TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Arvutiteaduste instituut

ITC70LT

Oliver Soom 132335IAPM

VÕRDVÕRKUDE ANDMEVAHETUSE RAKENDAMINE KÜBERKAITSEÕPPUSE ROBOTVÕRGUS

Magistritöö

Juhendaja: Margus Ernits
M.Sc
Doktorant

Tallinn 2016

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: Oliver Soom

09.05.2016

Annotatsioon

Antud diplomitöö eesmärk on suurendada küberkaitseõppustel kasutusel oleva võrguliikluse genereerija töökindlust.

Oskus ennast kaitsta erinevate küberrünnete puhul on saanud organisatsioonidele ning riikidele tähtsaks ülesandeks. Sellepärast korraldatakse erinevaid küberkaitsele orienteerituid õppusi. Eestis korraldatakse erinevaid küberkaitseõppusi, näiteks NATO küberkaitsekoostöö keskuse egiidi all korraldatav *Locked Shields* ning Eesti Infotehnoloogia Kolledžis toimuv Küberolümpia. Võistlusel on kaks osapoolt: ründav ja kaitsev. Õppuste eesmärgiks on kaitsva poole professionaalsuse tõstmine. Küberkaitse õppuste tegevus toimub kinnises mängukeskkonnas. Üheks kaitsvale poolele olustiku realistlikumaks tegemise meetodiks ning ründavale poolele kattevarju tekitamiseks kasutatakse võrguliikluse genereerijat, mille toimimise parandamiseks antud töö ongi suunatud.

Antud töös uuriti võrguliikluse genereerijaga seotud probleeme. Uuriti hetkeolukorda ning määratleti töös lahendatav probleem ning nõuded lahendusele. Järgnevalt analüüsiti erinevaid robotvõrkude ning võrdvõrkude arhitektuure, mille põhjal leiti sobiv lahendus küberkaitse võrguliikluse genereerija töökindluse suurendamiseks. Selle põhjal loodi lahenduse prototüüp ning testiti selle vastamist esitatud nõuetele.

Töö tulemusena valmis prototüüp-lahendus võrguliikluse genereerija töökindluse suurendamiseks, mis piloteerimise käigus ilmnenud puudujääkide kõrvaldamisel võib leida rakendust küberkaitse- õppustel.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 52 leheküljel, 6 peatükki, 20 joonist, kolm tabelit.

Abstract

Implementation of Peer-to-Peer Network Data Interchange in the Botnet of Cyber Defence Exercise

The purpose of this thesis is to increase reliability of network traffic generator which is used for the cyber defense exercises.

Ability to defend oneself against different cyber-attacks has become an important task for organizations and countries all over the world. That's why different cyber defense oriented exercises are organized. In Estonia, different exercises are held. For example, Locked Shields, which is organized by NATO, and Cyber Olympics, which is organized by Estonian Information Technology College. The exercise has two sides: attacking and defending. The purpose of these exercises is to increase the level of professionalism of the defending side. Exercises are held in closed gaming environment. One of the methods, for making environment closer to real life and offering some "smokescreen" for attacking side, is network traffic generator. The aim of this thesis is to fix and make better the operations, which are made by the traffic generator.

In thesis, problems related to the traffic generator were studied. Author examined current situation and defined the problem which will be solved in this thesis. As follows, architecture and related issues of botnets and peer-to-peer networks were analyzed. Based on the analysis, the requirements for solution was proposed. The solution for increasing reliability of network traffic generator are based on the proposed requirements.

Based on proposed solution, prototype was created and tested for meeting the requirements which were submitted for the solution.

The result of thesis was a prototype for increasing reliability of network traffic generator, which after further developments and testing could find harness for cyber defense exercises.

The thesis is in Estonian and contains 52 pages of text, 6 chapters, 20 figures, three tables.

