi. Samas on küberturbes välja kujunenud üldisemad tavad ja protsessid, kuidas erinevat tüüpi turbeintsidendile reageerida ja neid lahendada. Üheks aluseks võib võtta näiteks *CERT* (*Computer Emergency Response Team*) *Société Générale* – Intsidentidele reageerimis metoodika 2022 (*Incident Response Methodologies 2022, IRM-2022*)[10]. IRM-2022 koodihoidlas on mitmeid näiteid eri tüüpi intsidentide lahendamiseks, sealhulgas näiteks: suhtlusründe, kalastusründe, hajusa ummistusründe ja lunavararünde kohta. Kui võtta ette näiteks IRM-2022 #5 võrguründe metoodika (*IRM #5 Malicious Network Behavior*)[11] siis üldjoontes näeb võrguründe puhul turbeintsidendi lahendamine ette järgnevaid samme:

1. Ettevalmistus: Intsidendikäsitlemise alustalaks on eelduste loomine ja valmistumine turbeintsidentide lahendamiseks. Organisatsioon peab tagama võrgumonitooringu taristu olemasolu, mis võib hõlmata selliseid seadmeid nagu EDR (*Endpoint Detection and Response*), NIPS (*Network Intrusion Prevention System*) ja IPS (*Intrusion Prevention System*). Lisaks sellele peab organisatsioon omama võimekust isoleerida lõppseadmeid võrgust.
2. Tuvastamine: Turbeintsidendi avastamine on võimalik, kui seadmed (EDR, NIPS, IPS) suudavad genereerida ja edastada sündmusi ning logi, mille analüüsimise põhjal oleks võimalik tuvastada võrguründe.
3. Ohu piiramine ja maandamine: Turbeintsidendi mõju piiramine saavutatakse näiteks, kui EDR isoleerib lõppseadme võrgust või IPS seade blokeerib võrguründe liiklust.

4. Kõrvaldamine: Turbeohu eemaldamine eeldab ründeallika blokeerimist, näiteks IP-aadressi, domeeninime, kasutajaagendi või ründaja sertifikaadi JA3 (*SSL/TLS Fingerprint*) räsi näol.
5. Taastamine: Süsteemide taastamine normaalseks tööks.
6. Õppetunnid: Koostada ja täiendada protsessi, analüüsida tekkinud valukohti.



Joonis 1. Intsidendile reageerimise kuus sammu. *DomainTools*[12].

Kokkuvõtvalt peaks iga intsidendilahendus ning reageerimine koosnema eelmainitud punktidest, sarnaselt joonisele (Joonis 1), millest kõige tähtsamad on just esimene (ettevalmistus - *preparation*) ja viimane (õppetunnid – *lessons learned*) punkt. Esimene punkt on eeldus, et organisatsioon üldse saaks antud turbeintsidenti lahendada ning viimane punkt aitab reageerimisprotsessi paremaks ja tõhusamaks muuta. Automatiseeritud töövood ja süsteemid hõlbustavad intsidendihaldurite tööd, muutes suurema osa töövoost automaatseks. SOAR ja IRP süsteemide abil suudetakse automatiseerida sammud 2-4, ehk rünnaku tuvastamine, ohu piiramine ja maandamine ning ohu kõrvaldamine.

2.3 IRP ja SOAR süsteemid

Küberturbe meeskonna üks tähtsamaid tööriistu on intsidendihaldus ja -reageerimisplatvorm, mille eesmärgiks on kokku koguda organisatsioonis toimuvad turbeintsidendid ja intsidentidega seotud informatsioon. Need platvormid võimaldavad kuvada pooleli olevaid juhtumeid ja vajadusel prioriseerida turbeintsidendipileteid luues seeläbi olukorrateadlikust ning aitades parandada turbemeeskonna efektiivsust ja kiirust intsidentide avastamisel, analüüsimisel ning lahendamisel. Lisaks võimaldavad sellised platvormid automatiseerida aeganõudvaid ja korduvaid ülesandeid vähendades sellega inimressursi kulusid ja vigu intsidentide lahendamisel. Tehnoloogia arenedes on jäänud käsitsi täidetavad intsidendihaldustarkvarad tahaplaanile ning aina rohkem saavad kõlapinda automatiseeritud süsteemid, mis muudavad intsidendihalduse ja -reageerimise kiiremaks ning lihtsamaks.

Intsidendihaldus ja -reageerimisplatvormid (IRP) koosnevad mitmest alamkomponendist, nagu teavituste ja juhtumite haldussüsteemist, statistika kogumis- ja kuvamissüsteemist ning informatsiooni ja ohuindikaatorite (IoC) talletussüsteemist. Lisaks pakuvad tänavapäevased platvormid ka hulgaliselt integratsioone väliste analüüsitööriistade ja andmebaasidega nagu näiteks ohuteadmus- ja taksonoomiabaasidega ning ründevektorite kaardistamistööriistadega[13].

SOAR (*Security Orchestration, Automation and Response*) süsteemid on reageerimisplatvormidest astme võrra kõrgemal, sidudes kokku mitmed erinevad tehnoloogiad ja turbetööriistad ning automatiseerides töövooge. SOAR süsteem võimaldab organisatsioonidel koguda turbeandmeid erinevatest allikatest ning automatiseerida ja orkestreerida turbeintsidentide lahendamise protsesse. SOAR pakub turbemeeskondadele ühtset lahendust turbeintsidentide haldamiseks ja reageerimiseks. SOAR seob kokku kolm peamist komponenti: turbe orkestreerimine, automatiseerimine ja reageerimine. Orkestreerimist teostav komponent koordineerib integreeritud turbetööriistade tööd. Integreeritud tööriistade hulka võivad kuuluda näiteks nõrkuseotsingu tööriistad, lõppseadme pahavaratõrje süsteemid, tulemüürid, sissetungituvastus ja -tõrje süsteemid, SIEM (*Security Information and Event Management*) süsteemid ning ohuteadmusbaasid. Automatiseerimine kiirendab töövooge täites ülesanded, mida varem teostasid turbemeeskonnad, näiteks logianalüüs, piletite kontroll ning täitmine. Reageerimisprotsess koosneb kindlatest eelkirjeldatud reeglitest,

mida SOAR järgib. Reageerimistööriistad võimaldavad näiteks analüüsi tulemusel halvaloomulised IP-aadressid blokeerida tulemüürist. Kokkuvõtvalt tarbib automaatiseerimist teostav komponent andmeid turbetarkvaradest, analüüsib kogutuid andmeid ning analüüsi ja reeglite tulemusel reageerib sündmustele[1].

Kuigi IT-piletihaldustarkvarasid on erinevaid, on suuremjaolt neid võimatu või väga keeruline rakendada intsidendihaldus ja -reageerimisprotsessides. Teisalt võivad need tarkvarad olla liiga keerukad ja mahukad, mistõttu nende täielikku funktsionaalsust pole turbeintsidendi haldamiseks vaja või on vastupidiselt liiga primitiivsed, suutmata pakkuda turbemeeskondadele vajalikku funktsionaalsust. Seetõttu sobivad lahendused nagu Atlassian Jira, ManageEngine ServiceDesk, iTop ja SysAid paremini üldotstarbelisteks IT-piletisüsteemideks. Võimalik on ka kasutada mitut platvormi paralleelselt ja integreeritult, näiteks intsidentidele reageerimine toimub ühe platvormi peal ning vajalik piletiinfo liigub teise platvormi, kuid selliste paralleelsüsteemide ehitus ning juurutus nõuab palju aega, raha ning arendus tööjõudu, mida igal organisatsioonil pole võimalik loovutada.

IRP ja SOAR süsteemide valik organisatsioonis võib-olla väga keeruline ning tavaliselt ei saagi sellist toodet osta nn karbitootena, kuna süsteemi juurutamise võimalikkus ja keerukus sõltub suuresti organisatsioonist ja selle struktuurist, tegevusvaldkonnast, toote kasutusjuhtudest ning juba olemasolevast tarkvarast, riistvarast ja tootjatest, mida ettevõte parasjagu kasutab või plaanib kasutama hakata.

Siinkohal on hea välja tuua, et iga IRP tarkvara ei ole SOAR tarkvara, kuid enamustes SOAR tarkvarades on mingil määral integreeritud intsidendilahendus ja -reageerimisvõimekus. Antud töö keskendub automatiseeritud intsidendireageerimise platvormile, mitte SOAR süsteemi juurutamisele, kuna viimane nõuab keerukaid integratsioone erinevate turbetarkvaradega, nagu näiteks SIEM, EDR ja nõrkuste skannerid. Töö eesmärk on juurutada integratsioonivõimaluste rikas IRP, mis saaks olla aluseks tulevasele SOAR süsteemile.

2.4 Platvormi valiku analüüs

Tuntuid piletihaldus tarkvarasid on mitmeid, kuid tunduvalt vähem on platvorme, mis on mõeldud just turbeintsidentide haldamiseks ja reageerimiseks. Aina rohkem koguvad ka

populaarsust pilveteenuste pakkujate poolsed turbeintsidentide reageerimisplatvormid ja SOAR süsteemid nagu näiteks Microsoft Sentinel SOAR ja Rapid7 SOAR, kuid käesolevas töös ei keskenduta pilvepõhistele SOAR lahendustele, kuna SOC juhtkonna üheks kriteeriumiks oli asutusesisene lahendus, mis suudab funktsioneerida ilma interneti ühenduseta. Lisaks langevad valikust välja üldotstarbelised piletihaldustarkvarad, kuna need ei paku funktsionaalsust, mida intsidentidele reageerijad vajavad oma töös. Samuti ei ole võimalik organisatsioonil juurde palgata arendajaid, kes hakkaks IT-piletihaldus tarkvarale juurde kirjutama funktsionaalsust lähtuvalt tööle seatud funktsionaalsetest nõuetest.

Kuigi maailmas on tuntuid SOAR platvorme kümne ringis, keskendub järgnev analüüs organisatsiooni jaoks parima lahenduse väljaselgitamiseks TheHive 5-le ning selle võimalikele alternatiividele. Tutvudes teiste küberturbe valdkondade asutuste kasutuses olevate lahendustega võeti võrdluse aluseks TheHive 5. Samas on platvorm ka varasemalt tuntud organisatsiooni SOC-i poolt, tänu oma paindlikkusele, avatud dokumentatsioonile ja heale tugiteenusele. Lisaks on TheHive 5 üle maailma CSIRT-de ja SOC-de poolt tunnustatud ja väga levinud platvorm ning juhtivad küberturbe organisatsioonid nagu SANS (*SysAdmin, Audit, Network and Security Institute*)[14] ja ENISA (*European Union Agency for Cybersecurity*)[15] pakuvad mitmeid koolitusi spetsiaalselt sellele platvormile. TheHive 5 on algselt kavandatud kui (automatiseeritav) IRP, ent erinevate süsteemidega integreerimisel võib see muutuda SOAR süsteemiks. Sellest tulenevalt klassifitseerivad tunnustatud IT-uuringute ja -analüüsi ettevõtted, nagu Gartner TheHive 5 SOAR süsteemina. Kuna SOAR platvormidele ei ole hetkel olemas ühtset võrdlusalust, leiti Gartneri ja G2 võrdluste tulemustel veel kaks SOAR süsteemi: IBM QRadar SOAR ja Splunk SOAR, mida pakub ka Eestis tegutsev küberturbelahenduste edasimüüja. Järgnevalt on välja toodud kolme platvormi iseloomustus ning võrdlus, mille aluseks on võetud Gartneri[16] ja G2[17] SOAR toodete ülevaated.

1. IBM Security QRadar SOAR: On üks maailma tuntumaid SOAR tooteid ja ühtlasi ka üks eelistatumatest SOC meeskondade poolt. IBM QRadar on tuntuks saanud oma SIEM tarkvara poolest, mida kasutavad paljud suurettevõtted üle maailma. IBM SOAR on tõhus intsidentide reageerimise ja automatiseerimise platvorm, mis aitab organisatsioonidel küberintsidentidele kiiremini ja tõhusamalt reageerida. See integreerub erinevate turbetööriistadega, pakkudes dünaamilist juhtumite käsitlemist, automatiseeritud reageerimist ning juhtumipõhist süvaanalüüsi võimekust[18].

Gartneri üldine hinne antud tootele on 3.2/5[16]. G2 ülevaade toob välja, et antud toodet kasutatakse põhiliselt IT-sektoris. Ning kasutatakse ettevõtetes, kus on rohkem kui 1000 töötajat[17].

