TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Mattias Raba 206700IAIB

SISSETUNGIDE TUVASTAMISE JA TEAVITAMISE SÜSTEEM PAHATAHTLIKE VÕRGUPÄRINGUTE SEIREKS

Bakalaureusetöö

Juhendaja: Toomas Lepik
Magistrikraad

Tallinn 2025

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: Mattias Raba

15.01.2025

Annotatsioon

Sissetungide tuvastamise ja teavitamise süsteem pahatahtlike võrgupäringute seireks

Käesolev bakalaureusetöö käsitleb sisevõrgu turvamonitooringu lahenduse loomist, keskendudes stsenaariumile, kus ründaja on kompromiteerinud sisevõrgus oleva tööjaama või paigaldanud sinna enda kontrollitud seadme. Töö eesmärgiks on välja töötada kergelt paigaldatav ja efektiivne sissetungi tuvastamise ja teavitamise süsteem, mis avastab ebatavalist või pahatahtlikku tegevust võrguliikluses, tuvastades erinevaid võrgu kaardistamise ja ründamise tehnikaid. Lisaks luuakse integreerimisvõimalused väliste suhtlusplatvormidega nagu Slack, Microsoft Teams ja Mattermost, võimaldamaks reaalajas teavituste edastamist.

Töö jaguneb kaheks komponendiks: võrgu liiklust monitooriv agent, mis luuakse Go programmeerimiskeeles, ja Django-põhine veebirakendus, mis vastutab andmete haldamise ja kasutajaliidese eest. Süsteemi töökindlust valideeritakse testkeskkonnas, kasutades avatud lähtekoodiga ründesimulatsiooniraamistikku Atomic Red Team. Töös valideeritakse valminud lahenduse funktsionaalsust, mida võrreldakse teiste vabavaraliste sissetungi tuvastamise süsteemidega, nagu Suricata, Snort ja Zeek.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 23 leheküljel, 6 peatükki, 6 joonist, 1 tabelit.

Abstract

Intrusion Detection and Notification System for Monitoring Malicious Network Requests

This bachelor's thesis focuses on creating a security monitoring solution for an internal network, with a specific emphasis on a scenario where an attacker has compromised a workstation within the network or installed a device under their control there. The goal of the work is to develop an easy-to-install and effective intrusion detection and notification system that detects unusual or malicious activity in network traffic by identifying various network mapping and attack techniques. In addition, integration options are created with external communication platforms, such as Slack, Microsoft Teams and Mattermost, to enable real-time notifications.

The work is divided into two components: a network traffic monitoring agent, which is created in the Go programming language, and a Django based web application, which is responsible for data management and the user interface. The reliability of the system was validated in a test environment using the open-source attack simulation framework Atomic Red Team. The work validates the functionality of the completed solution, which is compared to other open source intrusion detection systems, such as Suricata, Snort, and Zeek.

The thesis is written in Estonian and is 23 pages long, including 6 chapters, 6 figures and 1 table.

Lühendite ja mõistete sõnastik

ARP	*Address Resolution Protocol*, aadressiteisenduse protokoll
ATT&CK	*Adversarial Tactics, Techniques, and Common Knowledge*, ründemeetodite ja taktikate küberjulgeoleku raamistik
CI/CD	*Continuous Integration and Continuous Development*, pidev integratsioon ja pidev tarne
CISA	*Cybersecurity and Infrastructure Security Agency*
DHCP	*Dynamic Host Configuration Protocol*, standardprotokoll, mis võimaldab arvutitel automaatselt saada TCP/IP-võrgus vajalikke parameetreid
DNS	*Domain Name System*, domeeninimede süsteem
EDR	*Endpoint Detection and Response*, otspunkti ohuavastus ja reageerimine
HIDS	*Host-based Intrusion Detection System*, seadmepõhine sissetungi tuvastuse süsteem
HTML	*Hyper Text Markup Language*, veebilehtede märgendkeel
IDS	*Intrusion Detection System*, sissetungi tuvastuse süsteem
IPS	*Intrusion Prevention System*, sissetungi ennetamise süsteem
LDAP	*Lightweight Directory Access Protocol*, teabekataloogide kasutamise ja halduse protokollistik
NIDS	*Network-based Intrusion Detection System*, võrgupõhine sissetungi tuvastuse süsteem
RDP	*Remote Desktop Protocol*, kaugjuurdepääsu protokoll
SMB	*Server Message Block*, rakenduskihi protokoll ja sõnumivorming failide, kataloogide ja seadmete ühiskasutuseks võrgus
SSH	*Secure Shell*, kaugjuurdepääsu krüptograafiline võrguprotokoll
Tor	*The Onion Router*, anonüümne võrk
WinRM	*Windows Remote Management*, Windows kaughaldustööriist

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Viidates Stastica.com [1] kogutud andmetele, teatati Ameerika Ühendriikides 2022. aastal ligikaudu 480 000 küberrünnaku juhtumist, mis näitab järjepidevat kasvu alates 2016. aastast, mil registreeriti umbes 250 000 juhtumit. Märkimisväärne hüpe toimus 2020. aastal, kui registreeriti üle 540 000 juhtumi. Eesti kontekstis toob Riigi Infosüsteemi Amet oma 2024. aasta küberturvalisuse aastaraamatus [2] välja, et mõjuga küberrünnakute (rünnakud, millega kaasnes andmeleke või sissemurdmine) arv on kasvanud 2237-lt aastal 2021 kuni 3314-ni aastal 2023.

Tänapäeva küberohud kasvavad kiiresti nii oma ulatuselt kui ka keerukuselt, mistõttu on organisatsioonide jaoks üha olulisem rakendada kriitiliste süsteemide ja andmete kaitsmiseks sügavuti kaitse (ingl k. *defence-in-depth*) strateegiat [3]. Sügavuti kaitse põhimõte seisneb mitmekihilise kaitse loomises, pakkudes rünnaku leevendamist isegi mõne kaitsekihi kompromiteerimisel. Selline lähenemine tunnistab, et ükski turvameede pole veatu, mistõttu erinevate kaitsemehhanismide kombineerimine suurendab oluliselt üldist turvalisust [4].

Sügavuti kaitse strateegia moodustab tervikliku lähenemise, mis ühendab endas mitmeid turvameetmeid erinevatel tasanditel, mis üldjuhul algab organisatsiooni võrgu ja selle perimeetri kaitsmisega [5]. Tulemüürid ja virtuaalsed privaatvõrgud reguleerivad admevoogude liikumist sisevõrgu ja interneti vahel. Lisaks kasutatakse sissetungimise tuvastamise ja ennetamise süsteeme, mis avastavad ja blokeerivad ebatavalised tegevused sisevõrgus. Samuti mängib olulist rolli lõpp-punktide ja seadmete kaitse, mille alla kuuluvad viirusetõrjetarkvarad ja EDR-süsteemid. Heuristilise analüüsi ja käitumismustrite monitooringu abil on võimalik tuvastada mittestandardseid tegevusi, mis võivad viitada pahavarale või sissemurdmisele [6].

Rakenduste haavatavuse ja andmete kaitseks kasutatakse automatiseeritud uuenduste rakendamist, teadaolevate turvaaukude eemaldamist, krüpteerimist ja andmekao ennetamise süsteeme. Oluline on ka juurdepääsukontroll, sealhulgas rollipõhine ligipääsukontroll ja mitmefaktoriline autentimine volitamata ligipääsu piiramiseks.

Kõigele lisaks on oluline ka kasutajate koolitamine, vähendamaks suhtlusrünnakute riski, hõlmates endas andmepüügikirjade äratundmise taseme tõstmist ja oskust delikaatselt ümber käia meilides varitsevate tundmatute manustega. Füüsiliste, tehniliste ja administra-

tiivsete kontrollide kombineerimisega vähendab kihiline kaitse tõenäosust, et üksainus haavatavus või turvaauk põhjustab märkimisväärset kahju.

Siiski, kui kõik eelnevalt loetletud kaitsemeetmed ebaõnnestuvad ja pahatahtlik osapool suudab kompromiteerida sisevõrgus oleva tööjaama, paigaldada lokaalvõrku enda kontrollitud seadme või saada ligipääsu sisevõrgule läbi WiFi, muutub võrguturbe monitooring antud olukorras üheks peamiseks kaitsemeetmeks. Vastavalt IBM poolt 2024. aastal koostatud raportile [7], võib lokaalvõrgu lekke tuvastamine ja isoleerimine võtta aega keskmiselt 258 päeva. Kasvutrendis olevate küberrünnakute taustal peab isegi hästi kaitstud sisevõrgus olema võrguturbe monitooringu komponent, pakkudes turvahoiatuste pidevat järelevalvet ja võimaldades organisatsioonidel proaktiivselt reageerida võimalikele intsidentidele enne nende eskaleerumist.