Lühendite ja mõistete sõnastik

ARPANET	*Advanced Research Projects Agency Network* esimene pakette vahetav arvutivõrk, mille arendas USA kaitseministeerium
ASCII	*American Standard Code for Information Interchange* 128-tähemärgiline keelemärkide tabel
Bot herder	sama, mis *botmaster*
Botmaster	indiviid, kes vastutab robotite eest või haldab neid
BRC	*BITNET Relay* suhtlussüsteem
C&C	*command and control* arvuti (tähtstruktuuriga võrgus üks, multiserver- või hargvõrgus mitu, võrdvõrgus puudub), mis annab zombivõrgule käske ja saab sealt teatiseid
CCDCOE	*NATO Cooperative Cyber Defence Centre of Excellence* NATO küberkaitsekoostöö keskus
DDoS	*Distributed Denial of Service* ummistusrünne, milles kasutatakse sihtsüsteemi või -võrgu liikluse mahu tunduvaks suurendamiseks suurt arvu ründavaid süsteeme
DHT	*Distributed Hash Table* võrdvõrkudes kasutatav soovitud materjali sisaldava sõlme leidmist toetav paisketabeli analoog
DNS	*Domain Name System* TCP/IP-võrgu komponentide, teenuste ja ressursside nimede hierarhiline süsteem
HTTP	*Hypertext Transfer Protocol* veebi aluseks olev hüpermeediumile suunatud standardne (RFC 7230,...,7235) rakenduskihi protokoll
IRC	*Internet Relay Chat* klient-server-mudelil põhinev 1988. aastal loodud standardne (RFC 1459) rakenduskihi protokoll tekstsõnumite vahetuseks jutusidena
ISP	*Internet service provider* ISACA: kolmas pool, kes võimaldab üksikisikutel ja organisatsioonidel pääseda internetti ja osutab mitmesuguseid internetiga seotud teenuseid

LS	*Locked Shields* igal aastal toimuv küberkaitseõppus, mida korraldab CCDCOE
Meepurk	lõks, mis on seatud avastama, tõrjuma või muul viisil käsitlema infosüsteemide lubamatut kasutamist, eeskätt ründeid ja rämpsposti
MIME	*Multipurpose Internet Mail Extensions* algset meiliprotokolli SMTP laiendav standard (alates 1992, RFC 2045 jt), mis võimaldab edastada teksti ja päiseid muudes kui ASCII-märgistikes
NATO	*North Atlantic Treaty Organization* Põhja-Atlandi Lepingu Organisatsioon, sõjaline liit
OSI	*Open Source Initiative* lähtekoodtarkvara levitamist toetav, edendav ja sertifitseeriv mittetulunduslik organisatsioon
P2P	*Peer-to-Peer* detsentraliseeritud side mudel, milles osalejatel on võrdsed võimed ja sideseanssi saab algatada suvaline pool; iga osaleja võib olla nii kliendiks kui ka serveriks
SINET	*Simulated Internet* küberkaitse õppustel kasutatav simuleeritud internet
TCP	*Transmission Control Protocol* TCP/IP-protokollistikku kuuluv ühendusepõhine transpordikihi (OSI mudelis vastab neljas kiht) protokoll
WWW	*World Wide Web* 1990. aastal loodud, klient-server-mudelil ja HTTP tüüpi protokollil põhinev hajus rakendus internetis

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Antud diplomitöö eesmärk on suurendada küberkaitseõppustel ja koolitustel kasutusel oleva võrguliikluse genereerija töökindlust ning juhitavust.

Magistritöös loodav lahendus vahetab välja olemasoleva, keskelt juhitava robotvõrgu töökindlama ning juhitavama arhitektuuri vastu, võimaldades võistlejate vahel võrdsemat võrguliikluse jaotamist ning võistluse taristu probleemide puhul võrguliikluse granulaarsemat juhtimist.