2. Splunk SOAR: On juhtiv turbeintsidentide haldamise ja automatiseerimise platvorm, mis aitab turbeanalüütikutel reageerida kiiremini ja tõhusamalt ohtudele. See võimaldab organisatsioonidel integreerida erinevaid turbetööriistu ja automatiseerida rutiinseid toiminguid. Splunk on infoturbes laialt kasutatav tootja, mis lisaks SOAR süsteemile pakub ka logide kogumis süsteeme ja SIEM lahendusi. Splunk on eelkõige hinnatud oma tugeva ökosüsteemipoolt, pakkudes kõikvõimalike lahendusi organisatsiooni turbemonitooringuks[19]. Gartner-i üldine hinne antud tootele on 4.3/5. G2 ülevaade toob välja, et põhiliselt kasutavad toodet, investeerimispangad ja IT sektor[16]. Sarnaselt IBM QRadar-ile kasutatakse Splunk SOAR-i kui ettevõttes on rohkem kui 1000 töötajat[17].
3. StrangeBee TheHive 5: On intsidentide reageerimise ja turbeoperatsioonide automatiseerimise platvorm, mis pakub paindlikku ja kohandatavat lahendust. See sobib eriti hästi organisatsioonidele, kes soovivad oma turbemeetmete rakendamist isikupärastada ning otsivad taskukohasemat alternatiivi[13]. Gartner-i üldine hinne tootele on 4/5[16]. G2 ülevaade toob välja, et toode leiab kasutust põhiliselt finants-, telekommunikatsiooni- ja IT-sektoris. Toodet kasutavad ettevõtted, kus on töötajaid rohkem kui 50, kuid on hästi skaleeritav ning sobib ka suurtematele 1000+ töötajatega ettevõtetele[17].

Väljatoodud 3 toodet on seatud pingeritta indikatiivse hinna järgi. Selliste toodete puhul on väga keeruline pingerida moodustada, kuna avalikest allikatest on tehnilise dokumentatsiooni ja hinnaklassi leidmine keeruline. Enamus toodete andmed kaasaarvatud hinna pakkumised organisatsioonile on konfidentsiaalsed ning nõuavad kohtumist tootjapoolse müügiosakonnaga või edasimüüjaga.

Kõigi kolme platvormi kohta otsustati SOC juhtkonna poolt teha hinnapäring. Hinnapäringu tulemusel osutus kõige odavamaks tooteks StrangeBee TheHive 5. Indikatiivne hind Splunk SOAR lahendusele osutus ligikaudu ~150% ning IBM QRadar SOAR hind ~250% kallimaks TheHive 5 platvormi indikatiivsest hinnast. Enamjaolt sõltub IRP ja SOAR platvormide hind kasutajate arvust, kes antud platvormi kasutavad. Indikatiivse hinna vahe tuleneb ka platvormide eraldiseisvusest ehk odavamad platvormid ei vaja kindla tarkvaratootja kogu ökosüsteemi. Teisisõnu TheHive 5

indikatiivne hind on ka soodsam, kuna suudab integreeruda tööriistade ja toodetega tarkvaratootjast sõltumata.

Turul pakutavatest toodetest ning võrdluses olnud kolmest tootest osutus valituks StrangeBee TheHive 5 platvorm, mille eelkäija oli TheHive-Projecti loodud, avatud lähtekoodiga TheHive 4 platvorm, mis on laialdaselt kasutusel üle maailma erinevates organisatsioonides ning turbekeskustes. TheHive platvormide laiale kasutusele viitab vabavaraliste integratsioonide ning liidestuste hulk, mida on arendatud TheHive 4 platvormile ning on suuresti ühilduvad ka TheHive 5 platvormiga. Lisaks kaalus TheHive 5 valiku ülesse indikatiivne hind, mis võrreldes teiste sarnaste toodetega on üle kahe korra soodsam. TheHive 5 on saadaval freemium hinnastamismudeli alusel[13], mis SOC meeskonna vaatest on üpriski kasulik. Selline hinnastamismudel võimaldab lühiaegsetel projektidel, õppustel ja koolitustel kasutada TheHive 5 platvormi piiratud funktsionaalsusega tasuta ning organisatsiooni siseselt põhiplatvormina kasutada tasulist, täis funktsionaalsusega versiooni. Tänu TheHive 5 omadustele on platvormi mugav õppustele kaasa võtta, võimaldades taristu koodina meetodi abil tasuta platvormi versiooni kiiresti juurutamist. Võrreldes Splunk-i ja IBM-iga on TheHive 5 mõeldud väiksematele organisatsioonidele ning paljud vajalikud integratsioonid on juba olemas või kergesti teostatavad, ehk platvorm on rohkem nn karbitoote omadustega.

G2 alusel[17] IBM QRadar ja Splunk SOAR tootel tasuta versioonid puuduvad, võimalik on küll kasutada prooviversiooni, kuid pikas perspektiivis see SOC-i ei rahulda. StrangeBee TheHive 5 tasuline versioon maksustatakse kasutajate ja organisatsioonide arvu põhjal. Platvormi tasuta versioonil[13] on aga järgmised piirangud:

- Saab luua vaid 2 analüütiku kasutajat
- Saab luua vaid ühe organisatsiooni
- Pole võimalik kasutada simultaanteenindus (Multi-Tenancy) funktsiooni
- Lubab integreerida vaid ühe MISP ja Cortex serveri

TheHive 5 platvormi valik tulenes lisaks hinnale ja populaarsusele ka mitmetest teistest kaalukatest põhjustest. Esiteks vastas see platvorm enamikule SOC juhtkonna poolt esitatud nõuetele, mis näitab selle sobivust organisatsiooni vajadustega. Järgnevalt on autori poolt välja toodud kolm tähtsamat kriteeriumit platvormi valikule, analüüsides TheHive 5, IBM QRadar ja Splunk SOAR dokumentatsiooni:

1. Üheks tähtsamaks kriteeriumiks platvormile oli MISP ja Cortex süsteemi integreerimis võimalus üle API. TheHive 5 toetab selliseid integratsioone vaikimisi, kuid IBM QRadar ja Splunk SOAR puhul need vaikimisi toetatud ei ole ning mõlemad nimetatud platvormid nõuavad eraldiseisvaid rakendusi integratsioonide loomise jaoks.
2. Lisaks oli oluline TTP raamistiku olemasolu platvormis, sarnaselt eelnevale punktile on TheHive 5 TTP raamistik integreeritud platvormi, kuid IBM ja Splunk vajavad selleks eraldiseisvaid rakendusi.
3. Vägagi kaalukas kriteerium oli platvormide konteineriseerimis võimalus. TheHive 5 toetab dokumentatsiooni järgi konteineriseerimist ning on olemas ka tootjapoolsed Docker-i ja Kubernetes tõmmised. IBM QRadar toetab dokumentatsiooni järgi vaid virtualiseeritud lahendust ning eraldiseisvat installatsiooni serverile. Splunk SOAR toetab paigaldust kasutades AWS (Amazon Web Services) tõmmist või eraldiseisvat tarkvara pakki.

Lisaks täidetud kriteeriumitele on SOC meeskonnal eelnev kogemus TheHive 5 platvormi kasutamisega, mis aitas suuresti kaasa valikule, kuna see soodustab sujuvat üleminekut ja efektiivsemat kasutamist. IBM QRadar ja Splunk SOAR on võrreldes TheHive 5-ga ka rohkem ressursinõudlikud ehk TheHive 5-te saab käitada väiksema serveri peal. Lisas 16 (vt. Tabel 3 lk.69) on toodud välja tabel platvormide ressursikasutuse kohta lähtudes kättesaadavatel dokumentatsioonidel. Lisas 2 (vt. Tabel 2 lk. 55) on esitatud tabelina autori ja SOC-meeskonna vaatest kokkuvõtlik hinnang platvormide vastavuse kohta olulisematele kriteeriumitele, põhinedes avalikult kättesaadaval dokumentatsioonil, tutvustavatel ülevaadetel ning varasema kasutajakogemuse analüüsil.

2.5 Intsidentide reageerimisplatvorm StrangeBee TheHive 5

StrangeBee TheHive 5 on skaleeritav ning koostööle orienteeritud turbeintsidentide reageerimisplatvorm, mis on tihedalt integreeritud MISP-ga. TheHive 5 on loodud, SOC (*Security Operations Center*), CSIRT (*Computer Security Incident Response Team*), CERT (*Computer Emergency Response Team*) meeskondadele ja kõigi teiste infoturbepraktikute jaoks, kes tegelevad turbeintsidentidega, mida tuleb kiiresti uurida ja lahendada. TheHive 5 platvormil on mitmeid eeliseid ja erisusi. Platvorm toetab häirete

ja juhtumite haldusvõimalusi, detailset kasutajahaldust, teavitusraamistikku, detailset statistika ja armatuuride koostamist ning suurepäraseid API kasutamis võimalusi[13].

Järgnevates alampeatükkides kirjeldatakse detailselt TheHive 5 integratsioone, mis olid SOC juhtkonna poolt seatud olulisteks kriteeriumiteks platvormile. Esimeses alampeatükis antakse ülevaade MISP-i integratsioonile, mis oli üheks tähtsamaks kriteeriumiks platvormi valikul (vt. lk. 19). Lisaks kirjeldatakse MITRE ATT&CK raamistiku ning seejärel Cortex-i, mis on üks olulisemaid platvorme TheHive 5 töövoogude automatiseerimisel. Viimases alampeatükis tuuakse välja erinevad komponendid, mida TheHive 5 vajab töötamiseks.

2.5.1 MISP

MISP (*Malware Information Sharing Platform*) on avatud lähtekoodiga tarkvara, mis võimaldab organisatsioonidel, ettevõtetel ja üksikisikutel jagada, salvestada ning analüüsida küberturvalisusega seotud ohuteavet. MISP-i peamine eesmärk on aidata küberkogukonnal tõhusalt reageerida pahavaraga seotud rünnakutele ja turbeohtudele. Platvorm võimaldab kasutajatel töötada koos, automaatselt reageerida turbeohtudele ja jagada reaalajas teavet, nagu pahavara näidiseid, sündmuste kirjeldusi ning ohuindikaatoreid (IoC). MISP on välja töötatud just selle jaoks, et erinevad küberturbe valdkonnas töötavad osapooled nagu teadlased, valitsusasutused ja eraettevõtted saaksid automatiseeritult jagada omavahel teavet küberohtude kohta[20].

MISP-i on võimalik integreerida TheHive 5 platvormiga. TheHive 5 tootjate poolelt on integratsioon tugevalt soovitatud[21]. TheHive 5 kasutab MISP-i järgmistel viisidel:

1. Sündmuste ja ohuindikaatorite importime MISP-ist TheHive 5 platvormi. Importimise tulemusel on võimalik MISP-i sündmustest luua juhtumid ning seeläbi paremini analüüsida küberohte.
2. Automatiseeritud teabe jagamine. TheHive võimaldab kasutajatel jagada turbeintsidentide uurimistulemusi ja ohuindikaatoreid tagasi MISP-i parandades sellega kübervaldkonna koostööd.
3. Ohuteabe automaatne rikastamine. TheHive ja MISP-ga saab kasutada ühilduvaid andmete rikastamisvahendeid, mille tulemusel saab automaatselt koguda ja analüüsida ohtutega seotud teavet ja ohuindikaatoreid, nagu domeenid, IP-aadressid, failide räsid, sertifikaatide JA3 räsid jms.

4. Töövoogude loomine ja teavituste integreerimine.

Kokkuvõtvalt on MISP oluline platvorm küberkogukonnale, kuna see soodustab ohuteabe jagamist ja koostööd küberohtude tuvastamisel ja tõrjumisel. TheHive 5 kasutab MISP-i integreeritud lahendusena, et pakkuda tõhusat ja koostööd toetavat töövoogu[21].

2.5.2 MITRE ATT&CK raamistik ja TTP

MITRE ATT&CK maatriks[22] on küberturbe valdkonnas tunnustatud raamistik, mis pakub süstematiseeritud teavet ründajate taktikate, tehnikate ja protseduuride kohta (TTP, *Tactics, Techniques and Procedures*). Raamistik aitab küberturbe valdkonna spetsialistidel ja organisatsioonidel paremini mõista ja hinnata küberohtusid, arendades tõhusamaid kaitsestrateegiaid, mis põhinevad tegelikel ründajate käitumismustril. Lisaks aitab detailselt kirjeldatud raamistik parandada organisatsioonide vahelist koostööd. ATT&CK maatriks on järjepidevalt arenev tööriist, kuna uusi ründevektoreid, taktikaid ja tehnikaid lisatakse regulaarselt, et tagada raamistiku ajakohasus ja täpsus[23]. Paljud turbeasutused ja turbetarkvaratootjad integreerivad MITRE ATT&CK raamistikku oma toodetesse ja teenustesse, pakkudes sellega oma klientidele tõhusamaid ja täpsemaid küberturbelahendusi.

TheHive 5 platvorm toetab MITRE ATT&CK maatriksit, see on integreeritud platvormi analüüsi ja reageerimisprotsessidesse, tagades turbespetsialistidele parema arusaama ründajate poolt kasutatud taktikatest, tehnikatest ja protseduuridest. ATT&CK maatriksi abil saab SOC kaardistada organisatsiooni nõrgad kohad ja luua rünnetele tõhusamaid vastumeetmeid[21].

2.5.3 Cortex

Cortex on avatud lähtekoodiga analüüsimootor, toetamaks turbeoperatsioonide automatiseerimist ja integreerida erinevaid turbetööriistu. Cortex on TheHive Projecti (nüüd StrangeBee) loodud platvorm, mis integreerub TheHive 5 platvormiga, mille tulemusel saab platvormide koosluse liigitada SOAR süsteemiks[24], [25].