1.1 Eesmärk

Käesolev bakalaureusetöö käsitleb stsenaariumi, milles ründaja on kompromiteerinud ühe sisevõrgus oleva tööjaama, paigaldanud lokaalvõrku enda kontrollitud seadme või saanud sisevõrgule ligipääsu läbi WiFi. Antud lõputöö fookuses on kergelt paigaldava sissetungi tuvastamise ja teavitamise süsteemi loomine, avastamaks ebatavalist või pahatahtlikku tegevust sisevõrgu liikluses.

1.2 Eelnevad tööd teema kohta

Sissetungi tuvastuse ja ennetamise süsteemide kohta leidub eelnevalt kirjutatud lõputöid. Hetkel tuntuid IDS-e analüüsis B. Adam, tuvastamaks sobivaimat monitooringusüsteemi merenduses kasutatavate infosüsteemide turvalisuse tagamiseks, oma bakalaureusetöös "Vabatarkvaraliste ründetuvastussüsteemide jõudlus- ja sobivusanalüüs eriotstarbelistes võrkudes" [8]. Töös keskenduti peamiselt iga vaadeldava IDS-i kiiruse ja ressursi kasutamise võrdlusele, jättes kõrvale sissetungimiste tuvastamise võimekuse, millele just käesolev lõputöö oma rõhuasetuse seab.

Samuti analüüsis eelenvalt nimetatuid IDS-e S. Farooghian oma bakalaureusetöös "Praktiliste oskuste töötuba – avatud lähtekoodiga sissetungimise turvasüsteemid" [9], käsitledes sissetungi tuvastuse süsteemide paigaldust, konfigureerimist ja tuvastust Suricata, Snort ja Zeek platvormide näitel. Käesolevas lõputöös arendatakse kohandatud lahendust, mis oleks võimeline tuvastama ründaja spetsiifilisemat käitumismustrit. Eesmärk pole logida kõike võrgus toimuvat, vaid ainult pahatahtlikku tegevust.

Lisaks teostas sarnaste IDS-ide vahel võrdlust E. Paas oma bakalaureusetöös “Võrguliikluse seire ja sissetungi tuvastuse süsteemi juurutamine ja analüüs S4A baasil” [10], kus ta võrdles süvitsi kolme erinevat sissetungi tuvastuse süsteemi, leidmaks parimat lahendust oma lõputöös käsitleva riigiasutuse jaoks. Sobivaimaks lahenduseks osutus Suricata for All, mis juurutati edukalt vastavalt väljatoodud spetsifikatsioonidele. Käesoleva lahenduse kasutuselevõtu töövaev peab oleme minimaalne, eristudes sedasi näiteks Snortist ja Zeekist, mille puhul tuleb ammendava tulemuse saamiseks antuid IDS-e põhjalikult seadistada.

2 Analüüs

Käesolevas peatükis antakse ülevaade loodava lahenduse poolt tuvastatavatest võrgu kaardistamise ja ründamise tehnikatest, süsteemile seatavatest nõuetest ja hetkel olemasolevatest lahendustest.

2.1 Kasutatavad ründemeetodid

Kasutatavate ründemeetodite valikul tugineti MITRE Corporation'i poolt pakutavale ATT&CK maatriksile [11] ja CISA 2023. aasta riskide ja haavatavuste hindamise raportile [12]. Kokku koondati potensiaalselt suurima tulemuslikkusega meetmed, moodustades sedasi realistliku ja mitmekesise ründestsenaariumi kogumi.

2.1.1 DNS päringud

DNS päringud võimaldavad ründajal kasutada mitmeid võrgu loendamise ja kaardistamise tehnikaid:

- TXT päringuid saab kasutada andmete eksfiltreerimiseks, kodeerides neisse tundlikku teavet, näiteks paroole või serveri konfiguratsioone, saates need sedasi välisele DNS või C2 serverile, mida pahatahtlik isik kontrollib [13].
- PTR päringuid kasutades, on ründajal võimalik teisendada sisevõrgu IP-aadressid nimedeks ja kaardistada sisevõrku, kogumaks sedasi infot võrguseadmete kohta. Sõltuvalt kasutatava pordiskaneerija tüübist, võib valitud tööriist teostada enne võrguseadme skanneerimist samasuguse PTR päringu, mille üheks selliseks näiteks on võrguhalduses laialdaselt kasutatav tööriist Nmap.
- Samuti võib pahatahtlik isik kasutada sõnastikku, domeeni A kirjete leidmiseks [14].

2.1.2 Portide skaneerimine

Portide skaneerimine on üks tavalisemaid meetodeid, mida ründaja saab kasutada võrgu kaardistamise või potentsiaalselt haavatavate ründepindade leidmise eesmärgil [15].

2.1.3 DHCP päringud

Ründajal on võimalik kasutada *DHCP Starvation Attack* meetodit [16]–[17], mille tulemusel ei saa ülejäänud legitiimsed masinad IP-aadresside puudumisel võrku kasutada. Lisaks võib pahatahtlik osapool uuendada oma MAC aadressi regulaarselt (mida vaikimisi teevad iOS ja Kali Linux operatsioonisüsteemid). Samuti võib pahatahtlik isik seadistada võrgus võlts DHCP serveri, mis petab seadmeid vastu võtma pahatahtlikke võrguseadeid (valesid DNS servereid või *gateway*'sid), suunamaks seadmete liikluse läbi ründaja kontrolli all oleva seadme. DHCP päringute analüüsimisel võib leida võrku teadaolevalt mittekuuluvaid masinaid, rikkudes sedasi sisevõrgule omanäolist mustrit.

2.1.4 Kasutajaagent

Pahatahtlik isik võib operatsiooniturvele tähelepanu pööramata jättes, kasutada rünnaku teostamisel tööriistu, mis on seostatavad teadaolevate kasutajaagentidega, nagu nmap, sqlmap, feroxbuster jne. Mittestandardsete või tuntud halbade kasutajaagentide päiste jälgimine võib aidata selliseid tegevusi tuvastada [18].

2.1.5 SMB päringud

Ründaja võib erinevate meetoditega sundida kohtvõrgu arvuteid pöörduma ründaja kontrollitud välisesse serverisse, kus Responder-nimelise tarkvaraga püütakse ohvri parooli räsi kätta saada, mida on hiljem võimalik murda [19].

2.1.6 LDAP päringud

Kui võrgus on kasutusel Windowsi domeenikontroller, siis võimaldab LDAP protokoll ründajal kõiki domeeniobjekte (kasutajad, grupid, arvutid) pärida, näiteks kasutades selleks tarkvara Bloodhound [20]–[21].

2.1.7 Administraatori protokollid

Ründaja saab kasutada SSH, SMB, RDP või WinRM protokolle, liikumaks tööjaamade vahel. Kui on teada, et antud ühendused algavad alati administraatori kasutatavast masinast, siis on võimalik selle alusel tuvastada ründaja poolt algatatuid ühendusi.

2.1.8 Tor-võrk

Tor-võrgu kasutamine ei pruugi alati viidata pahatahtlikule tegevusele, kuid leidub tuntuid pahavarasid (WhiteSnake [22], Raspberry Robin ja Agent Tesla), mis suhtlevad juhtserveriga läbi Tor-võrgu. Sibulavõrgu liikluse tuvastamine on kerge viis tuvastamaks ebatavalist käitumist sisevõrgus.

2.1.9 ARP päringud

Ründajal on võimalik kasutada ARP päringuid võrgus olevate seadmete leidmiseks ja tuvastamiseks. Arp-scan tööriist on saadaval Debian-põhistes Linuxi distributsioonides.

2.2 Funktsionaalsed nõuded

Funktsionaalsed nõuded on süsteemi kirjeldavad omadused või funktsioonid, mida arendajad peavad rakendama, võimaldaks süsteemi kasutajatel oma ülesandeid täita [23]. Üldiselt kirjeldavad funktsionaalsed nõuded süsteemi käitumist konkreetsetes tingimustes. Enimkasutatavate võrguründe tehnikate ja võrgu kaardistamise meetodite analüüsi käigus püstitati järgnevad funktsionaalsed nõuded:

- Kasutaja saab kuvada tuvastatuid turvahoiatusi.
- Vastavalt seadistustele edastab rakendus hoiatused kasutaja suhtlusplatvormidele.
- Rakendus tuvastab erinevat tüüpi DNS päringuid, kui nende hulk ületab künnise (TXT, PTR, A ja AAAA kirjed).
- Rakendus tuvastab võrgus tehtava portide skaneerimise.
- Rakendus oskab analüüsida DHCP päringuid ja tuvastab võrku lisatud petliku DHCP serveri. Lisaks tuvastab seadme MAC aadressi ja hostinime vahelise seose muutumist.
- Rakendus tuvastab kahtlase kasutajaagendi päise (Kali, Parrot, Raspberrypi jne).
- Rakendus tuvastab ebatavalise või kohtvõrgust väljuva SMB päringu.
- Rakendus tuvastab LDAP päringute kaudu tehtavat domeeni kaardistamist.
- Rakendus tuvastab Tor-võrgu suunas tehtava päringu.
- Rakendus tuvastab ARP päringute kaudu tehtava võrguluure.
- Rakendus tuvastab meepurgi pihta tehtava päringu.