Kaitsesuunitlusega küberkaitseõppuse tegevus toimub tavaliselt kahe osapoole vahel. Kaitsjad, keda kutsutakse Siniseks meeskonnaks (Blue Team), ning ründajad, keda esindab Punane meeskond (Red Team) [1] . Sinise meeskonna ülesanne on kaitsta väljamõeldud asutuse või organisatsiooni taristut Punase meeskonna rünnete eest [2] . Lisaks on esindatud administratiivtoiminguid teostav Valge meeskond ning õppuste taristu loomise ja haldamisega tegelev Roheline meeskond [1] ,[2] . Võistluse taristus asub hindamissüsteem, mis tegeleb kaitsvate infosüsteemide funktsionaalsuse seirega ning seda kasutatakse õppustel võistlevate meeskondade tulemuste hindamiseks.

Kaitsesuunitlusega praktiliste küberkaitseõppuste põhieesmärk on treenida kaitsva osapoole liikmeid. Samas kui õppuste jooksul tekitavad võrguliiklust ainult ründavad ning hindavad osapooled, võib olla kaitsval poolel nende üksteisest eraldamine lihtne ja õppused ei pruugi täita oma eesmärki. Selle vältimiseks luuakse *ca* 700[1] robotist koosneva robotvõrgu abil kattevari, mis ei luba erinevate osapoolte liiklust kergelt eristada. Iga aastaga on robotite arv kasvanud.

Õppustel kasutusel olev arhitektuur on suuteline keskselt haldama ning juhtima *ca* 700 robotist koosnevat robotvõrku. Antud arhitektuuri suurimaks puuduseks on sõltuvus keskserverist. Selle korrapärase töötamise lakkamisel kaotab robotvõrk kontrolli robotite üle ning kogu võrgu käitumine võib muutuda juhitamatuks

[1] *Locked Shields* 2014 näitel

Eelmise *Locked Shields*'i (edaspidi LS) (2014) lõpuks oli algupärasest robotite arvust alles jäänud *ca* 50%. Ülejäänud robotid muutusid erinevatel põhjustel kättesaamatuks ning selle tõttu ka juhitamatuks[2]. Töös lahendatav probleem on ühenduse alaline või ajutine kadumine roboti ning keskserveri vahel, mis muudab robotvõrgu mittejälgitavaks ning mittejuhitavaks.

Magistritöö tulemusena proovitakse antud probleemi lahendada ning loodud tulemus võib leida rakendust küberkaitsega seotud õppustel ja samateemalistel koolitustel, näiteks igal aastal korraldataval NATO õppusel LS[3] , Eesti Infotehnoloogia Kolledži virtuaallaborite süsteemis, mida rakendatakse Küberolümpial [4] , ja teiste samalaadsete ürituste raames.

Samuti on võimalik tulemust rakendada kõrgkoolide ja ülikoolide õppekavades olevate ainete jaoks, näiteks Tallinna Tehnikaülikooli [5] või Eesti Infotehnoloogia Kolledži küberturve tehnoloogia õppekava raames [6] .

1.1 Metoodika

Töö käigus kaardistab autor hetkel kasutusel olevat lahendust ning määratleb lahendatava probleemi. Järgnevalt analüüsib autor probleemi, mille raames kirjeldatakse antud probleemi olemust ja võimalusi ning vahendeid selle lahendamiseks. Probleemianalüüsi raames püstitatakse nõuded, millele loodav lahendus peab vastama, ning sellele tuginedes valitakse ka sobiv arhitektuur. Vastavalt püstitatud nõuetele luuakse lahenduse prototüüp. Loodud prototüübi sobivust nõuetele hindab töö autor testimise abil.

1.2 Ülevaade tööst

Töö koosneb viiest peatükist. Hetkeolukorra peatükis tutvustab autor küberkaitse õppusi: seal osalevaid pooli, õppuste taristut ning hetkel kasutusel olevat võrguliikluse genereerijat. Sellele järgnevas peatükis analüüsitakse kõigepealt probleemi ning vaadeldakse põhjalikult robotvõrke ning nendega seotuid aspekte. Seejärel püsitakse nõuded, millele uus võrguliikluse genereerija lahendus vastama peab, ning neist lähtuvalt valitakse sobiv lahendus. Teostuse peatükis tutvustatakse lahenduse prototüübi töötamist.