Cortex pakub lahendusi kahele küberturbe valdkonnas levinud probleemile. Esiteks, võimaldab Cortex suure hulga ohuindikaatorite ja vaatlusandmete analüüsimist ühe tööriista abil, vähendades mitme erineva vahendi vajadust. Teiseks aitab Cortex tagada

aktiivse ja automaatse reageerimise ohuolukordadele, samaaegselt toetades suhtlust teiste meeskondade ja osapooltega[25].

Cortex-i peamised kaks komponenti on analüsaatorid ja reageerijad ehk Cortex Neurons[26]. Analüsaatorite ülesanne on API päringu korral objekti või andmeid analüüsida ja rikastada. Näiteks saates tundmatu IP-aadressi VirusTotal-i kontrollimisele, saades tagasi informatsiooni antud IP-aadressi kohta teisisõnu toimub andmerikastus. Informatsioon loob ohuindikaatorile konteksti ning peale andmerikastust saab öelda, kas tegemist on healoomulise, halvaloomulise, või tundmatu IP-aadressiga. Reageerijate ülesanne on reegli, sündmuse või informatsiooni põhjal saata korraldus teise süsteemi. Näiteks tuntud halvaloomulise IP-aadressi tuvastamisel sundida tulemüüri blokeerima liiklust, mis pärineb selliselt IP-aadressilt. Selliste Cortex-i protsessiahelate loomisel on piiriks vaid oskused ja teadmised, kuna enamus vabavaralised turbetööriistasid toetavad API integratsioone. Cortex-i analüsaatorid ja reageerijad on tavaliselt eraldiseisvad programmi, kooditükid või konteinerid, mida Cortex vastavalt API päringule käivitab. TheHive Projecti poolt on loodud kohe kasutamiseks üle 100 Cortex Neuroni[26] sealhulgas näiteks tuntumad: AnyRun, Cuckoo Sandbox, CyberChef, MISP ja VirusTotal. Lisaks on võimalik ka ise luua oma vajaduste järgi analüsaatorid ja reageerijad[21].

2.5.4 TheHive 5 platvormi tööks vajalikud komponendid

TheHive 5 nõuab töötamiseks mitmeid erinevaid komponente. Antud platvormi peamine komponent on TheHive 5 mootor, mis tagab kogu platvormi ja integratsioonide töö. Platvorm vajab juhtumitega (häired, ülesanded, IoC-d) seotud andmete hoiustamiseks ka teksti indekseerimis- ja otsingumootorit ning TheHive 5 kasutab selleks Elasticsearch-i. Lisaks kasutab platvorm süsteemiandmete hoiustamiseks Cassandra NoSQL andmebaasi ning failiobjektide (näiteks manuste) hoiustamiseks avatud lähtekoodiga MinIO objektsalvestusteenust. Tegelikult on võimalik failiobjektide salvestamiseks kasutada ka lokaalset failisüsteemi, NFS (*Network File System*) võrgusalvestus teenust või mõnda teist objektsalvestusteenust, näiteks AWS (*Amazon Web Services*) S3 (*Simple Storage Service*) pilve teenust või CEPH-i [21].

Elasticsearch on avatud lähtekoodiga otsingumootor, mis võimaldab tekstipõhist otsingut ja suurandmete analüüsi ning põhineb Apache Lucene tehnoloogial. Elasticsearch-i kasutatakse põhiliselt suurandmete kogumiseks ja analüüsiks, sealhulgas logide, meetrika

ja turbesündmuste kogumiseks ja analüüsiks. Elasticsearch-i võlu peitub sissetuleva info kiires indekseerimises, mis võimaldab andmete kiiret ja mugavat otsimist ning visualiseerimist. Elasticsearch on osa Elastic Stack-i tööriistadest, mille hulka kuuluvad veel ka Kibana ja Logstash[27], [28].

Cassandra on avatud lähtekoodiga hajutatav ning skaleeritav andmebaasisüsteem, mis on mõeldud suurandmete hoiustamiseks ja haldamiseks. Cassandra on välja töötatud Apache Software Foundation poolt ning põhineb Google Bigtable ja Amazon Dynamo süsteemidel. Cassandra võimaldab sisestatud andmeid hoida nn suures arvus sõlmpunktides, mida saab vajadusel hajutada üle erinevate serverite või ka andmekeskuste (klastrid). Kokkuvõtvalt on Cassandra omadustelt NoSQL andmebaas, mis on kõrgkäideldav, skaleeritav ja kõrge rikketaluvusega ehk vastupidav riistvara- ning võrguriketele[29], [30].

Avatud lähtekoodiga ja skaleeritav objektsalvestusteenus MinIO võimaldab hoida andmeid ja faile suures mahus. MinIO on mõeldud ettevõtetele, kes soovivad talletada suurt hulka struktureerimata andmeid. Struktureerimata andmete hoiustamine on vajalik näiteks pilvepõhistele rakendustele, veebilehtedele, mobiilirakendustele, masinõppe mudelitele ning korreleeritavatele ja analüüsitavatele andmetele. MinIO on tuntud oma jõudluse ja vastupidavuse poolest, mistõttu on see ideaalne lahendus suurandmete töötlemiseks ja haldamiseks. MinIO skaleeritavus võime tagab lakkamatu töö andmemahu kasvades[31].

2.6 TheHive 5 juurutamiseks vajalikud süsteemihaldustööriistad ja -meetodid

Selleks, et antud platvormi oleks SOC-i kriteeriumite järgi võimalik paigaldada on vaja kasutusele võtta erinevad haldustööriistad ja -meetodid. Selles peatüks tuuakse välja haldusmeetodid ja -tööriistad, mida antud platvormi juurutamisel kasutatakse, lisaks analüüsitakse ja võrreldakse alternatiivseid tööriistu.

2.6.1 Taristu koodina

Taristu koodina (*Infrastructure as Code*, IaC) on tarkvaraarenduse ja süsteemiadministreerimise lähenemisviis, kus füüsilised ja virtuaalsed ressursid, nagu serverid, võrguseadmed, tarkvararakendused ja nendega seotud konfiguratsioonid, on

kirjeldatud koodina, kasutades kõrgema taseme programmerimiskeelt. Tarkvaraarenduse ehk rakendus koodina kvaliteeti tõstavad tööriistad nagu automaattestid, koodihoidlad ja pideva integratsiooni tarneahelad. Kasutades taristu koodina lähenemist on võimalik rakendada samu tööriistu, tehnikaid ja tarneahelaid taristu väljaehitamisel. IaC on tänapäeval möödapääsmatu haldusmeetod, kui soovitakse tagada kõrgkäideldavat pilvetaristut, kuna see võimaldab taristut hallata ja orkestreerida automaatselt vältides inimfaktorist tekkivaid vigu[32].

Rahuldades SOC juhtkonna poolt etteantud kriteeriumeid platvormile (vt. ptk.1.5, lk.19) on mõistlik läheneda projekti juurutamisel IaC vaatest. Koodi väljaarendamisel saab kasutada organisatsiooni koodihoidlat ja Git versioonihaldus. Kasutades IaC lähenemist saame koodis kirjeldada tingimused platvormi juurutamiseks. Lisaks muudab IaC paindlikumaks platvormi versiooniuuenduse ja rikke korral -tagasipöörde.

TheHive 5 platvormi juurutamiseks kasutatakse Ansible tööriistade komplekti, mis võimaldab kirjeldada kogu vajamineva taristu koodina. Pärast taristu kirjeldamist koodina piisab vaid mõnest käsureast, mis teeb kõik vajalikud toimingud platvormi ülesseadmiseks taristule.

Ansible tööriistadele on mitmeid alternatiive, näiteks Puppet, Terraform, SaltStack ja Chef. Töös on kasutatud Ansible tööriistu, kuna autor on varasemalt Ansible-ga palju kokkupuutunud, Ansible-t on lihtne õppida ja rohke dokumentatsiooni ning näidetega. Lisaks on Ansible süntaks YAML-põhine (*YAML Ain't Markup Language*), mis on kergesti inimloetav. Erinevalt Puppet-ist ja Chef-ist on Ansible agendivaba ega nõua kliendipoolset tarkvara. Viimaks on organisatsioonis kasutusel palju Ansible-ga ehitatud koodibaase, mis teeb uue infrastruktuuri kirjeldamise kergemaks.

2.6.2 Järjepidev integratsioon ja juurutus

Pidev integratsioon (*Continuous Integration*, CI) on koodi kirjutamise paradigma, mis suunab arendajaid tegema pidevalt väikseid koodimuudatusi ja salvestama muudatusi versioonihalduses tõstes sellega arenduse kvaliteeti. Järjepidev juurutus (*Continuous Delivery/Deployment*, CD) jätkab sealt kus, pidev integratsioon lõppeb ning automatiseerib rakenduste tarned soovitud keskkondadesse, sealhulgas tootmis-, arendus ja testkeskkondadesse[33]. Automatiseeritust aitab tagada tavaliselt tarneahel.

Vältimaks inimfaktorist tekkivaid vigu ning saavutamaks turvaline platvormi tarne arendus-, test- ja tootmiskeskkonda on töö autor valinud lisaks IaC meetodile ka CI/CD (*Continuous Integration, Continuous Delivery/Deployment*) meetodi. CI/CD meetod lubab defineerida kindlad reeglid, mille alusel kood hoidlast võetakse ning millistel tingimustel taristu või rakendus koodist juurutatakse. Lisaks võimaldab CI/CD meetod automaattestide lisamist vältimaks vigase koodi üleslaadimist taristule. Kasutades IaC ja CI/CD meetodeid, saab kirjeldada vajalikku infrastruktuuri koodina, tehes samaaegselt pidevaid, kuid väikseid muudatusi ning tarnida rakendus kasutades automatiseeritud tarneahelat.

Käesoleva projekti tarneahela loomiseks on kasutatud Jenkins-i tööriista, kuna Jenkins on populaarne, avatud lähtekoodiga, pika ajalooga, stabiilne ja pandliku konfiguratsiooniga CI/CD tööriist. Jenkins-ile on palju alternatiive nagu näiteks, JetBrains TeamCity, Atlassian Bamboo, GitLab CI/CD, ehk paljud koodihoidlate tootjad pakuvad ka CI/CD lahendusi. Valitud sai Jenkins, kuna vajalikud tööriistad, serverid ja süsteem on juba organisatsioonis paigaldatud ning autoril on ka eelnev kogemus Jenkins-iga. Lisaks ei vaja Jenkins kolmanda osapoole kasutajakontot ega interneti ühenduse olemasolu.

Lisaks tasuks mainida ka tööriista nimega HashiCorp Vagrant[34], mida kasutatakse lokaalses masinas TheHive 5 platvormi juurutamise testimiseks, vältimaks vigase taristu koodi jõudmist asutuse koodihoidlasse. Vagrant-i tööriist pakub lihtsat viisi, kuidas defineerida virtuaalmasinate loomist koodina. Paralleeli võib siin kohal tuua Docker Compose failiga, kus kirjeldatakse ära konteinerid, mis peavad koostööd tegema rakenduse töö tagamiseks. Sarnaselt töötab ka Vagrant, kuid koodis saab defineerida mitte ainult konteinerite loomist vaid ka virtuaalmasinate loomist olenemata virtualiseerimistarkvara tootjast (VMware, QEMU, Virtualbox, Hyper-V jpt). Lisaks virtuaalmasinate loomisele saab Vagrant-iga ka defineerida masina valmendus. See tähendab, et Vagrant suudab peale masina loomist lisada ka näiteks masinale DNS (*Domain Name System*) kirjed, käivitada Powershell või Bash käskluseid või hoopis Ansible Playbook. Lisaks on Vagrant-i ka lihtne paigaldada ja kasutada ning on alternatiiviks käsitsi hallatavatele virtuaalmasinatele.

2.6.3 Konteinerarhitektuur ja mikroteenused

Mikroteenused on arhitektuuriline lähenemisviis, mille puhul üks rakendus või platvorm koosneb mitmetest väikestest teenustest. Väiksed teenused on eraldiseisvad protsessid

ning suhtlevad tavaliselt üle võrguvirna (TCP/IP, *Transmission Control Protocol/Internet Protocol*) kasutades sageli HTTP (*Hypertext Transfer Protocol*) protokolli API päringuteks. Erinevalt monoliit rakendusega, mis on ehituselt üks tervik, võimaldavad mikroteenused muuta rakenduse osiseid eraldiseisvalt, segamata rakenduse funktsionaalsust. Selle tulemusel tagatakse rakenduse parem modulaarsus ja vajadusel ka skaleeritavus ning käideldavus[35].

Konteineriseerimine on rakenduse taristu virtualiseerimise tehnika, mis pakub isoleeritud ja ressursikasutuse poolest kergemat keskkonda. Selline tehnika võimaldab rakenduste kiiret arendamist ja juurutamist. Konteineriseerimine eristub traditsioonilisest virtualiseerimisest kus iga virtuaalmasin omab enda operatsioonisüsteem, konteineriseerimise puhul jagavad konteinerid hosti operatsioonisüsteemikihti[36].

Rakenduste konteineriseerimine aitab tagada mikroteenuste arhitektuurilise lähenemisviisi, isoleerides iga teenuse keskkonna ning võimaldades rakenduse toimimist sõltumata kasutatavast hosti operatsioonisüsteemikihist.