2.3 Mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded viitavad süsteemi kvaliteediatribuutidele, kuidas süsteem peab käituma, täpsustades tarkvaranõuete kriteeriume, mille alusel hinnatakse süsteemi töökäiku. Järgnevalt on välja toodud süsteemi mittefunktsionaalsed nõuded:

- Süsteem peab olema projekteeritud nii, et seda saaks tulevikus hõlpsalt laiendada uute funktsionaalsustega.
- Süsteemi uuenduste tegemisel tuleb tagada andmete terviklikkus.
- Süsteem peab olema kaitstud volitamata ligipääsu, andmete lekkimise ja pahatahtlike rünnakute eest.
- Süsteemi ülesseadmine, seadistamine ja haldamine peab olema kergesti mõistetav ja kasutjasõbralik.

2.4 Olemasolevad lahendused

IDS ehk sissetungi tuvastuse süsteem on tehnoloogiline lahendus, mis võimaldab tuvastada ja analüüsida ebatavalist või pahatahtlikku tegevust nii arvutivõrgus kui ka sinna kuuluvates tööjaamades. IDS lahendused jagunevad üldiselt kahte gruppi: võrgupõhised sissetungi tuvastuse süsteemid (NIDS) ja seadmepõhised sissetungi tuvastuse süsteemid (HIDS) [24]. HIDS keskendub üksikute seadmete tegevuste jälgimisele, analüüsides näiteks süsteemilogisid, failide terviklikkust ja kasutajate tegevusi, et avastada pahatahlikku käitumist, märke volitamata juurdepääsust või andmete muutmisest. HIDS ei pruugi alati väljastada reaalajas hoiatusi, sest antud süsteemi töökäik põhineb sageli logifailide hetktõmmiste võrdlemisel, mis võib viia viivitusteni sõltuvalt nende suurustest ja analüüsi sagedusest. NIDS seevastu analüüsib võrguliiklust, kasutades mustri- ja anomaaliapõhiseid meetodeid, et avastada võrgutasemel ringlevaid ohtusid. Selline tsentraalne võrgupõhine lähenemine võimaldab kiiresti tuvastada ja hoiatada ebatavaliste andmepakketide mustrite eest, mis võivad viidata rünnakule. Kuna antud bakalaureusetöös käsitletakse võrgupõhiseid sissetungi tuvastamise meetmeid, siis uuris autor just sellise võimekusega olemasolevaid lahendusi.

2.4.1 Snort

Snort, mida hetkel omab Cisco, on üks maailma kõige laialdasemalt kasutatavaid avatud lähtekoodiga sissetungi tuvastamise ja ennetamise süsteeme, mida tuntakse oma paindlikkuse ja tõhususe poolest reaalajas liikluse analüüsimisel ja pakettide kontrollimisel. Selle tugevusteks on kohandatavad reeglikogumid, lai protokollide

toetus ja võime tuvastada ning blokeerida mitmesuguseid rünnakuid, mistõttu on see sobilik paljudesse võrgukeskkondadesse. Snorti kasutatakse peamiselt keskmise suurusega ja suurtes ettevõtetes, mis vajavad kohandatavat ja paindlikku monitooringulahendust. Oma laiapindsest kasutusvõimalustest tulenevalt võib konfigureerimine osutuda keeruliseks [25].

2.4.2 Suricata

Open Information Security Foundation'i poolt arendatud Suricata on avatud lähtekoodiga IDS/IPS lahendus, mis pakub tänu lõimtöötlusele (ingl k. *multi-threaded processing*) võimekat liikluse mahtu, toetades samal ajal paljusid protokolle. Lisaks arenenud tuvastusele on Suricata tugevuseks võime teha süvitsi minevat analüüsi, näiteks HTTP/2 ja TLS-dekrüpteerimist. Kuigi Suricata on kohandatud läbi töötlema suurel hulgal võrgupakette, võib valesti valitud riistvara korral tekkida pudelikael. Suricata on ideaalne organisatsioonidele, mis vajavad suure kiirusega võrguliikluse detailset uurimist [26].

2.4.3 Zeek

Zeek on passiivne, avatud lähtekoodiga võrguliikluse analüüsija, mida peamiselt kasutatakse võrgu turvamonitorina, kuid mis toetab ka süsteemi jõudluse analüüsimist ja rikkeotsingut. Zeeki peamine eelis on ulatuslik logide kogum, mis kajastab võrguaktiivsust ning HTTP ja DNS päringuid. Logid on struktureeritud JSON-failidena, sobides selliselt andmete töötlemiseks või välissüsteemidesse suunamiseks. Zeek on optimeeritud võrguliikluse tõlgendamiseks ja logide genereerimiseks, mitte baitide vastavuse kontrollimiseks või protokollide analüüsimiseks, pakkudes efektiivset ja kompaktset lahendust võrguanalüüsiks [27].

2.4.4 Cisco Secure Firewall

Cisco Secure Firewall on võrgu baasil toimiv sissetungi tuvastamise süsteem, mille IDS ja IPS lahendused kaitsevad süsteeme pahatahtliku liikluse eest, pakkudes mitmekihilist võrgu turvalisust. Cisco Secure Firewall pakub ulatuslikku kaitset, sealhulgas kohandatavaid reegleid, sissetungi tuvastamist, VPN-tuge ja täiustatud liikluse analüüsi, integreerides sedasi sujuvalt teiste Cisco turvalahendustega. Kuigi süsteem pakub tugevaid turvafunktsioone ja laialdast protokolli tuge, on see sageli kallis ja nõuab koolitatud personali selle seadistamiseks ning haldamiseks, mis ei pruugi väiksemates organisatsioonides võimalik olla. Cisco Secure Firewall on laialdaselt kasutusel suurtes ettevõtetes, pakkudes tipptasemel kaitset keerukates võrgukeskkondades [28].

2.4.5 Check Point

Check Point pakub täiustatud turvalahendusi, sealhulgas tulemüüri seadistamist, VPN-e, sissetungi tuvastamise ja ennetamise meetmeid. Selle tugevusteks on ulatuslikud turvameetmed, kõrge jõudlus, täiustatud ohtude tuvastamine ja tsentraliseeritud halduse võimalused, mis muudavad selle ideaalseks suuremahulistes keskkondades. Siiski võivad selle lahendused olla kulukad ja nende seadistamine aeganõudev, eriti suurtes infrastruktuurides, kus on keerulised turvanõuded. Check Point Software Technologies tooted sobivad suurtele ettevõtetele ja teenusepakkujatele, kes vajavad ulatuslikku kaitset kogu oma võrgu infrastruktuuris [29].

3 Kasutatavad tehnoloogiad

Järgnevates alapeatükkides kirjeldatakse tehnoloogiaid, mis valiti loodava süsteemi arendamiseks.

3.1 Django

Django on vabavaraline raamistik [30], mida saab kasutada veebirakenduste kiireks ja tõhusaks arendamiseks. Enamikud veebirakendused koondavad endas ühiseid funktsioone nagu autentimine, andmete pärimine andmebaasist, kasutajate ja küpsiste haldamine. Vältimaks sarnaste funktsionaalsuste igakordset uuesti kirjutamist veebrirakenduse jaoks, muudab Django arendaja töö lihtsamaks, koondades erinevad funktsioonid taaskasutatavate moodulite kogumiks, mida nimetatakse veebirakenduse raamistikuks. Django üheks tugevaimaks küljeks on tema mudelid, mis toimivad andmebaasi ja serveri koodi vaheliste liidestena, teisendades andmebaasitabelid klassideks või objektideks Pythoni koodis – seda protsessi nimetatakse objekt-relatsiooniliseks peegeldamiseks (ingl k. *object-relational mapping*). Arendajad kasutavad Django veebiraamistikku, eemärgiga oma koodi efektiivsemalt kirjutada ja seda paremini organiseerida, vähendades selliselt oluliselt veebirakenduse loomiseks kuluvat aega.