[2] Robotvõrgu haldaja sõnul

Testimise osas testitakse lahenduse prototüübi vastamist esitatud nõutele. Viimases peatükis võetakse tehtud töö kokku ning tehakse ettepanekud edaspidisteks tegevusteks.

1.3 Tänusõnad

Autor tahab kõigepealt tänada Margus Ernitsat, kelle juhendamise, toetuse ning heade nõuannete abil antud magistritöö valmis. Samuti on autor tänu võlgu Krista Pärtelile keeleliste nõuannete eest ning Lauri Võsandile, kelle asjakohane kriitika aitas kaasa töö valmimisele. Lisaks tahab autor tänada kõiki neid, kes jäid nimeliselt mainimata, kuid kes olid selle töö juures abiks ja toeks.

2 Hetkeolukord

Kaitsesuunitlusega küberõppuste eesmärk on harjutada tegutsemist küberrünnete korral. Õppustel on ülesandeks kaitsta küberrünnete eest ette antud stsenaariumi järgi mõne väljamõeldud asutuse või riigi infotehnoloogilist taristut. Õppustel osalejate tulemusi hinnatakse punktidega. Antud töös kirjeldatakse kahte küberõppust (LS ja Küberolümpia), sest parandatav võrguliikluse generaator on neis kasutusel.

LS on koostöös NATO Küberkaitsekoostöö keskuse (CCDCOE), Eesti Kaitseväe, Riigi Infosüsteemi Ameti, Eesti Küberkaitseliidu ning välispartneritega korraldatav tehniline küberkaitseõppus, mille eesmärk on treenida ning harjutada tegutsemist küberrünnete korral [2] . Õppusel osaletakse meeskonniti.

Teiseks uuritavaks küberkaitse õppuseks on Eesti Infotehnoloogia Kolledži, Eesti Kaitseministeeriumi ning Vequirty poolt korraldatav Küberolümpia. Esimene Küberolümpia peeti 14. veebruaril aastal 2015. Õppus on suunatud tudengitele, kes õpivad bakalaureuse-, rakenduskõrghariduse- ja magistritasemetel. Küberolümpial osaletakse individuaalselt.[4]

2.1 Õppuste osapooled

Tavaliselt on kaitsesuunitlustega tehnilistel õppustel osalejad jagatud viide rühma: Sinised, Punased, Kollased, Valged ning Rohelised. [1] ,[7]

2.1.1 Sinine meeskond

Sinine meeskond etendab õppustel kaitsvaid vägesid. Sinise meeskonna liikmed on oma igapäevaelus IT-spetsialistid [7] . Sinine meeskond on õppuste peamiseks sihtrühmaks, kelle treenimise jaoks antud üritusi korraldatakse. Siniste meeskondade liikmete ülesanneteks on ette antud taristu kaitsmine ning turvamine Punase meeskonna rünnete eest [1] . Sinisel meeskonnal on õppuste planeerimise käigus piiratud võimalus tutvuda ning harjutada õppustel turvatava taristu kaitsmist. Õppuste alguses viiakse taristu tagasi oma vaikimisi seadistusse. [8]

2012. aasta LS-il osales 10 Sinist meeskonda, 2014. aastal 12 Sinist meeskonda ning 2015. aastal juba 16 Sinist meeskonda. LS-is võistlevad meeskonnad on pärit NATO või tema partnerriikidest. [1] ,[2] ,[9]

Küberolümpial 2015 osales 21 võistlejat, viiest erinevast Eesti kõrgkoolist [10] .

2.1.2 Punane meeskond

Punase meeskonna ülesandeks on ette antud stsenaariumite järgi teha Siniste meeskondade kaitse all olevatele taristutele ründeid. LS-il kasutab Punane meeskond rünnete planeerimiseks "valge kasti" lähenemist. Punane meeskond teeb koostööd Rohelise meeskonnaga, et ehitada nõrka taristut ning selle abil teada saada Sinine meeskonna taristu nõrku kohti. [1]

Küberolümpial on Punane meeskond asendatud tarkvaraga ning ründamist teostatakse skriptide ja robotvõrkude abil, mida juhitakse ja käivitatakse korraldusemeeskonna poolt [11] .