Konteinertööriistu on mitmeid, näiteks: Docker, Podman, CRI-O jpt. Lisaks on ka olemas konteinerorkestreerimise tööriistad, nagu Docker Swarm, Kubernetes, OpenShift jpt. Orkestreerimise tööriistad muudavad konteinerite halduse lihtsamaks. Käesolevas töös kasutatakse konteinerite loomiseks Docker-it ja orkestreerimistarkvarana Docker Swarm-i. Docker-i kasutamise põhjuseks on valmisolek infrastruktuuri näol. Lisaks on Docker laialt levinud ja kerge õppimiskõveraga platvorm. Docker-i valiku põhjuseks on ka see, et SOC-i infrastruktuuris pole veel valminud Kubernetes klaster ning hetkel kasutavad paljud teised SOC-i projektid Docker-it. Seega on kõige mõistlikum viis TheHive 5 platvormi juurutamiseks kasutada Docker-i konteinerite loomis- ja haldustööriistu.

2.6.4 Pöördproksi

Erinevalt tavalisest proksiserverist paistab pöördproksi server kliendile nagu tavaline veebiserver. Tehes veebipäringuid pöördproksi serveri poole otsustakse pöördproksi konfiguratsiooni abil kuhu tuleks veebipäringud saata. Pöördproksi peamine kasutusala on pakkuda internetiklientidele juurdepääsu tulemüüri taga asuvatele teenustele. Pöördproksi suudab päringuid suunata mitmete erinevate serverite poole. Lisaks võimaldab pöördproksi teha koormusejaotust erinevate serverite ja teenuste vahel[37].

Pöördproksi tarkvarasid on mitmeid nagu näiteks, Nginx, Apache, HAProxy, IIS, Traefik, Squid jpt. Enamus veebimootoreid toetavad pöördproksi seadistust ehk võimaldavad veebiserveril töötada pöördproksina.

DevOps (*Development Operations*) vaatest võib pöördproksi olla ka tööriist, mis lubab kerge vaevaga suunata veebiliiklus õigesse sihtpunkti. Lisaks lubab pöördproksi konfigureerida TLS-i (Transport Layer Security) tagades krüpteeritud võrguliikluse pöördproksi ja kliendi vahel. Võimalik on ka CSP-d (*Content Security Policy*) ehk turbepoliitikat rakendades muuta hallatav veebileht turvalisemaks.

Käesolevas töös on valitud pöördproksi tarkvaraks Nginx, kuna Nginx tagab head kiirust ja jõudlust ning on tarkvarana maailmas laialt kasutatav ehk Nginx-i kohta on mitmeid erinevaid foorumeid ja palju avalikku dokumentatsiooni. Lisaks on Nginx proksi väga paindlik ja kergelt konfigureeritav. Nginx proksi on ka SOC-is kasutusel ning konfiguratsioonist on loodud mitmeid malle. Mallid aitavad kerge vaevaga konfigureerida just vajaliku funktsionaalsuse TheHive 5 teenuse jaoks.

3 Lahenduskäik: TheHive 5 platvormi juurutamine

Käesolev peatükk keskendub TheHive 5 platvormi juurutamisele SOC-is, tutvustades kasutatud haldustööriistu ja -meetodeid. Esimene alampeatükk käsitleb platvormi algset konteineriseerimist, konfigureerimist ning testimist. Teises alampeatükkis kirjeldatakse platvormi paigaldus taristu koodina ning testitakse platvormi juurutust taristule Vagrant tööriista abil. Järgmisena tutvustatakse tarneahela loomist ja platvormi juurutamist asutuse privaatpilves. Peatükk lõppeb platvormi testimise alampeatükiga. Lisa 3 (vt. Joonis 3 lk. 56) pakub lihtsustatud ülevaadet platvormi juurutamise etappidest.

3.1 Platvormi konteineriseerimine ja testimine Docker abil

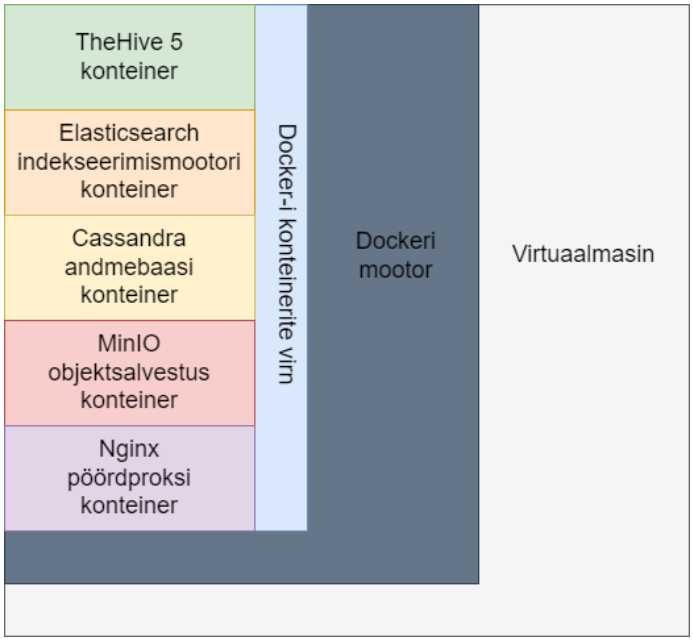
Projekti elluviimise käigus on esmaseks etapiks TheHive 5 platvormi vajalike konteineriseeritud teenuste koostöö testimine. Selle saavutamiseks on vaja arendusarvutis käivitada TheHive 5 teenuskonteinerite virn. Enne järgmiste sammudega jätkamist tuleb initsieerida Docker Swarm teenus. Docker Swarm võimaldab kasutada Docker Stack tööriista, mis on paindlikum kui Docker Compose, kuid konteinerite kirjeldamiseks kasutab samuti Compose faili. Pärast Docker Swarm initsieerimist on võimalik kasutada Docker Stack-i ning kirjeldada Compose faili abil mitme konteineri koostööd.

TheHive 5 Docker Compose faili dokumentatsioon on kättesaadav StrangeBee ametlikul veebilehel[1]. Kuigi dokumentatsioonis esitatud Compose fail võimaldab esialgse platvormi testkeskkonnas käivitada, ilmnevad selles mitmed puudused ressursikasutuse ja teenusetõrgetest taastumise osas. Need puudused võivad põhjustada teenuskonteinerite töö peatumise pärast teatud aja möödumist. Näiteks ei ole kõigi teenuste puhul piiratud mälu ja protsessori kasutust. Samuti ei ole käsitletud juhtumeid, kui teenuse töö lakkab või konteineri uuendamine ebaõnnestub. Seetõttu tuleb Compose faili täiendada ja muuta.

TheHive 5 Docker Compose fail koosneb neljast kirjeldatud teenusekonteinerist:

[1] https://docs.strangebee.com/thehive/setup/installation/docker/ (vaadatud 8. aprill 2023).

1. TheHive 5 mootori konteiner, mis on platvormi tuum ja pakub veebiliidest.
2. Cassandra andmebaasi konteiner, mis talletab platvormi süsteemiandmed.
3. Elasticsearch indekseerimis- ja otsingumootori konteiner, mis talletab juhtumitega seotud andmed.
4. MinIO objektsalvestusteenuse konteiner, mis talletab juhtumitega seotuid manuseid.



Joonis 2. Platvormi teenuskonteinerite topoloogia.

Lisaks nendele neljale konteinerile lisatakse virna viies konteiner - Nginx konteiner, mida tootjapoolsest Docker Compose failist ei leita. Joonisel 2 on kujutatud platvormi tööks vajalike konteinerite topoloogia (Joonis 2).

3.1.1 Nginx pöördproksi konteineriseerimine ja konfigureerimine

Nginx konteineri lisamine teenusevirna pakub platvormile mitmekülgset paindlikkust. Esiteks toimib Nginx konteiner pöördproksina, võimaldades suunata sissetulevat veebiliiklust TheHive 5 platvormile ning vajadusel ka teistele komponentidele, nagu MinIO halduspaneelile. Teiseks võimaldab Nginx hõlpsasti lisada teenusele veebisertifikaadi, krüpteerides seeläbi võrguliiklust kliendi ja Nginx pöördproksi vahel. Autor ei pea selles kontekstis vajalikuks krüpteerida võrguliiklust pöördproksi ja platvormi serveri vahel, kuna mõlemad konteinerid asuvad samal serveril. Lisaks saab Nginxi abil suunata kogu krüpteerimata HTTP (*Hypertext Transfer Protocol*) liikluse, mis jõuab porti 80, krüpteeritud HTTPS (*Hypertext Transfer Protocol Secure*) kanalile

ehk pordile 443. Kolmandaks võimaldab Nginx kiiresti luua läbipääsureeglid, tagades administraatoritele juurdepääsu haldusliidestele, kuid keelates tavakasutajatele neile juurdepääsu. Lõpuks aitab Nginx veebilehte turvalisena hoida, kuna Nginx konfiguratsioonis saab määrata CSP (*Content Security Policy*) ja ka teisi turbega seotuid päiseid.

Nginx konfiguratsiooni, TLS (*Transport Layer Security*) sertifikaati ja TLS sertifikaadivõtit hoitakse koodihoidlas krüpteerituna ning konteinerisse loetakse need Docker *Secret* ja Docker *Config* argumentide abil, mis määratakse Docker Compose failis. Järgnevas tabelis (Tabel 1) on esitatud TheHive 5 Docker-i teenusvirna käivitamiseks vajalikud failid ja nende otstarbed.

Tabel 1. Docker-i teenusvirna tööks vajalikud failid ja nende otstarbed.

Failinimi	**Otstarve**
TheHive_5-compose.yml	Docker Compose fail, milles on kirjeldatud Docker-i virna teenused.
nginx.conf	Pöördproksi konfiguratsioonifail, mis loetakse sisse TheHive_5-compose.yml faili käivitamisel.
nginx.cert	Pöördproksi avalik sertifikaadifail, tagamaks TLS-ühenduse kliendi ja proksiserveri vahel. Loetakse sisse paralleelselt TheHive_5-compose.yml failiga.
nginx.key	Avaliku sertifikaadi privaatvõti. Loetakse sisse paralleelselt TheHive_5-compose.yml failiga.
stack_deploy.sh	Bash skript, mis käivitab teenusvirna Docker Stack-is. (Lisa 7)

Katsetusetapis ei pidanud autor vajalikuks krüpteerida *nginx.key* faili ehk TLS-sertifikaadivõtit, kuna sertifikaat oli genereeritud testimise eesmärgil domeeni example.com jaoks. SOC-i virtuaaltaristu loomisel peidetakse ja krüpteeritakse kõik tundlikud andmed, sealhulgas (sertifikaatide võtmed, paroolid, kasutajanimed, API-võtmed jne).

Nginx konteinerteenuse kirjeldus Compose failis on väljatoodud lisas 4 (vt lk.57). Sarnaselt Nginx teenusele on ka kirjeldatud teisesed teenused Compose failis. Selles lisas on näha, et kõigepealt on kirjeldatud Nginx konteineritõmmis (*Container Image*). Konteinertõmmis asub organisatsiooni sisemises Docker-i tõmmiste hoidlas (*Docker*

Image Repository). Esmases arendusfaasis võib kasutada ka avalikke hoidlaid, kuid tootmiskeskkonnas pole serveritel avalikku internetti ligipääsu. Seega probleemi ennetamiseks laeti vajalikud konteineritõmmised privaatsesse hoidlasse. Nginx puhul on kasutuses Alpine Linux versiooniga 1.23.3. Alpine Linux valiti, kuna see on konkurentidest oluliselt kergema jalajäljega ehk mahult väiksem. Peale teenusetõmmise määramist on ära kirjeldatud võrgud, mille külge antud teenus ühendub. Kõik konteinerteenused on isoleeritud ja suhtlevad kindlaksmääratud reeglite alusel, tagades turvalisuse ja vähendades teenuste kompromiteerimise võimalusi. Lisa 15 (vt. Joonis 5 lk.68) pakub ülevaadet konteinerteenuste võrgust. Lisaks võrgule on määratletud ka võrgupordid, mis avatakse Nginx pöördproksiga suhtlemiseks. Pöördproksi jaoks tuleb avada HTTP ja HTTPS pordid ehk port 80 ja 443. Viimaks on kirjeldatud Nginx konfiguratsiooni, sertifikaadi ja sertifikaadivõtme paigaldamiseks vajalikud lähte- ja sihtkohad.