3.2 Go

Go, tuntud ka kui Golang, on Google'i poolt loodud staatiliselt tüübitud, kompileeritav programmeerimiskeel, mille peamisteks tugevusteks on kerge loetavus ja lihtne paralleeltöötlus [31]. Go kompileeritakse otse masinkoodiks, pakkudes sedasi koodi kiiremat täitmist võrreldes Pythoniga, mis kõigepealt genereeritakse baitkoodiks, mida Python Virtual Machine seejärel rea kaupa tõlgendab. Go on sobiv valik võrgurakenduste arendamiseks, kuna ta pakub unikaalset kombinatsiooni omadustest, mis on kohandatud tänapäeva võrgu programmeerimise vajadustele. Go kergekaalulised protsessid (ingl k. *goroutines*) ja sisseehitatud paralleelsuse primitiivid - *Channels, Mutexes* ja *Waitgroups* - muudavad keele erakordselt tõhusaks tuhandete samaaegsete ühenduste haldamisel, mis on võrgurakendustes oluline nõue. Go ulatuslik teekide kogum pakub hulgalist võrgutööriistada kogumit, lihtsustamaks tavapäraseid ülesandeid nagu serverite loomine, HTTP päringute käsitlemine ja madala taseme võrguoperatsioonide teostamine.

3.3 Docker

Docker on tehnoloogia, mis võimaldab arendajatel pakendada rakendusi koos nende vajaminevate teekidega kaasaskantavatesse konteineritesse. Konteineriseerimine tagab, et rakendused käituvad igas keskkonnas identselt, muutes rakenduste arendamise, testimise ja skaleerimise lihtsamaks. Docker on eriti kasulik mikroteenuste arhitektuuride haldamiseks, võimaldades tõhusat ressursikasutust ja kiiret juurutamist, muutes selle suurepäraseks valikuks kaasaegsete rakenduste arendamisel [32].

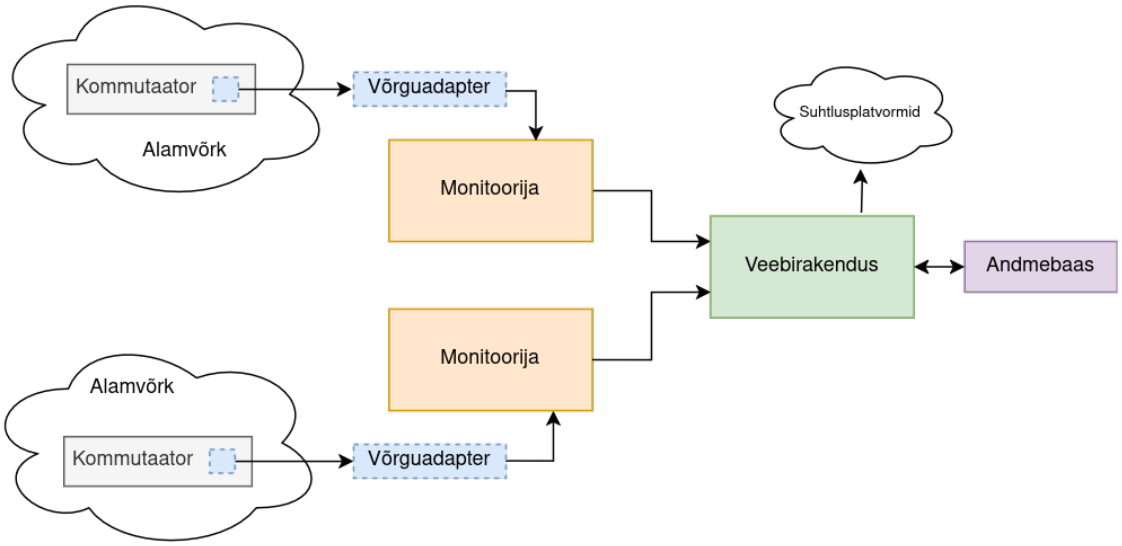
4 Rakenduse arhitektuur

Süsteemi arhitektuur viitab kontseptuaalsele raamistikule, mis määratleb süsteemi struktuuri, käitumise ja olulisemad komponendid, sealhulgas nendevahelised suhted ja integreerimise, eesmärgiga täita süsteemi nõudeid [33]. Antud bakalaureusetöö on jaotatud kaheks süsteemiks: võrguliikluse monitoorimise eest vastutav komponent ehk agent ja Django-põhine veebirakendus, mis hõlmab endas andmebaasihaldust ja kasutajaliidest.

4.1 Süsteemi loogiline ülesehitus

Võrgupõhise sissetungi tuvastamise süsteemi füüsiliseks paigaldamiseks kohtvõrgus on mitmeid valikuid, pidades silmas strateegilisi kohti võrgu infrastruktuuris. Jättes kõrvale võimaluse paigutada NIDS võrgu tulemüürist ettepoole, on selle üks levinuim paigutuskoht tulemüüri taga. Kui kuulata ainult tulemüüri pordi liiklust, siis oleks süsteemile nähtav ainult internetist tulev ja sinna suunas liikuv võrguliiklus. Kuid soovime kuulata ka sisevõrgus toimuvat liiklust, mille jaoks on vajalik, et kommutaatori kõikide siseportide liiklus oleks peegeldatud suurema läbilaske võimega porti. Sedasi käitudes, on loodava lahenduse monitooringu komponendil võimalus keskenduda potensiaalselt kompromiteeritud sisevõrgu masinate tuvastamisele. Samuti võib suurema infrastruktuuri korral paigaldada mitmeid NIDS'e võrgu erinevatesse segmentidesse, näiteks perimeetervõrgu (ingl k. *demilitarized zone*) otspunkti, või kriitiliste serverite lähedusse, pakkudes sedasi täiendavat paindlikkust ja ülevaadet võrgus toimuvast. Kokkuvõtvalt tuleb NIDS'i paigutust kohtvõrgus hoolikalt kaaluda vastavalt võrgu turvavajadustele ja asutuse ressursipiirangutele, tagamaks võimalikult tõhusa ja täpse turvamonitooringu.

Käesoleva tervikliku töö raames edastab iga üles seatud agent monitooringu käigus tuvastatud teavitused Django veebirakendusele. Veebirakendus salvestab iga juhtumi andmebaasi, peale mida kontrollitakse kas antud hoiatust tuleb kasutaja poolt paika pandud reeglite kohaselt edastada ka välistele suhtlusprogrammidele (vt Joonis 1).



Joonis 1. Loodava tervikliku süsteemi loogiline ülesehitus.

4.2 Agendi arhitektuur

Agendi komponendi loomisel leiti, et paljude kriitiliste ründemeetodite või võrgu kaardistamise tehnikate tuvastamiseks oli sobilik kasutada mitmelõimelisust (ingl k. *multi-threading*), näiteks nimeserveri, DHCP ja ARP päringute, portide skaneerimise ja meepurgi pihta tehtavate päringute jaoks. Golangis on gorutiinide kasutamine mõistlik, sest need võimaldavad samaaegselt teostada erinevaid ülesandeid, parandades seeläbi rakenduse jõudlust ja ressursikasutust. Gorutiinide sobivus I/O intensiivsete operatsioonide ja andmete paralleelse töötlemisega seotud süsteemide loomiseks, võimaldab süsteemil skaleeruda suuremate koormuste all. Iga loodud lõim analüüsib kindlate filtrite alusel leituid võrgupakette, peale anomaalse liikluse leidmist, edastatakse see veebirakendusele.

Agendi ülesseadmise jaoks luuakse Debian installatsiooni pakett, mis on kompeleeritav nii amd64 kui ka arm64 arhitektuuri silmas pidades (võimaldades monitooringu komponendi kasutust ka väiksematel seadmetel, nagu näiteks Raspberry Pi-l). Installatsiooni pakett loob automaatselt agendile vastava *systemservice*'i.

4.3 Veebirakenduse arhitektuur

Veebiliidese rakenduse kirjutamisel võeti aluseks Django raamistik, mis võimaldas hoida rakenduse teenus- ja esitluskihi ühtse tervikuna. Rakendus on ehitati üles kolme osana:

- Alerting komponent tegeleb hoiatuste loomise ja haldamisega, kus määratletakse

Alert mudel, mis salvestab endas erinevat tüüpi teavet hoiatuse kohta (aeg, tüüp, lähte- ja sihtkoht, sõnum jne). Agendi mudel hoiab endas kogu teavet agendi konfiguratsioonifaili loomiseks.

- Base komponenti kuuluvad baas- ja alammudelid ning utiliidid, mida kasutatakse teiste rakenduse komponentide poolt.
- Integration komponent haldab integratsioone väliste süsteemidega (Slack, Teams ja Mattermost). Integration mudel esindab ühendusi väliste teenustega ja selle alammudel hoiab endas teavet, mis võimaldab edastada ainult valitud tüüpi sõnumeid kasutajale sobivatel kellaaegadel.

Andmebaasina võeti kasutusele PostgresSQL-i andmebaas, mis peale Django mudelite loomist, jäi oma struktuurilt üsna lihtsaks.