2.1.3 Roheline meeskond

Rohelist meeskonda saab pidada küberkaitse õppuste siseseks ISP-iks [7] . Rohelise meeskonna ülesannete hulka kuulub õppuste põhitaristu paigaldamine ning seadistamine. Rohelised disainivad ning ehitavad mängukeskkonna koostöös Punaste ja Valgete meeskondadega. Roheline meeskond vastutab ka Sinise meeskonna võrgu ehituse, automaatse hindamissüsteemi, võrguliikluse genereerija ning monitoorimise lahenduse arendamise eest [1] . Õppuste planeerimise faasis teeb Roheline meeskond koostööd Punase meeskonnaga, ehitamaks taristut, milles oleks piisaval hulgal nõrku kohti, mis lubaks korraldada erinevaid tüüpi ründeid [8] . Hilisemaks õppustejärgseks analüüsiks peaks Roheline meeskond suutma salvestada kogu õppuste võrguliikluse [7] .

2.1.4 Valge meeskond

Valge meeskonna ülesannete alla kuuluvad harjutuste ettevalmistamine ning nende kontrollimine sooritamise vältel, mängu stsenaariumite kirjutamine, harjutuste eesmärkide määramine, ülesannete andmine Punasele meeskonnale ning nende täitmise jälgimine. Lisaks otsustab Valge meeskond, millal peab teatud õppuste osa algama ning otsustab punktide andmise üle. Valge meeskond jälgib, et keegi ei teeks sohki, näiteks ei

kasutaks keelatud seadeid tulemüüri juures. Valge meeskond kuulutab tihti välja ka õppuste võitja. [1] ,[7]

2.1.5 Kollane meeskond

Kollase meeskonna ülesannete hulka kuuluvad õppustel toimuva kohta informatsiooni kogumine ning selle meeskondadele jagamine, informatsiooni grupeerimine ja visualiseerimine. Informatsiooni kogutakse meeskondadelt tulevatest raportitest [1] .