Nginx teenuse konfiguratsiooni näidis on välja toodud lisas 5 (vt. lk.58). Osa konfiguratsioonielementidest on peidetud, tagades ümbritseva infrastruktuuri kohta käiva info konfidentsiaalsuse. Konfiguratsiooni esimeses serveriplokis on määratletud liikluse ümbersuunamise reegel. See tähendab, et kui liiklus jõuab porti 80, suunatakse klient ümber krüpteeritud HTTPS pordile 443. Antud näite puhul kasutatakse ümbersuunamiseks käsitsi seadistatud URL-i (*Uniform Resource Locator*). Automatiseerimis käigus asendatakse käsitsi sisestatud väärtused automaatsete muutujatega vastavalt DNS konfiguratsioonile. Järgmises serveriplokis on määratletud juhud, kui liiklus jõuab krüpteeritud 443 pordile. Plokis on määratletud, kus asuvad TLS sertifikaat ja TLS võti, ning lisaks on määratud, et kasutatakse TLS-i versiooni 1.3. Veebilehe turvalisuse tagamiseks on määratletud ka CSP päis ja muud turbesätted, nagu näiteks päis *X-XSS-Protection*. Veebiserveri turbesätete üksikasju käesolevas töös ei käsitleta. Lõpuks on määratletud TheHive 5 platvormi asukoht, kuhu Nginx peab liiklust suunama. Platvormile liikluse suunamiseks kasutatakse Docker-i sisemist DNS-i, mis leiab teenuskonteineri võrgust ülesse teenuse nime järgi. Selle tulemusel suunatakse pöördproksi pordile 443 saabunud liiklus edasi teenuskonteinerile nimega *thehive* pordile 9000. Visuaalne ülevaade Nginx-i pöördproksist on toodud lisas 6 (vt. Joonis 4 lk.59).

Nginx pöördproksi konteineriseerimine ja konfigureerimine võimaldavad TheHive 5 platvormile suuremat paindlikkust ja turvalisust, pakkudes mitmeid eeliseid, nagu veebiliikluse suunamine, veebisertifikaadi lisamine, läbipääsureeglite loomine ja

turbepäisete lisamine. Docker Compose faili abil saab hõlpsasti hallata Nginx konteinerteenuse konfiguratsiooni ja seoseid teiste teenustega, tagades süsteemi turvalisuse, stabiilsuse ja modulaarsuse.

3.2 Taristu koodina kirjeldamine ja Vagrant testkeskkond

Eelnevas etapis edukalt läbitud konteineripõhise teenusvirna seadistamine tagab rakenduse kõrgeima kihi korrektse töö. Järgnevas alajaotises keskendutakse Ansible Playbook-i loomisele, mis võimaldab sujuvat platvormi tarnet sõltumata kasutatavast taristust. SOC-i juhtkonna esitatud kriteeriumide kohaselt (vt. ptk.1.5 lk.19) peab platvorm olema paigaldatav ka privaatpilve välistesse keskkondadesse. Selle saavutamiseks peetakse sobivaks kasutada Vagrant tööriista, mis võimaldab kiiresti seadistada vajalikke virtuaalmasinaid. Vagrant tööriistaga testimiskeskkonna loomine ja Ansible Playbook väljatöötamine toimuvad paralleelselt, kuna iga Ansible Playbook käivitamisega ilmnevad vead, mida tuleb koodis parandada ning seejärel katsetada parandatud vigu testkeskkonna.

3.2.1 Vagrant tööriista abil testkeskkonna loomine

Vagrant-il põhinev testimiskeskkond pakub projekti arendamisel mitmeid eeliseid. Esiteks võimaldab Vagrant kiiresti servereid luua ja kõrvaldada, mis on tõhusam ja kiirem kui organisatsiooni privaatpilves, kus virtuaalmasinate loomisel tuleb määratleda ka kõik sellega seotud komponendid (võrgud, turbegrupid, veebilüüsid, ligipääsuvõtmed jne). Teiseks on vaja testida platvormi paigaldamist privaatpilve väliselt, kuna see oli üks SOC-i juhtkonna kriteerium.

Vagrant tööriistade paigaldamine arendus tööjaamale on lihtne ja mugav. Lisaks Vagrant-ile on vaja paigaldada ka hüperviisori tarkvara ning Vagrant-i ja hüperviisori vaheline pistikprogramm, mille abil Vagrant loob ning haldab virtuaalmasinaid. Käesoleva testimiskeskkonna loomisel valiti hüperviisoriks Oracle VirtualBox, kuna see on populaarne, tasuta kättesaadav ja kergesti paigaldatav. Alusmasina operatsioonisüsteemiks valiti Ubuntu 20.04. Täpsemad paigaldusjuhised on leitavad

Vagrant-i ametlikult veebilehelt[1]. Lisas 8 (vt. lk.61) on esitatud Bash skript Vagrant-i paigaldamiseks.

Pärast Vagrant-i paigaldamist saab loodava taristu kirjeldada Vagrantfile abil. Selle faili abil loob Vagrant virtuaalmasinaid. Lisa 9 (vt. lk.62) sisaldab loodud Vagrantfile näidet. Näite põhjal loob Vagrant Ubuntu 20.04 virtuaalmasina ja sobiva võrguadapteri. Virtuaalmasinale antakse 8Gb virtuaalmälu ja 2 virtuaalprotsessorit. Seejärel luuakse muutuja nimega *$script*, millele omistatakse organisatsiooni privaatsed Ubuntu tarkvarahoidlad ja määratakse DNS-serveri asukoht võrgus. Seejärel taaskäivitatakse loodava virtuaalserveri nimelahendus teenus, et DNS serveri asukoha muudatus jõustuks. Järgnevalt käivitatakse virtuaalmasinas käsuviip ja antakse sisendiks eelmainitud muutuja. Viimase etapina käivitatakse Ansible Playbook, mille ülesandeks on lõpetada masina seadistamine ja paigaldada TheHive 5 platvorm. Playbook-i käivitamisel antakse ette ka mõned argumendid, mis tagavad Ansible korrektse töö. Argumentide hulka kuuluvad Playbook-i faili nimi, Ansible Vault-i[2] salafraas, tingimuslikult täidetavate ülesannete silt (*Tag*) ning Ansible rollide asukohad.

3.2.2 Taristu koodina: Ansible Playbook

Taristu automatiseeritud paigaldamise ja haldamise võimaldamiseks tuleb kogu taristu kirjeldada koodina. Selle saavutamiseks kasutab autor Ansible Playbook-e, mis võimaldavad taristu paigaldamist ja konfigureerimist lihtsalt ja tõhusalt. Käesolevas töös on loodud neli erinevat Playbook-i, mis on omavahel seotud ja täidavad erinevaid ülesandeid.

Kolm peamist Playbook-i hõlmavad järgmist: infrastruktuuri loomine (*playbook-infrastructure.yml*), hosti konfigureerimine (*playbook-host.yml*) ning rakenduse paigaldamine (*playbook-app.yml*). Lisaks on kasutusel Vagrant Playbook (*vagrant-playbook.yml*), mis ühendab hosti ja rakenduse Playbook-id ning määratleb vajalikud muutujad Vagrant keskkonnas, nagu avalik Nginx veebisertifikaat ja krüpteeritud sertifikaadivõti. Näide *vagrant-playbook.yml* kohta on toodud lisas 10 (vt. lk.63). Käesolevas töös ei keskenduta *playbook-infrastructure.yml* loomisele, kuna see on

[1] https://developer.hashicorp.com/vagrant/docs/installation (vaadatud 8. aprill 2023).

[2] https://docs.ansible.com/ansible/latest/vault_guide/index.html (vaadatud 8. aprill 2023).

valmistatud organisatsiooni poolt ning on spetsiifiline organisatsiooni taristu jaoks. Siiski tuleb autoril luua ja kohandada *playbook-host.yml* ning *playbook-app.yml*, mis peale taristu loomist seadistavad hosti ning paigaldavad hostile TheHive 5 platvormi.

3.2.3 Taristu koodina: Hosti konfiguratsioon koodina

Nii Vagrant-keskkonnas kui ka privaatpilve taristul on vajalik esmalt konfigureerida host. Hosti konfiguratsiooni eest vastutab *playbook-host.yml* (näide vt. Lisa 11 lk.64). Kuigi antud Playbook on suhteliselt vähese koodiga, täiustub see pidevalt uute ülesannetega. Mainitud Playbook määratleb neli ülesannet, mida tuleb Ansible-l täita.

Mitmetes ülesannetes kasutatakse organisatsiooni poolt määratletud Ansible rolle. Ansible rollid[1] vähendavad koodi dubleerimist ja muudavad korduvkasutatavad ülesanded efektiivsemaks. Lisa 11 (vt. lk.64) näites on taaskasutatud organisatsiooni poolt loodud kolme rolli: Ubuntu automaatsete uuenduste loomise roll, NTP (*Network Time Protocol*) serveri konfiguratsiooni roll ja SNMP (*Simple Network Management Protocol*) monitooringuagendi konfiguratsiooni roll. Kuigi käesolevas töös ei keskenduta organisatsiooni poolt loodud Ansible rollide üksikasjalikule kirjeldamisele, selgitatakse lühidalt, mida iga roll teeb. Antud Playbook-i viimane ülesanne on autori poolt loodud ning pole otseselt rollipõhine, kasutades seda ainult konkreetse Playbook-i piires.

Esimene roll hõlmab Ubuntu automaatsete uuenduste seadistamist, tagades hosti ajakohasuse ja kõigi turbeuuenduste automaatse paigaldamise. NTP roll seadistab hosti jaoks NTP serveri asukoha võrgus. SNMP roll konfigureerib hosti saatma monitooringu jaoks vajalikku teavet, määratledes ka SNMP kasutajatunnuse, autentimise tüübi, autentimise parooli ja liikluse krüpteerimise salafraasi.

Viimane ülesanne hõlmab hostile püsiva salvestusruumi konfigureerimist. Kuna töötaval hostil on kaks virtuaalketast (üks operatsioonisüsteemi ja teine andmete jaoks), tuleb enne teise ketta kasutusele võtmist teha mitmeid toiminguid, et see muuta kasutatavaks. Esmalt tuleb luua uus LVM (*Logical Volume Manager*) sektioon (*Partition*), seejärel luua LVM sektsioonile *ext4* tüüpi failisüsteem. Järgnevalt on vaja luua haakekoht (*Mount Point*) nimega *thehive_data* ja lõpuks haakida (*Mount*) loodud failisüsteem sellele kohale.

[1] https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_reuse_roles.html (vaadatud 8. aprill 2023).

Haagitud kausta ülesandeks saab olema Docker-i andmete salvestamine. Täpsema koodinäidise püsiva salvestusruumi loomisest leiab lisast 12 (vt. lk. 65).

3.2.4 Taristu koodina: Platvormi konfiguratsioon koodina

Sarnaselt *playbook-host.yml*-le on loodud *playbook-app.yml* ning selle eesmärk on lõpule viia kõik platvormiga seotud seadistused. Nende hulka kuuluvad näiteks Docker-i paigaldamine, Docker Swarm-i initsialiseerimine ja lõpuks Docker Stack-i käivitamine, kasutades Docker Compose faili muutujana. Täpsema koodi näite leiab lisast 13 (vt. lk.66). Selles Playbook-is on esimene ehk Docker-i installatsiooni roll imporditud organisatsiooni koodihoidlast, samas kui viimased kaks rolli on loodud autori poolt, kuna need ei olnud organisatsiooni koodihoidlas saadaval.

Antud Playbook-i esimene ülesanne on paigaldada ja konfigureerida Docker, kasutades organisatsiooni poolt loodud Ansible rolli. Sellele rollile antakse kaasa muutujad, mis hõlmavad lisapakke Docker-i haldamiseks ning sisemise tarkvarahoidla aadressi, kust Docker-i saab alla laadida. Seejärel muudetakse Docker-i failide salvestamise asukoht eelnevas Playbook-is loodud püsivale salvestuspinnale. Viimaks konfigureeritakse loodavate konteinerite IP-aadresside vahemik, et vältida konflikte organisatsiooni olemasoleva IP-aadressi ruumiga.

Teine ülesanne on autori poolt loodud ning sisaldab Docker Swarm-i initsialiseerimist. Selle ülesande käigus laaditakse alla vajalikud Python-i teegid, et tagada ühilduvus Ansible ja Docker Swarm-i vahel, ning lõpuks initsialiseeritakse Docker Swarm ehk konteinerorkestreerimise tööriist.

Playbook-i viimases ülesandes käivitatakse loodud konteinerite virn Docker Swarm-i peal. Ansible ülesandele antakse kaasa kolm muutujate faili. Esimeses muutujafailis on määratletud kõik krüpteeritud salafraasid ja kasutajatunnused, nagu Docker-i tõmmiste allalaadimiseks mõeldud tarkvarahoidla mandaadid, samuti TheHive 5, Elasticsearch, MinIO ja Cassandra salafraasid. Teise muutujana antakse kaasa Docker Compose fail ning viimase muutujana Nginx konfiguratsioon, sertifikaat ja sertifikaadivõti. Antud ülesanne seab valmis kogu platvormi, kasutades Ansible Docker Stack liidest ja kolme muutuja faili.

Valminud Ansible Playbook failid ja Vagrant-i konfiguratsioon võimaldavad kiiret platvormi juurutust taristule.

3.3 Järjepideva integratsiooni ja juurutuse tarneahela loomine

Pärast taristu koodina kirjeldamist ja Vagrant testkeskkonnas platvormi paigalduskoodi testimist on aeg luua tarneahel, et tagada lihtsustatud ning kontrollitud platvormi tarne privaatpilve taristule. Jenkins tarneahela loomiseks tuleb konfigureerida projektipõhine Jenkinsfile ja organisatsiooni koodihoidla veebihaak, mis suudab koodi üleslaadimisel Jenkins-it teavitada. Jenkinsfile näidis on välja toodud lisas 14 (vt. lk.67). Näite failis on ära kirjeldatud neli etappi.