Andmete kuvamiseks kasutati administraatoriliidest, mis pakub automaatselt genereeritud haldusliidest, võimaldades andmebaasi mudeleid hallata ilma täiendava koodi kirjutamiseta. Täiendavate vaadete loomiseks määrati vastavad URL-konfiguratsioonid ja mallid, mis määravad, kuidas andmed kasutajale esitatakse, võimaldades dünaamiliselt genereerida HTML koodilõike, kasutades Django mallimootorit.

Agendi konfiguratsiooni edastamiseks ja hoiatuste vastuvõtmiseks loodi REST arhitektuuri [34] põhimõtetele vastavad lõpp-punktid. Serializer klass võeti kasutusele eesmärgiga teisendada konkreetse agendi mudeli objekti andmed sobivale JSON-kujule. Vaadete rakendamiseks kasutati ViewSet klasse koos vastavate autentimisklassidega. Seansipõhist autentimist kasutati lõpp-punktide puhul, mida kasutab sisselogitud kasutaja, ja *token*-põhist autentimist rakendati agentide puhul, tagades sedasi turvalise ja kontrollitud ligipääsu süsteemile.

4.4 Veebirakenduse ja agendite suhtlus

Lähtudes eeldusest, et agent ja veebirakendus ei asu samas alamvõrgus, polnud võimalik kasutada kergeimat lahendust, milles peale kasutaja poolt veebiliideses seadistatud agendi konfiguratsiooni salvestamist, see edastatakse vastavale agendile. Sellest tulenevalt võeti kasutusele süsteem, kus kõigepealt kasutaja seadistab veebiliideses agendi, seejärel laeb alla agendi konfiguratsiooni JSON-failina, saamaks selle viimase etapina paigaldada agenti jooksutava masina konfiguratsiooni kataloogi. Peale esmast seadistust käib agent kindla ajaperioodi tagant veebirakenduselt pärimas uut konfiguratsiooni.

4.5 CI/CD

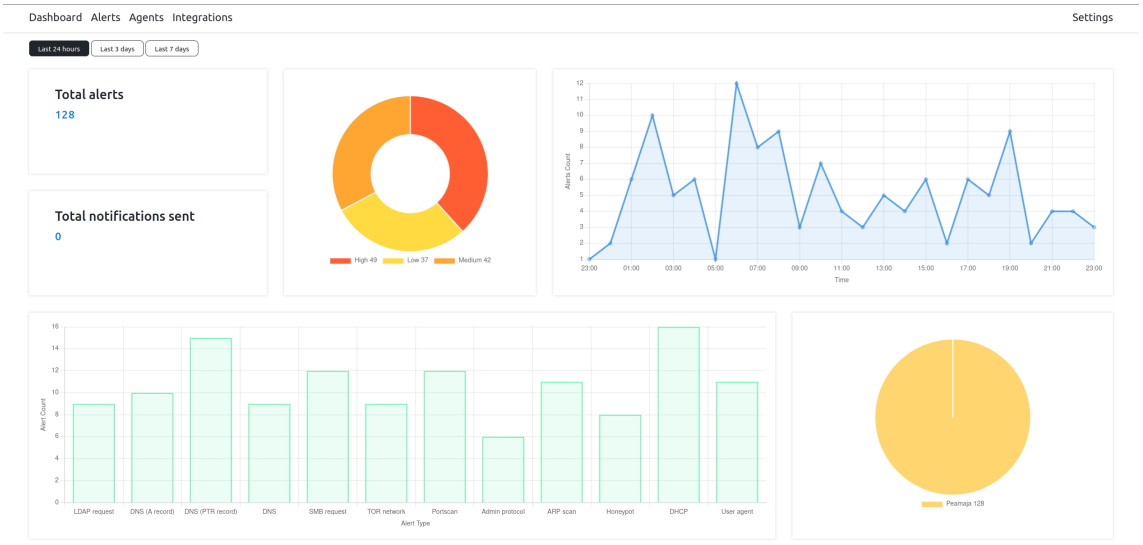
Veebirakenduse arendamisel kasutatava pideva integreerimise konveieri töövoog koosnes kolmest etapist. Esmalt paigaldakse vajaminevad teegid, et tagada keskkonna valmisolek rakenduse testimiseks ja analüüsimiseks. Seejärel sooritatakse paralleelselt automaattestide käivitamine ning koodi staatiline analüüs tööriistaga Ruff. Agendi arendamisel kasutatava pideva integreerimise konveieri töövoog paigaldas samuti esimese sammuna kasutatavad teegid. Seejärel kompileeriti agendi komponendi binaarkood ja koostati Debian pakett.

5 Tulemused

Järgnevas peatükis tutvustatakse arendusprotsessi tulemusi, andes esmalt ülevaate loodud funktsionaalsustest. Seejärel käsitletakse lahenduse töökindluse valideerimist ja selle võrdlust olemasolevate lahendustega. Lõpuks tuuakse välja süsteemi edasiarendamise võimalused.

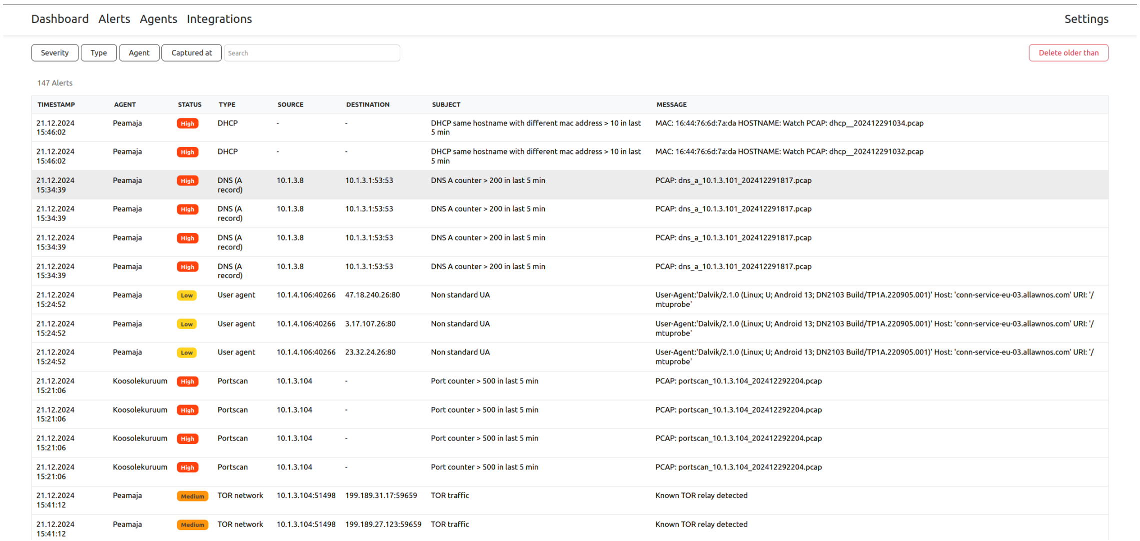
5.1 Loodud funktsionaalsused

Esmasel veebirakenduse käivitamisel genereeritakse administraator kasutaja, mille parool kuvatakse kasutajale terminalis. Peale edukat sisselogimist, suunatakse kasutaja vaikimisi *dashboard* vaatele (vt Joonis 2), kus on võimalik saada kiire ülevaade, kas viimase ühe, kolme või seitsme päeva hoiatustest. Loetletud on sisenenud ja saadetud hoiatused. Lisaks kuvatakse kasutajale erinevad graafikud, kus hoiatused on kokku võetud tüübi, agendi ja ohutaseme alusel. Samuti kuvatakse valitud ajaperioodi intsidendid ka ajateljel sageduse järgi.



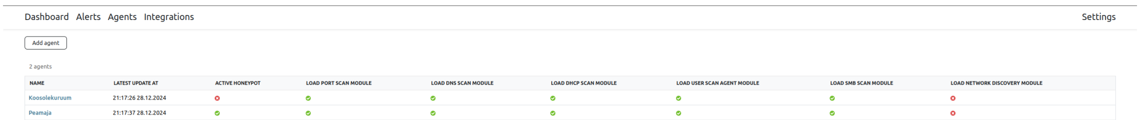
Joonis 2. Veebirakenduse dashboard vaade.

Hoiatuste detailse loetelu lehel (vt Joonis 3) kuvatakse kasutajale kõik süsteemi poolt leitud turvahoiatused koos kogu talletatud informatsiooniga. Hoiatusi saab ohutaseme, tüübi ja tuvastamise kuupäeva alusel filtreerida ning täpsemate lähte- ja sihtkohtade, pealkirjade ja sõnumite leidmiseks saab kasutada tekstiotsingut. Lisaks on iga andmetulba väärtused sorteeritavad.