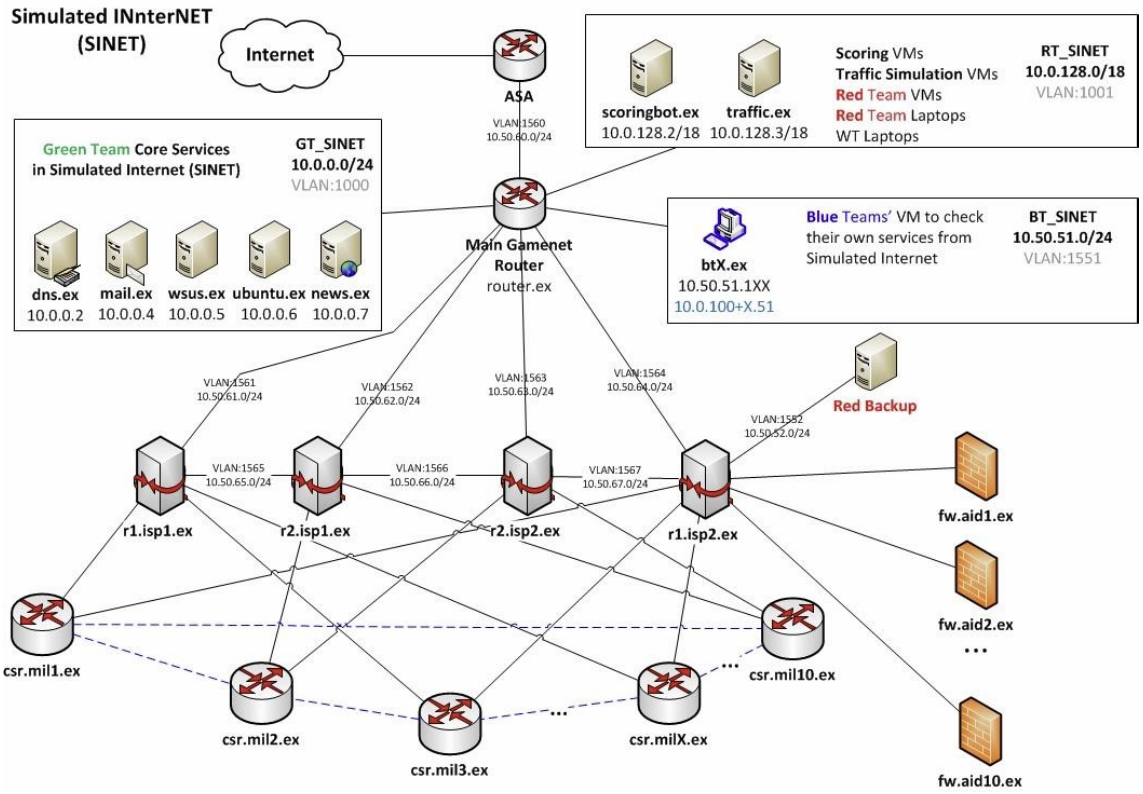
2.2 Õppuste taristute kirjeldus

Tehnilise olukorra peatükis kirjeldatakse küberõppustel kasutusel olevat taristut ning selle praegust olukorda. Peatüki teises osas kirjeldatakse õppustel kasutatavat võrguliikluse genereerijat (vt Võrguliikluse genereerijaSissejuhatus).

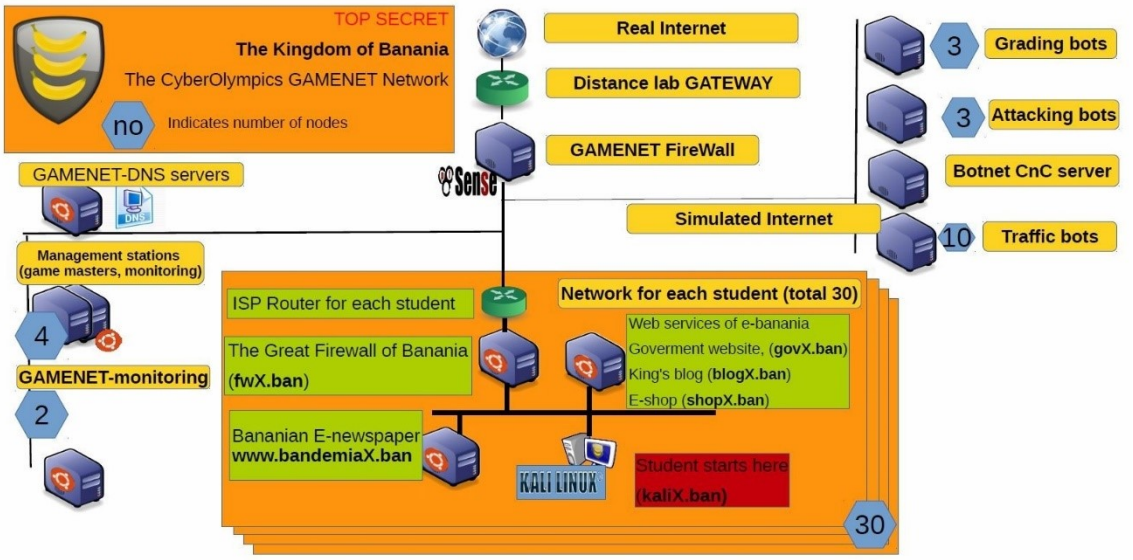
2.2.1 Õppuste keskkond

Küberkaitseõppuse tegevus toimub välisest võrguühendusest eraldatud mängukeskkonnas. LS-il (vt Joonis 1 [1]) ning Küberolümpial (vt Joonis 2 [11]) on selle nimeks Gamenet.