Esimeses etapis paigaldatakse vajalikud organisatsiooni poolt loodud Ansible rollid. Teises etapis käivitatakse Ansible Playbook, mis seadistab vajaliku infrastruktuuri. Muutujatena lisatakse koodihoidla SSH (Secure Shell) privaatvõtme identifikaator, et Jenkins saaks platvormi koodihoidlast kätte. Järgnevalt lisatakse inventuuri muutuja, mis näitab, milliste serverite suunas hakkab Ansible Playbook tegevusi tegema. Tegemist on dünaamilise muutujaga, mis sõltub soovitud keskkonna paigaldamisest (tootmis-, test- või arenduskeskkond). Järgmise muutujana lisatakse Ansible Playbook-i failinimi, mida tuleks antud etapis käivitada. Viimase muutujana lisatakse Ansible Vault-i dešifreerimisvõtme identifikaator, mis võimaldab krüpteeritud Ansible muutujaid avada (näiteks kasutajanimed, paroolid või mandaadid). Kolmas ja neljas etapp sarnanevad teisele etapile muutujate poolest, kuid kolmanda etapi eesmärk on konfigureerida host (NTP, SNMP, Uuendused) ja neljanda ehk viimase etapi ülesanne on viia lõpule platvormi ülesseadmine (Docker, Docker Swarm, TheHive 5 platvorm). Jenkinsfile loomise järel on tarneahel edukalt loodud. Järgmiseks tuleb projekti kood edastada koodihoidlasse, mille alusel Jenkins loob taristu, konfigureerib hosti ning paigaldab TheHive 5 platvormi.

3.4 Platvormi konfigureerimine ja integratsioonid

Peale platvormi edukat juurutust asutuse privaatpilves tuleb teostada esmased teenuste konfiguratsioonid. Käesoleva töö raames on vaja seadistada ainult Cortex ja MISP integratsioonid ning MinIO salvestusruum, kuna hetkel ei ole Ansible Playbook neid konfigureerinud.

Cortex-i konfigureerimisel tuleb esmalt siseneda Cortex-i veebilehele. Pärast sisselogimist saab luua kasutaja ning genereerida kasutajale API võtme. Seejärel tuleb siseneda TheHive 5 platvormi veebilehele ning määrata Cortex-i serveri asukoht võrgus, kasutades FQDN-i (*Fully Qualified Domain Name*) või IP-aadressi, ning sisestada Cortex-is loodud TheHive 5 kasutajakonto API võti. Mõne sekundi pärast ilmub roheline tuli, näidates ühenduse loomist Cortex-i serveriga. Sarnaselt saab MISP-i integreerida TheHive 5 platvormiga.

Teise etapina tuleb MinIO portaalis määrata TheHive 5 salvestusruumi identifikaator. See on vajalik, et TheHive 5 saaks salvestada intsidentide piletitele lisatud manuseid. Selleks tuleb siseneda MinIO halduspaneeli veebilehele ning määrata sama identifikaator, mida kasutati TheHive 5 Compose failis. Pärast salvestusruumi määramist saab proovida manuse lisamist, et hinnata seadistuse õnnestumist.

Lisaks konfigureeriti platvormile privaatpilve-põhine varundus, et tagada andmete säilimine platvormi rikke korral. Käesolevas töös varundamise aspekte ei käsitleta.

3.5 Platvormi testimise protsess ja tulemused

Pärast platvormi juurutamist organisatsiooni privaatpilves viidi läbi turbe- ja jõudlustestid, et tagada platvormi töökindlus ja turvalisus. Esimeses etapis teostati OWASP ZAP (*Open Web Application Security Project Zed Attack Proxy*) tööriistaga platvormi veebilehel nõrkuseotsing, et tuvastada võimalikke turberiske, nõrkusi ja konfiguratsiooni vigu.

Skaneerimise tulemusena avastati mitmeid nõrkusi, nagu veebiserveri poolt lekkivad metaandmed, puuduvad CSP päised, mitmeid peidetud faile ning üks haavatav JavaScripti teek. Nõrkuste leidmisele järgnes nende kõrvaldamine. Enamik nõrkusi suudeti parandada Nginx konfiguratsiooni muutmisega, näiteks rakendades CSP-d ja keelates peidetud failide laadimise. Lisaks teavitati tarkvaratootjat haavatavast JavaScripti teegist.

Teises etapis viidi läbi veebilehe kiirus- ja stressitaluvusetestid Apache JMeter tööriista abil. Testimine hõlmas 50 kasutaja simuleerimist, kes sooritasid veebilehele pöördumisi 10 korda 100-sekundilise perioodi jooksul. Testimise ajal saavutati sujuv navigeerimine

ning veebilehe läbilaskevõime tulemuseks üle 500 päringu minutis, mis organisatsiooni siseselt kasutatava veebilehe jaoks on piisava varuga.

Lisaks integreeriti platvormi paigalduskoodi SNMP monitooringu agendi installatsioon, mis võimaldab reaalajas jälgida serveri ressursikasutust monitooringuplatvormil. Turbe-, stressitaluvuse- ja kiirustestide ajal täheldati platvormi ressursikasutust normi piires ning tavakasutusfaasis sujuvust ja stabiilsust.

Kokkuvõttes aitasid läbiviidud testid tagada platvormi turvalisuse, töökindluse ja jõudluse. Testimise tulemused võimaldasid tuvastada ja kõrvaldada platvormi nõrkused ning saavutada SOC-i jaoks platvormil rahuldava jõudluse.

4 Platvormi kasutuselevõtt ja tulemused

Pärast edukat platvormi juurutust, seadistamist ning katsetamist arendus- ja testkeskkonnas tarniti platvorm ka tootmiskeskkonda ning anti üle peakasutajale. Tootmiskeskkonna monitooringu tulemustel on platvorm täiskoormuse all väga hästi töötav ning platvormi veebileht suudab vastata päringutele kiiresti, isegi kui kogu meeskond ja automaatika sellel korraga töötavad.

Järgmise etapina hakkab platvormi peakasutaja koostöös SOC meeskonnaga platvormile looma juhtumimalle, mille põhjal saab turbeintsidente registreerida ja neile reageerida ning kuvada intsidentidega seotud statistikat. Täpsema intsidendihaldusstatistika saamiseks on vaja platvormi pikemalt kasutada. Kuid esmaseid hinnanguid saab anda sellegipoolest.

Uue platvormiga mõõdetakse keskse soorituse indikaatoreid, mida vanema platvormiga mõõta ei saanud. Uus platvorm lubab mõõta näiteks järgmiseid keskse soorituse indikaatoreid: tõese tuvastus (*True Positive*, TP) ja väärtuvastuse (*False Positive*, FP) suhe, keskmist tuvastus aega (*Mean Time to Detect*, MTTD), häire triaaži aega (*Alarm Time to Triage*, TTT), häire aega kvalifitseerumiseks (*Alarm Time to Qualify*, TTQ), keskmist kinnitus aega (*Mean Time to Acknowledge*, MTTA) ja keskmist lahendus aega (*Mean Time to Resolve*, MTTR)[38]. Uute mõõdikute abil saab intsidentidele reageerimist paremini prioriseerida. Samuti aitavad mõõdikud turbetööriistasid paremini platvormi konfigureerida, üritades minimeerida TP ja FP suhet.

Kui võtta aluseks näiteks nädala põhjal kogutud statistika, siis saab uuel platvormil MTTR tulemuseks 3,8 päeva. Mõõtes sama näitajat aasta varem, samal ajal vanast platvormist saame tulemuseks 8,4 päeva. See näitab, et uue platvormi kasutuselevõtt on organisatsioonis parandanud intsidentide lahendamise kiirust.

Kokkuvõttes on uue platvormi kasutuselevõtt võimaldanud SOC meeskonnal paremini hallata ja analüüsida turbeintsidente ning saada olulist statistikat, mida varasem süsteem ei võimaldanud. See aitab organisatsioonil paremini mõista oma tubevõimekust ja teha tõhusamaid otsuseid töövoogude parendamiseks.

4.1 Edasiarendused tulevikus

Esimene edasiarendus, mida kaaluda tuleks, on Cortex Neuronite lisamine TheHive 5 platvormi. See võimaldaks luua automatiseeritud töövooge ja integratsioone teiste andmerikastus- ning turbetööriistadega. Integreerides turbetööriistu, tuleks esmalt valida need tööriistad, mis suudavad edastada andmeid TheHive 5 platvormile ning genereerida teavitusi, nagu näiteks viirustõrje- ja tulemüürisüsteemid.

Järgmise arendusetapina tuleks täielikult rakendada platvormi automaatne konfigureerimine pärast paigaldust, sealhulgas integratsioonide lisamine ja peakasutajate loodud mallide integreerimine platvormiga.

Kubernetes klastri valmides tuleks juurutada objektsalvestusteenus näiteks CEPH või MinIO eraldiseisva Kubernetes instantsina, kuhu TheHive 5 saaks oma andmeid talletada, eemaldades objektsalvestusteenuse vajaduse teenusevirnast ning muutes teenust lihtsamaks. Seejärel tuleks kogu projekt Kubernetes konteineritele üle viia, tagamaks platvormi lihtsama halduse. Lõpuetapina võiks projekti ümber kujundada mitme sõlmpunktiliseks teenuseks, tagamaks teenuse skaleeritavuse ja kõrgkäideldavuse nõudluse kasvades.

Kokkuvõte

Käesoleva bakalaureusetöö tulemusel juurutati organisatsioonis SOC-i jaoks sobiv intsidendihaldus ja -reageerimisplatvorm StrangeBee TheHive 5, mis vastas SOC juhtkonna poolt sätestatud kriteeriumitele.

Platvormi juurutamiseks kasutati kaasaegseid süsteemihaldustööriistu ja -meetodeid. Loodi mugav ja kiiresti kasutatav testkeskkond, mis võimaldas hõlpsasti testida platvormi ja selle juurutamist. Organisatsiooni privaatpilves seati üles kolm eraldiseisvat, kuid homogeenset keskkonda: arenduskeskkond, proovikeskkond ja tootmiskeskkond. TheHive 5 platvorm seati üles organisatsiooni privaatpilves Jenkins-i tarneahela abil ning platvormile tehti esimesed integratsioonid teiste andmerikastustööriistadega, mis lõi aluse organisatsiooni SOAR süsteemi arendamisele.

Bakalaureusetöö tulemusena saavutati oluline samm SOC meeskonna jätkuvas arengus, mis aitab tõhustada intsidentidele reageerimist organisatsioonis. Uue platvormi kasutuselevõtt suurendab turbemeeskonna olukorrateadlikkust küberruumis ning parandab meeskonna tööefektiivsust.

Kasutatud kirjandus

[1] „What is SOAR (Security Orchestration, Automation and Response)? | Definition from TechTarget“, *Security*. https://www.techtarget.com/searchsecurity/definition/SOAR (vaadatud 2. aprill 2023).

[2] Palo Alto Networks, „The State of SOAR Report, 2020“, 2020. Vaadatud: 16. aprill 2023. [Online]. Available at: https://media.bitpipe.com/io_15x/io_154375/item_2268964/the-state-of-soar-report-2020.pdf

[3] „Chandramouli - 2022 - Guide to a Secure Enterprise Network Landscape.pdf“. Vaadatud: 16. aprill 2023. [Online]. Available at: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-215.pdf

[4] „iTop: open source ITIL ITSM CMDB Software“. https://www.combodo.com/itop-193 (vaadatud 1. aprill 2023).

[5] EVS-ISO/IEC, „EVS-ISO/IEC 27035:2012 Infotehnoloogia. Turbemeetodid. Infoturvaintsidentide haldus.“, *EVS-ISO/IEC 27035:2012*, 9. mai 2012. https://www.evs.ee/Download/ViewBrowsingServiceSubscription?productId=30443&language=EstonianLanguage (vaadatud 16. aprill 2023).

[6] „The Twelve-Factor App“. https://12factor.net/codebase (vaadatud 16. aprill 2023).

[7] M. Vielberth, F. Böhm, I. Fichtinger, ja G. Pernul, „Security Operations Center: A Systematic Study and Open Challenges“, *IEEE Access*, kd 8, lk 227756–227779, 2020, doi: 10.1109/ACCESS.2020.3045514.

[8] „Complete Guide to the PPT Framework | Smartsheet“. https://www.smartsheet.com/content/people-process-technology (vaadatud 18. aprill 2023).

[9] „What is a Security Operations Center (SOC)?“, *Security*. https://www.techtarget.com/searchsecurity/definition/Security-Operations-Center-SOC (vaadatud 1. aprill 2023).

[10] C. S. Générale, „IRM-2022 (Incident Response Methodologies 2022)“. 2. aprill 2023. Vaadatud: 3. aprill 2023. [Online]. Available at: https://github.com/certsocietegenerale/IRM

[11] C. S. Générale, „IRM-5-MaliciousNetworkBehaviour.pdf“. 2. aprill 2023. Vaadatud: 3. aprill 2023. [Online]. Available at: https://github.com/certsocietegenerale/IRM/blob/e7d33db72b7dc59c74429b0b1a1f9bc329b416b2/EN/IRM-5-MaliciousNetworkBehaviour.pdf

[12] DomainTools, „Building your IR Plan Step-by-Step“, *DomainTools | Start Here. Know Now.*, 14. mai 2020. https://www.domaintools.com/resources/blog/building-your-ir-plan-step-by-step/ (vaadatud 3. mai 2023).