Joonis 3. Veebirakenduse hoitatuste detailse loetelu vaade.

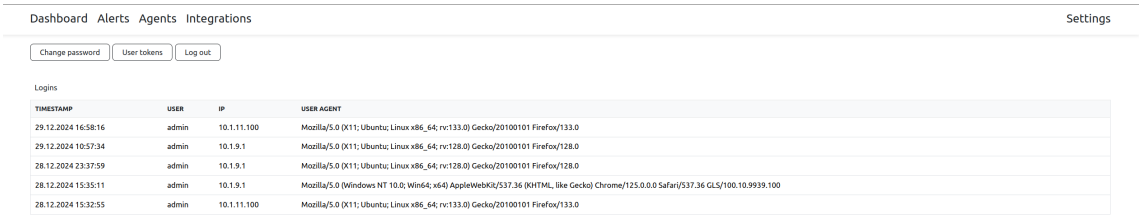
Agentide loetelus (vt Joonis 3) kuvatakse kasutajale, millised moodulid on antud agentidel hetkel aktiveeritud. Uue agendi lisamisel või olemasoleva muutmisel kuvatakse kasutajale detailne agendi konfiguratsiooni vorm.



Joonis 4. Veebirakenduse agentide loetelu vaade.

Integratsioonide järjendis leiab üles seatud väliste suhtlusprogrammide objektid. Integratsiooni detailvaates on kasutajal võimalik seadistada suhtlusprogrammi kaudu saadetavate sõnumite tüübid, kellaajavahemik kuna seda tehakse ja hoiatuste intensiivsus, mis määrab kui tihti igat sarnast teadet edastatakse. Samuti on näha kas antud integratsiooni parameetritega on õnnestunud teha edukas testpäring ja kas antud liidestus on hetkel aktiivne.

Sätete vaates (vt Joonis 5) on kasutajal võimalik vahetada kehtivat kasutaja parooli ja seadistada identifikaator *token* agendi ja veebirakenduse vaheliseks suhtluseks. Samuti kuvatakse administraator kasutjale kõik senised veebirakenduse sisselogimised.

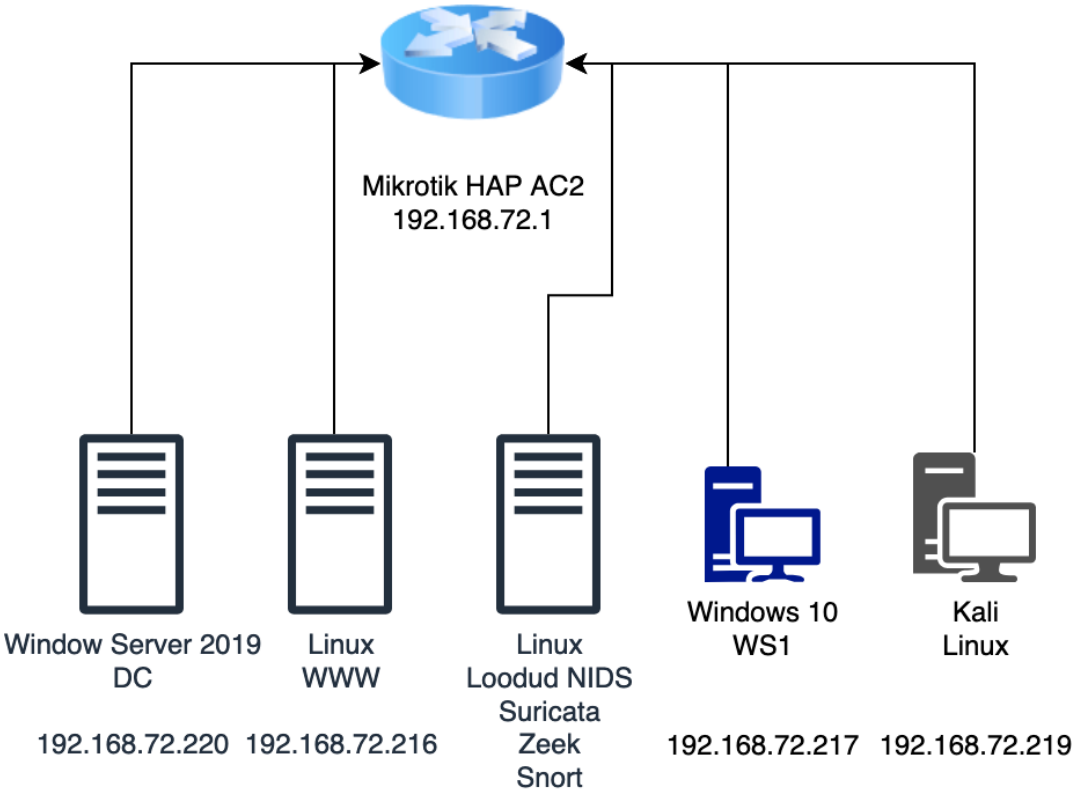


Joonis 5. Veebirakenduse sätete vaade.

5.2 Süsteemi töökindluse valideerimine

Enne loodava lahenduse kasutamist kliendipoolses sisevõrgus, seati ülesse kontrollitud testkeskkond koos järgnevate seadmetega:

- Windows 10 tööjaam
- Windows 2019 domeenikontroller
- Linux veebiserver
- Linux tööjaam, koos loodud NIDS, Suricata, Snort ja Zeek installatsiooniga
- Kali Linux tööjaam



Joonis 6. Testkeskkonna võrgujoonis.

Lahenduse kontrollimiseks võeti kasutusele avatud lähtekoodiga raamistik Atomic Red Team, mis võimaldab turvaekspertidel simuleerida standardiseeritud ja dokumenteeritud ründemeetodeid, et hinnata, kuidas kasutuses olevad turvasüsteemid suudavad pahatahtlikku tegevust tuvastada ja võimalusel ka peatada. Raamistikku lisati uued ründemeetodite kogumikud, simuleerimaks varasemalt analüüsis loetletuid ründemeetodeid (Lisa 2). Võrdluspildi loomiseks otsustas töö autor kasutada avatud lähtekoodiga sissetungi tuvastuse süsteeme: Suricata, Snort ja Zeek.

Peale IDS-ide paigaldamist ja seadistamist, uuendati kõik süsteemid vaikimisi reeglikogumitega. Töö autor otsutas mitte lisada kolmandate osapoolte reeglikogumikke, sest eelkõige sooviti võrrelda süsteemide baasfunktsionaalsust. Autori poolt loodud lahenduse funkstionaalsete nõuete ühekaupa testimise ja kolme avatud lähtekoodiga IDS tuvastusmeetmete omavahelise võrdlemise tulemusena valmis järgnev tabel.

Tabel 1. Loodud süsteemi funktsionaalsuste võrdlus olemasolevate lahenduste vahel.

Tehnika	**Loodud NIDS**	**Suricata**	**Snort**	**Zeek**
DNS TXT	+	+		
DNS PTR	+	+		
DNS A	+	+		+
Portide skaneerimine	+	+	+	+
DHCP *starvation attack*	+			
DHCP MAC aadressi vahetus	+			
Võlts DHCP server	+			+
Kasutajaagendid	+	+		+
Väljuvad SMB päringud	+	+		+
LDAP pärinugd	+			+
Administraatori poolt tehtavad päringud	+	+		
Tor päringud	+			
ARP päringud	+		+	

Läbiviidud valideerimise käigus kinnitati, et loodud lahendus tuvastas kõik ründemeetodid edukalt. Suricata osutus teistest vabavaralistest lahendustest parimaks, tuvastades iga DNS pärgingu tüübi, portide skaneerimise, erinevad kasutajaagendid ja administraatori poolt kasutatavad protokollid ning väljuvad SMB päringud. Samuti töötas hästi Zeek, tuvastades DNS A, LDAP ja välvjuvad SMB päringud, erinevad kasutajaagendid, võlts DHCP serveri ja portide skaneerimise. Kõige halvemini toimis Snort, tuvastades

ainult portide skaneerimise ja ARP päringud ning oli testimise käigus ainuke IDS, mille logidest tuvastati valepositiive. Võrdluse põhjal saab väita, et Snorti poolt pakutav baasfunktsionaalsus polnud piisavalt võimekas, pakkumaks vajalikku katvust mitmekülgsete ründemeetodite tuvastamisel. Suricata ja Zeeki baasfunktsionaalsused pakkusid vaid osalist ründemeetodite tuvastamist. Valideerimise käigus osutus autori loodud NIDS kõige tõhusamaks ja sobivamaks kasutamiseks turvalisuse tagamisel sisevõrgus, kus ründaja on kompromiteerinud lokaalvõrgus oleva tööjaama ning üritab läbi selle võrgus edasi liikuda.