Küberolümpia Gamenet koosneb võistlejate poolt hallatavast taristust, mida on kuni 30 koopiat: üks igale võistlejale. Võistleja poolt hallatavasse taristusse kuuluvad tulemüür, veebiserverid, milles hoitakse erinevaid teenuseid, ning marsruuterid. Võistlejate süsteeme rünnatakse kolme ründeroboti poolt. Võistlejate sooritusi hinnatakse hindamisroboti abil, kes testib tudengite poolt hallatavate teenuste funktsioneerimist. Hindamisroboteid juhitakse Gameneti haldusserveritest. Ründavatele robotitele teevad kattevarju liiklust tekitavad robotid, kelle juhtimisega tegeleb eraldi keskserver. Lisaks kuuluvad õppusi toetavasse taristusse ka Gameneti DNS-serverid, seireserverid ning tulemüürid. [11]

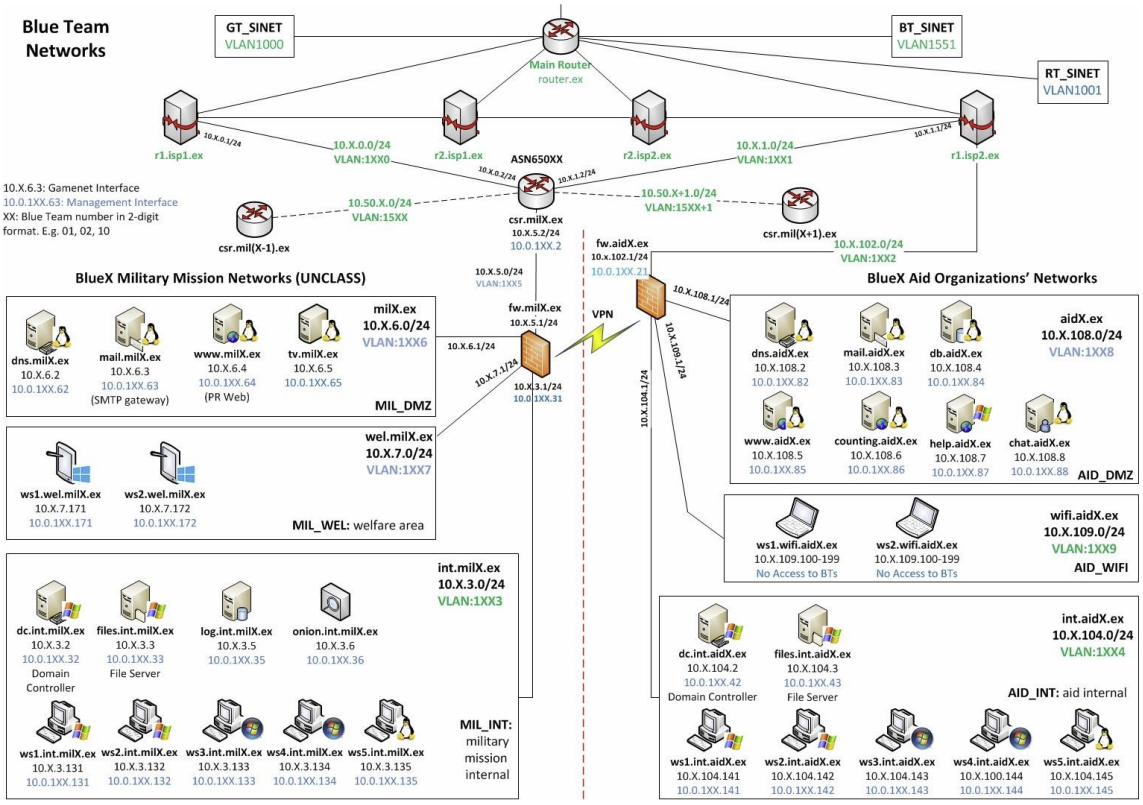


Joonis 1. SINET *Locked Shields 2013*



Joonis 2. Küberolümpia Gamenet

LS mängukeskkond koosneb kahest osast: Kaitsva poole taristust (vt Joonis 3 [1]) ning simuleeritud internetist, mida LS puhul nimetatakse SINET-iks.