[13] StrangeBee, „TheHive 5 - Overview“, *StrangeBee*. https://www.strangebee.com (vaadatud 18. aprill 2023).

[14] „Leveraging TheHive & Cortex for automated IR | SANS Institute“. https://www.sans.org/webcasts/leveraging-thehive-cortex-automated-ir-113265/ (vaadatud 1. mai 2023).

[15] „orchestration-of-csirt-tools-tools-admin.pdf“. Vaadatud: 1. mai 2023. [Online]. Available at: https://www.enisa.europa.eu/topics/training-and-exercises/trainings-for-cybersecurity-specialists/online-training-material/documents/orchestration-of-csirt-tools-1/orchestration-of-csirt-tools-tools-admin.pdf

[16] G. Inc, „IBM Security SOAR vs Splunk SOAR vs TheHive 2023 | Gartner Peer Insights“, *Gartner*. https://www.gartner.com/market/security-orchestration-automation-and-response-solutions/compare/product/ibm-security-soar-vs-splunk-soar-vs-thehive (vaadatud 19. aprill 2023).

[17] „IBM Security QRadar SOAR vs. Splunk SOAR (Security Orchestration, Automation and Response) vs. TheHive“, *G2*. https://www.g2.com/compare/ibm-security-qradar-soar-vs-splunk-soar-security-orchestration-automation-and-response-vs-thehive (vaadatud 19. aprill 2023).
[18] „IBM Security QRadar SOAR - Overview“, 14. märts 2023. https://www.ibm.com/products/qradar-soar (vaadatud 4. aprill 2023).
[19] „Splunk SOAR“, *Splunk*. https://www.splunk.com/en_us/products/splunk-security-orchestration-and-automation.html (vaadatud 19. aprill 2023).
[20] MISP, „MISP Open Source Threat Intelligence Platform & Open Standards For Threat Information Sharing“. https://www.misp-project.org/ (vaadatud 3. aprill 2023).
[21] Strangebee, „TheHive 5 Documentation“, *StrangeBee Docs*. https://docs.strangebee.com/ (vaadatud 3. aprill 2023).
[22] „MITRE ATT&CK®“. https://attack.mitre.org/ (vaadatud 4. aprill 2023).
[23] R. Al-Shaer, J. M. Spring, ja E. Christou, „Learning the Associations of MITRE ATT & CK Adversarial Techniques“, *2020 IEEE Conference on Communications and Network Security (CNS)*, juuni 2020, lk 1–9. doi: 10.1109/CNS48642.2020.9162207.
[24] „Cortex“. TheHive Project, 27. veebruar 2023. Vaadatud: 5. aprill 2023. [Online]. Available at: https://github.com/TheHive-Project/CortexDocs
[25] Strangebee, „TheHive Project Documentation“. https://docs.thehive-project.org/cortex/ (vaadatud 5. aprill 2023).
[26] „GitHub - TheHive-Project/Cortex-Analyzers: Cortex Analyzers Repository“. https://github.com/TheHive-Project/Cortex-Analyzers (vaadatud 5. aprill 2023).
[27] „What is Elasticsearch?“, *Elastic*. https://www.elastic.co/what-is/elasticsearch (vaadatud 1. aprill 2023).
[28] „Elasticsearch Tutorial“. https://www.tutorialspoint.com/elasticsearch/index.htm (vaadatud 1. aprill 2023).
[29] A. Lakshman ja P. Malik, „Cassandra: a decentralized structured storage system“, *ACM SIGOPS Oper. Syst. Rev.*, kd 44, nr 2, lk 35–40, apr 2010, doi: 10.1145/1773912.1773922.
[30] „Welcome to Apache Cassandra’s documentation! | Apache Cassandra Documentation“. https://cassandra.apache.org/doc/latest/index.html (vaadatud 1. aprill 2023).
[31] M. Inc, „MinIO | High Performance, Kubernetes Native Object Storage“, *MinIO*. https://min.io (vaadatud 1. aprill 2023).
[32] A. Wittig ja M. Wittig, *Amazon Web Services in Action*, 1st tr. USA: Manning Publications Co., 2015.
[33] I. Sacolick, „What is CI/CD? Continuous integration and continuous delivery explained“, *InfoWorld*, 15. aprill 2022. https://www.infoworld.com/article/3271126/what-is-cicd-continuous-integration-and-continuous-delivery-explained.html (vaadatud 6. aprill 2023).
[34] „Vagrant | HashiCorp Developer“, *Vagrant | HashiCorp Developer*. https://developer.hashicorp.com/vagrant (vaadatud 8. aprill 2023).
[35] „Microservices“, *martinfowler.com*. https://martinfowler.com/articles/microservices.html (vaadatud 6. aprill 2023).
[36] M. J. Scheepers, „Virtualization and Containerization of Application Infrastructure: A Comparison“, 2014, Vaadatud: 6. aprill 2023. [Online]. Available at: https://thijs.ai/papers/scheepers-virtualization-containerization.pdf
[37] „Forward Proxies and Reverse Proxies“. https://httpd.apache.org/docs/current/mod/mod_proxy.html (vaadatud 7. aprill 2023).
[38] „TheHive v5.1: Improved features“, *StrangeBee*, 1. märts 2023. https://blog.strangebee.com/thehive-v5-1-improved-features/ (vaadatud 23. aprill 2023).

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

Mina, Karl Rikkonen

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Automatiseeritud intsidentide reageerimisplatvormi juurutamine turbeoperatsioonide keskuses TheHive 5 näitel“, mille juhendajad on Siim Vene ja Karl Mendelman
   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.

10.05.2023

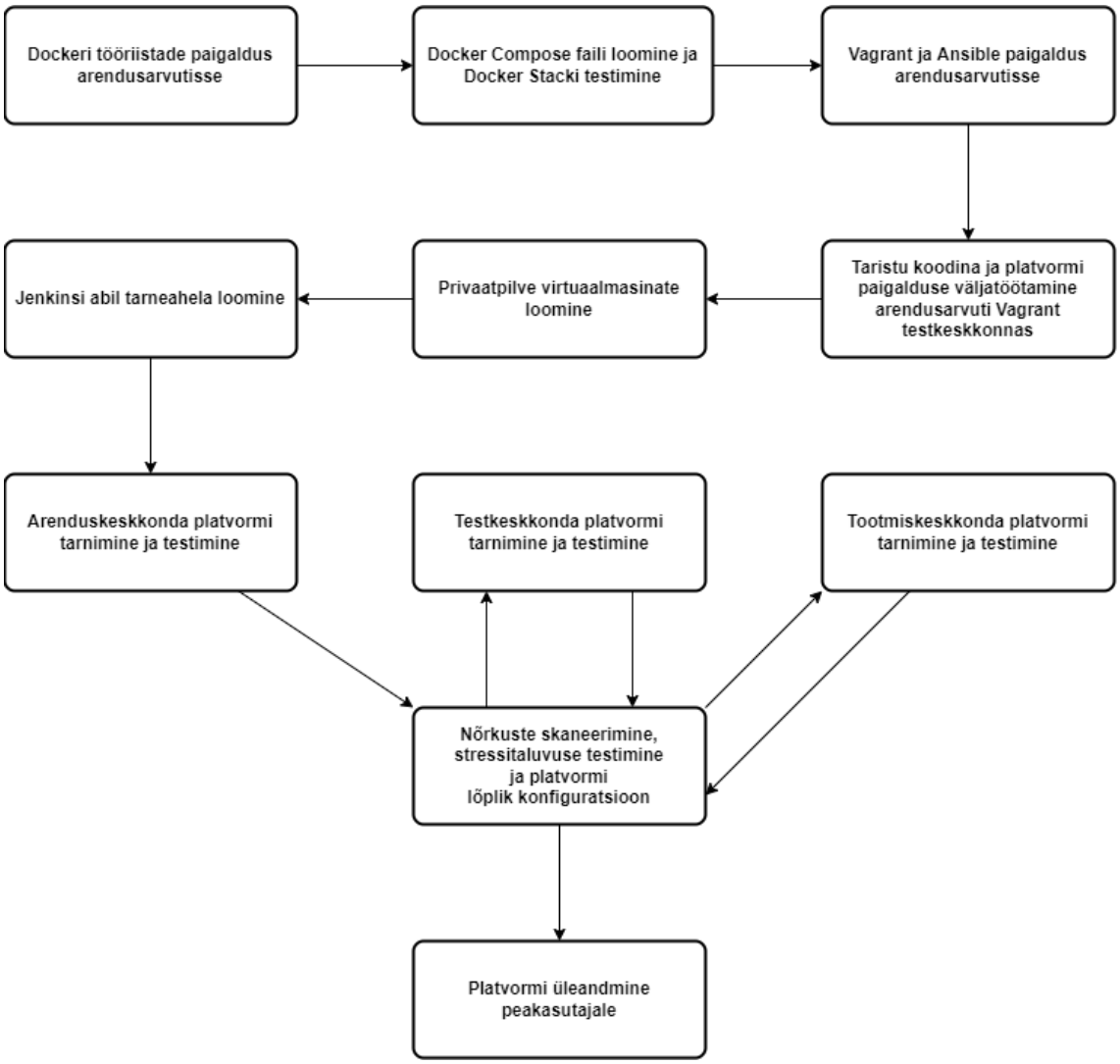
1 Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või 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 – IRP ja SOAR platvormide hinnang

Tabel 2. IRP ja SOAR platvormide hinnang.

Kriteeriumid punktiskaalal 0 -5 (0 pole võimalik/puudulik, 5 hea)	**Splunk SOAR**	**IBM QRadar SOAR**	**StrangeBee TheHive 5**
Lihtne kasutada ja kasutajasõbralik	4	3	5
Tugev kogukonna toetus	3	3	5
Turbetööriistadega integreerimise võimalus	4	4	4
Andmerikastustööriistadega integreerimise võimalus	5	4	5
Paindlike töövoogude loomine	5	4	4
Skaleeritavus	5	5	5
Agiilne platvormi arendus	4	3	5
Toetab konteineriseerimist	2	1	5
Toetab taristu koodina lähenemist	3	2	5
Toetab integratsiooni MISP-ga	3	3	5
TTP klassifikaatorite lisamine	4	4	5
Statistika kogumine ja armatuurlaudade kuvamine	5	4	4
Juhtumi loomine MISP sündmuse põhjal	3	3	5
Keskse soorituse mõõtmine ja kuvamine	4	4	5
Kiire juurutus võimalus	3	2	5
Mitme keskkonna paralleel juurutus	3	2	5
Avatud lähtekoodiga platvorm või platvormi osad	0	0	2
Pole vaja lisaarendusi	4	3	4
Töötab välisvõrguühenduseta	4	4	4
Freemium võimalus	0	0	5
Meeskonnal varasem kokkupuude platvormiga	2	2	4
Toetab simultaanteenindust (Multi-Tenancy)	4	4	4
Platvormi ressursikasutus	3	3	5
Punktisumma kokku	**77**	**67**	**105**

Lisa 3 – TheHive 5 juurutamise etappide diagramm



Joonis 3. Platvormi juurutamise etapid.