Võrdluses väljatoodud IDS-ide puhul ei piisa üksnes nende ülesseadmisest, lisaks on mõistlik kasutada lisaprogramme kogutud logide efektiivsemaks analüüsimiseks. Võrguturbe haldusplatvormid pakuvad mugavat kasutajaliidest, võimaldades IDS-ide poolt tuvastatud võrguliikluse tõhusat visualiseerimist, haldamist ja analüüsimist. Suricata, Snort ja Zeek toetavad ELK Stack terviklikku lahendust, mis hõlmab endas kolme avatud lähtekoodiga toodet: Logstash, Elasticsearch ja Kibana. Logstash on serveripoolne andmete töötlemise konveier, mis töötleb IDS poolt saadetud andmed ja edastab need Elasticsearchile, mis tegeleb nende andmete hoiustamisega. Kibana võimaldab kasutajal indekseerituid andmeid visualiseerida ja otsida. Samuti on iga vaadeldud sissetungi tuvastamise süsteemi puhul võimalik kasutada rohkem kohandatud lahendusi, andmete efektiivsemaks visualiseerimiseks ja analüüsimiseks. Suricata puhul on võimalik kasutada Scirius-nimelist tarkvara, Zeeki puhul leiduvad tarkvarad Zeekurity ja Zeekcut ning Snorti mugavamaks kasutamiseks on arendatud Snorby.

5.3 Loodud lahenduse kasutamine kliendi kohtvõrgus

Peale loodud süsteemi funktsionaalsuse valideerimist, installeeriti süsteemi komponendid ühe Raspberry-Pi 5 peale (2TB NVME, 8GB RAM, 1GB NIC), mis seati üles Jõhvi vallavalitsuses. Kahenädalase katseperioodi käigus monitooris üks agent ühte alamvõrku, kus igapäevases kasutuses oli ca 60 tööjaama ja 10 serverit. Katseperioodi käigus ei tuvastatud Jõhvi vallavalitsuse sisevõrgus sinna mittekuuluvat ebatavalist tegevust. Seadistuse käigus leidis kinnitust tõdemus, et hästi dokumenteeritud võrk ja ülevaade kõigist seal leiduvatest seadmetest on oluline, võimaldamaks hoida esialgsele seadistusele kuluva töömahu mõistlikuse piirideses. Kui mistahes asutuse kasvamise käigus on võrku pidevalt lisatud uusi seadmeid, mille käigus pole algusest peale lähtutud näiteks heast tavast hoida alamvõrgus olevad seadmed vastavalt nende tüüpidele erinevates IP-aadresside vahemikus, muudab see võrgu esialgse tervikpildi koostamise natuke töömahukamaks. Jõhvi vallavalitsuse puhul ei tekkinud sellist probleemi, kuid mitmesaja seadmega võrgus muudab selline eeltegevuse puudumine süsteemi ülesseadmise oluliselt keerulisemaks. Lisaks ei osanud töö autor ette näha olukorda, kus ühe tööjaamaga võib seotud olla

mitu MAC ja IP-aadressi, mis on tingitud mitme lokaalvõrku ühendatud võrgukaardi olemasoluga.

Peale katseperioodi lõppu uuris töö autor, kas ja kuidas jäädi rahule Jõhvi vallavalitsusse paigaldatud võrgupõhise sissetungi tuvastuse süsteemiga. Peamise edasiarenduse soovina toodi välja agendi kongifuratsiooni vormi kasutajasõbralikumaks muutmise vajadus, millega töö autor ka nõustub. Lahenduse arendusprotsessi käigus lisandus skoopi uusi võrgu kaardistamise ja ründamise tehnikaid, samuti suurenes nende poolt kasutatavate lisaparameetrite hulk, mis tulenes vajadusest mugavamalt ümber käia võimalike valepositiivsete hoiatuste esialgse seadistamisega. Baasfunktsionaalsuste kohene olemasolu sai positiivset tagasisidet, mida põhjendati süsteemi esialgsele ülesseadmisele kulunud mõistliku ajaga. Lisaks küsiti tagasisidet veebirakenduse detailsemate disaniküsimuste kohta, mille arvamust võtab töö autor tulevikus arvesse, kuid mille baasil ei otsustatud koheseid muudatusi teha. Siinkohal tugineti asjaoludele, et tegu on tervikliku süsteemi esimese iteratsiooniga, mille kasutajaskond on hetkel veel liiga väike selliste otsustuskohtade hindamiseks.

5.4 Edasiarenduse võimalused

Loodud süsteemile on mitmeid edasiarenduse võimalusi, mis aitaksid süsteemi funktsionaalsust veelgi laiendada. Märkimisväärse täiendusena võiks lahendusele lisada täieliku IPv6 toe. Arvestades, et IPv6 kasutuselevõtt on pidevalt kasvamas, on seega hädavajalik, et lahendus suudaks tuvastada ja analüüsida ka selle protokolli liiklust. IPv6 keerukus ja suurem aadressiruum, esitavad täiendavaid väljakutseid, samas võimaldavad ka paremat võrgu nähtavust ning seeläbi ka turvalisuse parandamist.

Teine oluline täiendus oleks Tor-releede nimekirja automaatse uuendamise mehhanismi loomine, suurendades seeläbi süsteemi võimet tuvastada ja analüüsida Tor-võrgu liiklust. See muudaks antud lahenduse dünaamilisemaks ja vähendaks halduskoormust.

Kolmanda edasiarendusena võiks lisada võrguseadmete avastamise funktsionaalsuse. Lahenduse esialgse seadistamise ajal kaardistatakse kõik võrku ühendatud seadmed koos nende IP-aadressidega. Tulevikus, kui võrku lisanduvad uued masinad, oleks võimalik neid automaatselt tuvastada ning hoiatusena administraatorile hindamiseks edastada, kas tegemist on lubatud seadmega või potentsiaalse ohuga. Selline funktsionaalsus suurendaks süsteemi suutlikkust säilitada võrgu terviklikkust ja tuvastada ebatavalisi muudatusi.

Kokkuvõttes võimaldavad need edasiarendused suurendada loodud lahenduse paindlikkust, automatiseeritust ja turvalisust. Sellised täiendused ei nõuaks suuri arhitektuurilisi

muudatusi, mistõttu oleks neid suhteliselt lihtne rakendada. Edasine töö võiks keskenduda nende funktsionaalsuste realiseerimisele ja integreerimisele olemasolevasse lahendusse, et vastata veelgi paremini tänapäeva võrguturbe vajadustele.

6 Kokkuvõte

Antud bakalaureusetöö käsitleb stsenaariumi, milles ründaja on kompromiteerinud ühe sisevõrgus oleva tööjaama, paigaldanud lokaalvõrku enda kontrollitud seadme või saanud sisevõrgule ligipääsu läbi WiFi. Lõputöö eesmärk oli arendada kergelt paigaldatav sissetungi tuvastamise ja teavitamise süsteem, avastamaks ebatavalist või pahatahtlikku tegevust lokaalvõrgus. Eesmärgiks oli luua monitooringusüsteem, mis oleks võimeline tuvastama spetsiifilist ründaja käitumismustrit, erinedes sedasi populaarsetest avatud lähtekoodiga IDS-idest nagu Suricata, Snort ja Zeek.

Lõputöö eesmärgi saavutamiseks koondati kokku enimkasutatavad ründemeetodid, tuginedes CISA 2023. aasta riskide ja haavatavuste hindamise raportile, lisaks anti ülevaade hetkel olemasolevatest lahendustest. Analüüsi tulemusel seati süsteemile funktsionaalsed ja mittefunktsionaalsed nõuded ning valiti arendusprotsessis kasutatavad tehnoloogiad.

Töö käigus valmis Go-s kirjutatud agendi komponent, mis vastutab ründemeetodite või võrgu kaardistamise tehnikate tuvastamise eest, ja Django-põhine veebiliides, mis võimaldas hoida rakenduse teenus- ja esitluskihi ühtse tervikuna. Valminud terviklikku lahendust valideeriti esmalt testkeskkonnas, mille käigus koostati ka ründemeetodite tuvastamise võrdlus olemasolevate lahendustega. Lisaks võeti loodud lahendus kasutusele Jõhvi vallavalitsuses. Lõputöö tulemusena valminud sissetungi tuvastamise ja teavitamise süsteem täitis seatud eesmärke.

Loodud süsteemile on mitmeid edasiarenduse võimalusi, mis aitaksid süsteemi võimekust veelgi täiustada, hõlmates endas IPv6 toe lisamist, Tor-releede automaatset uuendamist ja uute võrguseadmete avastamise funkstionaalsust.

Kasutatud kirjandus

[1] Statista. *Annual number of cyberattacks in the United States from 2016 to 2022*. [Kasutatud: 26-10-2024]. URL: https://www.statista.com/forecasts/1448523/us-cyberattacks-annual.