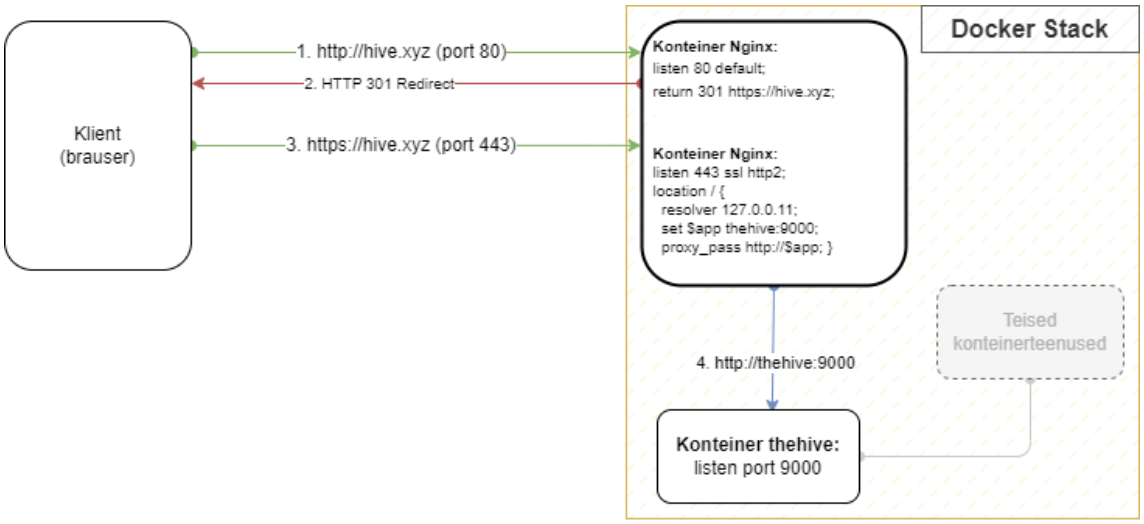
Lisa 4 – Koodi näide: Nginx konteineri kirjeldus Docker Compose failis

```
version: "3.6"
services:
  thehive: #...
  cassandra: #...
  elasticsearch: #...
  minio: #...
  nginx:
    image: xyz.internal/nginx:1.23.3-alpine
    networks:
      front-end:
    ports:
      - target: 80
        published: 80
      - target: 443
        published: 443
    configs:
      - source: nginx_config
        target: /etc/nginx/conf.d/default.conf
        mode: 0600
      - source: nginx_ssl_cert_config
        target: /etc/nginx/ssl/nginx.crt
        mode: 0600
    secrets:
      - source: nginx_ssl_key_secret
        target: /etc/nginx/ssl/nginx.key
        mode: 0600
configs:
  nginx_config:
    file: nginx.conf
  nginx_ssl_cert_config:
    file: nginx.cert
secrets:
  nginx_ssl_key_secret:
    file: nginx.key
networks:
  front-end:
  cassandra:
  elasticsearch:
volumes:
#...
```

Lisa 5 – Koodi näide: Nginx pöördproksi konfiguratsioon

```
server {

  listen 80 default;
  server_name _;
  server_tokens off;
  return 301 https://hive.xyz.internal/;
}

server {

  listen 443 ssl http2;
  server_name hive.xyz.internal;
  server_tokens off;

  ssl                 on;
  ssl_certificate     /etc/nginx/ssl/nginx.crt;
  ssl_certificate_key /etc/nginx/ssl/nginx.key;
  ssl_protocols TLSv1.3;

  set $CSP "default-src 'self'";
  #...
  #...
  add_header Content-Security-Policy $CSP always;
  #...

  location / {

    resolver 127.0.0.11;
    set $app thehive:9000;
    proxy_pass              http://$app;
    proxy_http_version      1.1;
    #...
    #...
    #...
  }
}
#...
```

Lisa 6 – Nginx tööpõhimõtte visuaalne ülevaade



Joonis 4. Nginx pöördproksi tööpõhimõte.

Lisa 7 – Koodi näide: Docker Stack-i käivitamis skript

```
#!/bin/bash

# Näide:
# docker stack deploy -c (Compose faili asukoht) (teenuse nimi)

docker stack deploy -c ./the_hive_5_compose.yml thehive5
```

Lisa 8 – Koodi näide: Vagrant tarkvara paigaldus Ubuntu 20.04-le

```
#!/bin/bash

sudo apt update
sudo apt install -y vagrant virtualbox
vagrant plugin install vagrant-vbguest
```

Lisa 9 – Koodi näide: Vagrantfile

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|

  config.vm.box = "generic/ubuntu2004"
  config.vm.network :private_network, ip: "192.168.x.x", name: "vboxnet0"

  config.vm.provider "virtualbox" do |prov|

    prov.memory = 8192
    prov.cpus = 2

  end

  $script = <<-SCRIPT
  sudo sed -i 's/us\.archive\.ubuntu\.../' /etc/apt/sources.list
  sudo sed -i 's/archive\.canonical\.../' /etc/apt/sources.list
  sudo sed -i 's/security\.ubuntu\..../' /etc/apt/sources.list
  echo "\
  [Resolve]
  DNS=192.168.x.x" | sudo tee /etc/systemd/resolved.conf
  sudo systemctl restart systemd-resolved
  SCRIPT

  config.vm.provision "shell", inline: $script

  config.vm.provision "ansible" do |ansible|

    ansible.playbook = "vagrant-playbook.yml"
    ansible.vault_password_file = ".vault"
    ansible.skip_tags = "not_vagrant"
    ansible.galaxy_role_file = "ansible-requirements.yml"

  end

end
```

Lisa 10 – Koodi näide: vagrant-playbook.yml

```
---

- import_playbook: playbook-host.yml

- import_playbook: playbook-app.yml
  vars:
    project_dns: hive.vagrant
    project_ip: "{{
hostvars[inventory_hostname]['ansible_default_ipv4']['address'] }}"
    nginx_ssl_crt: |
      -----BEGIN CERTIFICATE-----
      #...
    nginx_ssl_key: !vault |
          $ANSIBLE_VAULT;1.1;AES256
          30613531346638366534333434353864 3#...
```

Lisa 11 – Koodi näide: playbook-host.yml

```
---
- name: Configure unattended-upgrades
  hosts: all
  become: true
  vars:
    unattended_origins_patterns:
      - 'origin=Ubuntu,archive=${distro_codename}-security'
      - 'o=Ubuntu,a=${distro_codename}'
      - 'o=Ubuntu,a=${distro_codename}-updates'
      - 'o=Ubuntu,a=${distro_codename}-proposed-updates'
  roles:
    - unattended-upgrades

- name: Configure ntp
  become: true
  hosts: all
  vars:
    ntp_from_pkg: false
    timesync_ntp_hosts:
      - "192.168.x.x"
  roles:
    - ntp

- name: Configure snmp
  become: true
  hosts: all
  vars:
    snmp_user: user_name
    snmp_authtype: priv
    snmp_password:  !vault |
          $ANSIBLE_VAULT;1.1;AES256 #...
    snmp_encryption:  !vault |
          $ANSIBLE_VAULT;1.1;AES256 #...
    snmp_agentadress_adress:
      ipv4: "{{ ansible_default_ipv4.address }}
  roles:
    - snmp

- name: Configure Presistant Volumes for Server
  become: true
  hosts: all
  roles:
    - volume
```

Lisa 12 – Koodi näide: salvestuspinna loomise roll main.yml

```
# Create LVM
- name: Create a new primary partition for LVM
  parted:
    device: /dev/vdb
    number: 1
    flags: [ lvm ]
    state: present

# Create FS
- name: Create a ext4 filesystem on /dev/vdb1
  filesystem:
    fstype: ext4
    dev: /dev/vdb1

# Create Mount Folder
- name: Create a presistant volume mount dir (thehive_data)
  file:
    path: /thehive_data
    state: directory
    mode: '0755'

# Mount Disk to Folder
- name: Mount vdb1 to thehive_data
  mount:
    fstype: ext4
    src: /dev/vdb1
    path: /thehive_data
    state: mounted
```

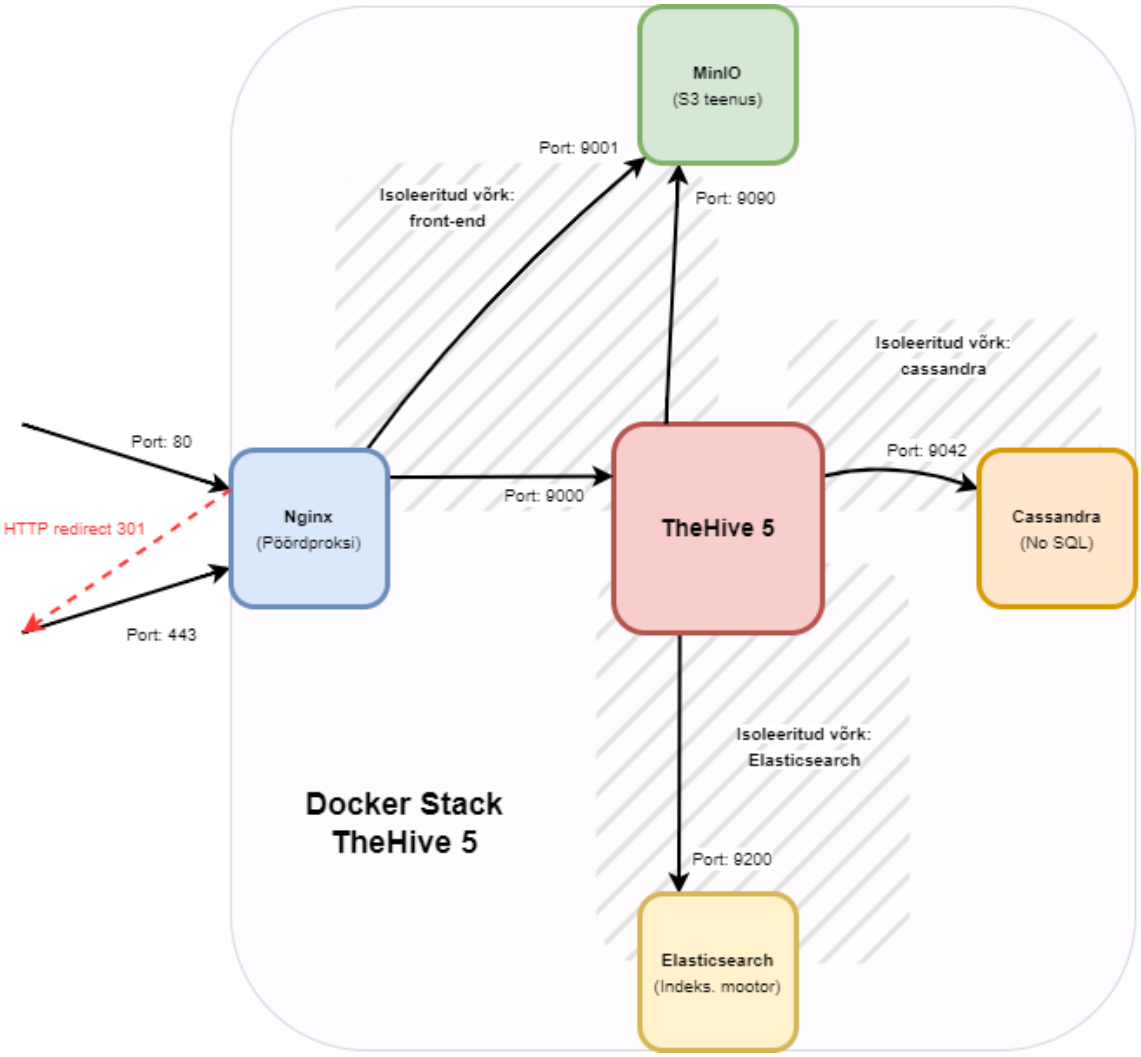
Lisa 13 – Koodi näide: playbook-app.yml

```
---

- name: Install and Configure Docker
  hosts: all
  become: true
  vars:
    docker_extra_apt_packages:
      - python3-docker
    docker_apt_ignore_key_error: true
    docker_apt_source: "http://xyz.internal/download.docker.com"
    docker_daemon_json_config:
      data-root: "/thehive_data/docker_volume"
      default-address-pools:
        - base: "192.168.x.0/24"
          size: 24
  roles:
    - docker

- name: Initialize Docker Swarm
  hosts: all
  become: true
  roles:
    - docker_swarm

- name: Compose and Deploy TheHive 5 on Docker Swarm
  hosts: all
  become: true
  vars_files:
    - ./vars/TheHive 5_vars.yml
    - ./vars/TheHive 5_compose.yml
    - ./vars/nginx_conf.yml
  roles:
    - TheHive 5
```

Lisa 14 – Koodi näide: Jenkinsfile

```
pipeline {
  agent any
  stages {
    stage('Setup requirements') {
        steps {
          echo "Installing Anisble Galaxy dependencies:"
          sh "ansible-galaxy install -f -r ansible-requirements.yml \
          -p roles/"
        }
    }
    stage('Configure Infrastructure') {
      steps {
        ansiColor('xterm') {
          ansiblePlaybook(
            credentialsId: "ssh-key-id",
            inventory: "./inventory/${env.INVENTORY_TAG}/hosts",
            playbook: "playbook-infrastructure.yml",
            vaultCredentialsId: "ansible-vault-id",
          )
        }
      }
    }
    stage('Configure Host') {
       #...
    }
    stage('Configure Application') {
       #...
    }
  }
  post {
    always {
      cleanWs()
    }
  }
}
```

Lisa 15 – Konteinerteenuste võrgudiagramm



Joonis 5. Konteinerteenuste võrgudiagramm.

Lisa 16 – IBM QRadar SOAR, Splunk SOAR ja TheHive 5 ressursinõuded.

Tabel 3. Platvormide ressursinõuded.

Ressurss	**IBM QRadar SOAR**	**Splunk SOAR**	**TheHive 5**
Operatiivmälu (Memory - RAM)	16Gb (minimaalselt) 32Gb (soovituslik)	16Gb (minimaalselt) 32Gb(soovituslik)	4Gb (Minimaalselt) 16Gb (soovituslik)
Salvestuspind (Disk Space)	200Gb (minimaalselt)	200Gb (minimaalselt)	100Gb (soovituslik)
Protsessori tuumad (CPU Cores)	4 tuuma (minimaalselt) 8 tuuma (soovituslik)	4 tuuma (minimaalselt) 8 tuuma (minimaalselt)	2 tuuma (minimaalselt) 6 tuuma (soovituslik)
Võrguühendus (Network)	1 Gbps (minimaalselt)	1 Gbps (minimaalselt)	100 Mbps (minimaalselt)
Operatsioonisüsteem (Operating System) Ja paigaldusvõimalus	RHEL, RHEL (OVA)	RHEL, CentOS, AWS Image	Linux-based OS Docker, Kubernetes