[2] Riigi Infosüsteemi Amet. *Küberturvalisuse aastaraamatud*. [Kasutatud: 26-10-2024]. URL: https://www.ria.ee/kuberturvalisus/kuberruumi-analuus-ja-ennetus/olukord-kuberruumis?view_instance=0&current_page=1#aastaraamatud.

[3] DOT Security. *Infographic: The Layered Cybersecurity Defense*. [Kasutatud: 26-10-2024]. URL: https://dotsecurity.com/insights/blog-layered-cybersecurity-defense.

[4] Syteca. *12 Cybersecurity Best Practices & Measures to Prevent Cyber Attacks in 2024*. [Kasutatud: 27-10-2024]. URL: https://www.syteca.com/en/blog/best-cyber-security-practices.

[5] Fortinet. *What Is Defense In Depth*. [Kasutatud: 26-10-2024]. URL: https://www.fortinet.com/resources/cyberglossary/defense-in-depth.

[6] DOT Security. *12 Basic Types of Network Security Measures*. [Kasutatud: 27-10-2024]. URL: https://dotsecurity.com/insights/blog-types-of-network-security-measures.

[7] IBM. *Report: Escalating Data Breach Disruption Pushes Costs to New Highs*. [Kasutatud: 26-10-2024]. URL: https://newsroom.ibm.com/2024-07-30-ibm-report-escalating-data-breach-disruption-pushes-costs-to-new-highs.

[8] B. Adam. *Vabatarkvaraliste ründetuvastussüsteemide jõudlus- ja sobivusanalüüs eriotstarbelistes võrkudes*. [Kasutatud: 22-10-2024]. URL: https://digikogu.taltech.ee/et/Item/65d76896-c5fb-4437-8bcc-c392d464e076.

[9] S. Farooghian. *Praktiliste oskuste töötuba – avatud lähtekoodiga sissetungimise turvasüsteemid*. [Kasutatud: 24-10-2024]. URL: https://digikogu.taltech.ee/et/Item/9aaa03ee-2deb-499b-be4d-69fd0ad8e144.

[10] E. Paas. *Võrguliikluse seire ja sissetungi tuvastuse süsteemi juurutamine ja analüüs S4A baasil*. [Kasutatud: 24-10-2024]. URL: https://digikogu.taltech.ee/et/Item/dff7a38c-52dd-4766-a9f3-9a36a954dbc1.

[11] MITRE ATT&CK. *Enterprise Matrix*. [Kasutatud: 26-10-2024]. URL: https://attack.mitre.org/matrices/enterprise/.

[12] CISA. *FY23 Risk and Vulnerability assessments (RVA) Results*. [Kasutatud: 29-10-2024]. URL: https://www.cisa.gov/sites/default/files/2024-09/InfographicFY23RVA508.pdf.

[13] MITRE ATT&CK. *Application Layer Protocol: DNS*. [Kasutatud: 26-10-2024]. URL: https://attack.mitre.org/techniques/T1071/004/.

[14] Siddhesh Parab. *DNS Bruteforcing*. [Kasutatud: 26-10-2024]. URL: https://sidxparab.gitbook.io/subdomain-enumeration-guide/active-enumeration/dns-bruteforcing.

[15] Avast Antivirus. *What is port scanning?* [Kasutatud: 26-10-2024]. URL: https://www.avast.com/business/resources/what-is-port-scanning#pc.

[16] Twingate. *What is DHCP Spoofing? How It Works & Examples*. [Kasutatud: 26-10-2024]. URL: https://www.twingate.com/blog/glossary/dhcp%20spoofing.

[17] Science Direct. *Mitigation of DHCP starvation attack*. [Kasutatud: 26-10-2024]. URL: https://www.sciencedirect.com/science/article/pii/S0045790612001140.

[18] Medium. *Threat Hunting - Suspicious User Agents*. [Kasutatud: 26-10-2024]. URL: https://detect.fyi/threat-hunting-suspicious-user-agents-3dd764470bd0.

[19] 0xdf hacks stuff. *Getting Creds via NTLMv2*. [Kasutatud: 26-10-2024]. URL: https://0xdf.gitlab.io/2019/01/13/getting-net-ntlm-hases-from-windows.html#.

[20] MITRE ATT&CK. *BloodHound*. [Kasutatud: 26-10-2024]. URL: https://attack.mitre.org/software/S0521/.

[21] Olaf Hartong. *Graphing MITRE ATT&CK via Bloodhound*. [Kasutatud: 26-10-2024]. URL: https://medium.com/falconforce/graphing-mitre-att-ck-via-bloodhound-87c11aadc119.

[22] Andrey Polkovnychenko. *New .NET Malware "WhiteSnake" Targets Python Developers, Uses Tor for C&C Communication*. [Kasutatud: 26-10-2024]. URL: https://jfrog.com/blog/new-malware-targets-python-developers-uses-tor-for-c2-communication/.

[23] AltexSoft. *Functional and Nonfunctional Requirements: Specification and Types*. [Kasutatud: 26-10-2024]. URL: https://www.altexsoft.com/blog/functional-and-non-functional-requirements-specification-and-types/.

[24] Neumetric. *HIDS vs NIDS: Unravelling the Differences in Intrusion Detection Systems*. [Kasutatud: 11-01-2025]. URL: https://www.neumetric.com/hids-vs-nids/.

[25] Snort. *What is Snort?* [Kasutatud: 18-10-2024]. URL: https://www.snort.org/.

[26] Suricata. *Suricata dokumentatsioon*. [Kasutatud: 18-10-2024]. URL: https://docs.suricata.io/en/latest/what-is-suricata.html.

[27] Zeek. *Zeeki dokumentatsioon*. [Kasutatud: 18-10-2024]. URL: https://docs.zeek.org/en/master/about.html.

[28] Cisco. *Cisco Secure Firewall At-a-Glance*. [Kasutatud: 18-10-2024]. URL: https://www.cisco.com/c/en/us/products/collateral/security/firesight-management-center/at-a-glance-c45-736624.html.

[29] Check Point Software Technologies. *Check Point Intrusion Detection System*. [Kasutatud: 18-10-2024]. URL: https://www.checkpoint.com/cyber-hub/network-security/what-is-an-intrusion-detection-system-ids/.

[30] Django. *Django koduleht*. [Kasutatud: 30-10-2024]. URL: https://www.djangoproject.com/start/overview/.

[31] Go. *Go koduleht*. [Kasutatud: 30-10-2024]. URL: https://go.dev/doc/effective_go.

[32] Docker. *What is Docker?* [Kasutatud: 30-10-2024]. URL: https://docs.docker.com/get-started/docker-overview/.

[33] Tan Dang. *What Is System Architecture? A Simple Explanation*. [Kasutatud: 30-10-2024]. URL: https://www.orientsoftware.com/blog/system-architecture/.

[34] Codeacademy Team. *What is REST?* [Kasutatud: 30-10-2024]. URL: https://www.codecademy.com/article/what-is-rest.

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

Mina, Mattias Raba

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Sissetungide tuvastamise ja teavitamise süsteem pahatahtlike võrgupäringute seireks", mille juhendaja on Toomas Lepik
   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.

15.01.2025

[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 - Atomic Red Team raamistikku lisatud ründemeetodite näited

DNS sõnastiku rünnaku simuleerimine, kus ühekaupa proovitakse sõnastikust võetuid alamdomeene, eesmärgiga tuvastada võimalikke valesti konfigureeritud ja haavatavaid DNS kirjeid.

```
attack_technique: T9000.002
display_name: Completed NIDS Validator - DNS A
atomic_tests:
  - name: DNS-dictionary-attack
    description: Simulates a DNS dictionary attack by querying common
       subdomains from a wordlist to detect vulnerable or misconfigured
        DNS records.
    supported_platforms:
      - linux
    executor:
      name: bash
      elevation_required: false
      command: |
        for n in `cat /usr/share/SecLists-2024.1/Discovery/DNS/subdomains
           -top1million-5000.txt`;
        do dig @192.168.72.1 $n.skynet.local +short | grep '\.' && echo "
           $n.skynet.local "; done
```

LDAP päringute simuleerimine domeeni objektide leidmiseks, kasutades ldspsearch tööriista.

```
attack_technique: T9000.007
display_name: Completed NIDS Validator - LDAP
atomic_tests:
  - name: LDAP Query for Domain Objects
    description: Simulates LDAP queries to extract domain objects using
       ldapsearch.
    supported_platforms:
      - linux
    executor:
      name: bash
      elevation_required: false
      command: |
        ldapsearch -x -H ldap://192.168.72.220 -D "miles.dyson@skynet.
           local" -w "KalaKala1234" -b "DC=skynet,DC=local" "(
           objectClass=user)"
```