Joonis 3. Sinise meeskonna taristu *Locked Shields* 2013

Kaitsva poole ning selle alla kuuluvate meeskondade taristud koosnevad võrdsest arvust virtuaalmasinatest. Sinna hulka kuuluvad virtuaalsed marsruuterid, tulemüürid, Microsoft Windowsi ja Linuxi-põhised tööjaamad ning serverid, mis majutavad andmebaase ning veebirakendusi. Lisaks kuuluvad taristu hulka domeenikontrollerid, faili, -nime- ja e-posti serverid, Androidi virtuaalmasinad, IP-kaamerad ning VoIP lahendused. [1] ,[2] ,[8]

Õppuste mängukeskkonna teise osa moodustab SINET (Simulated Internet), mis omakorda jaguneb kolmeks.

GT_SINET (Roheline meeskond): sinna ossa kuuluvad õppuste haldamiseks vajalikud süsteemid: tulemüürid, marsruuterid ning masinad, mis teenindavad Rohelise meeskonna jaoks vajalike teenuseid. [1]

RT_SINET (Punane meeskond): sinna alla kuuluvad ründeid tegevad masinad, hindamise ning taustaliikluse tekitamist juhtiv keskserver [1] .

BT_SINET (Sinine meeskond): selle kaudu saab kaitsev pool kontrollida kaitstavate süsteemidele ligipääsetavust SINET-ist [1] .

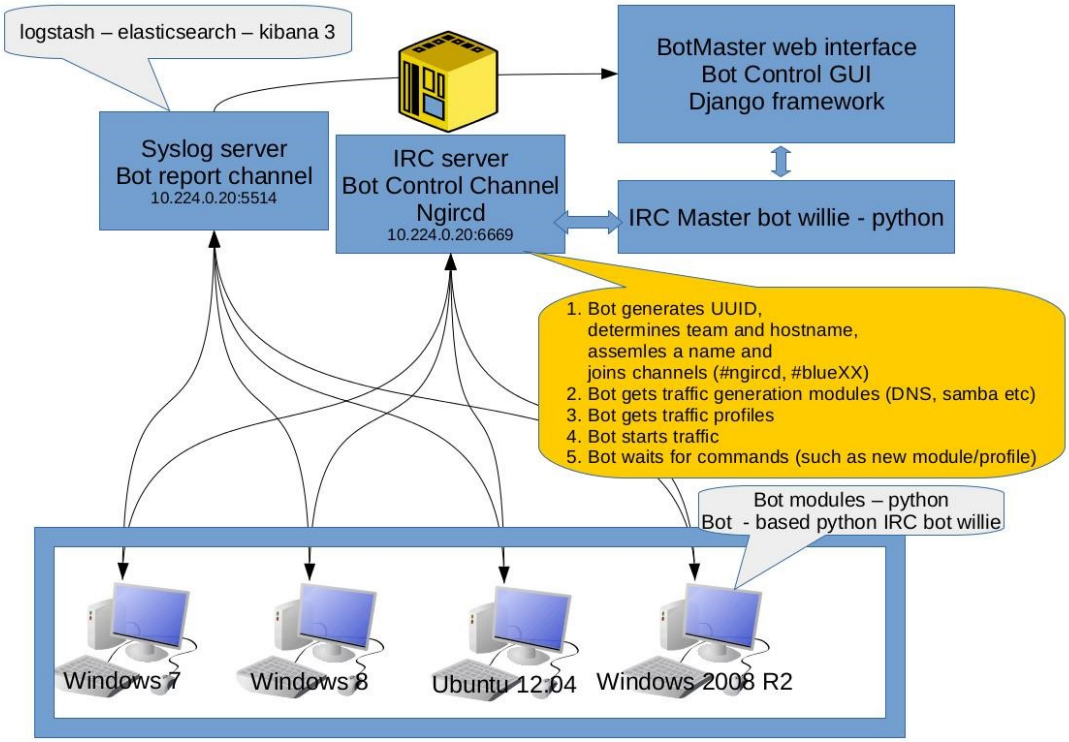
2.2.2 Võrguliikluse genereerija

Õppustel kasutatakse võrguliikluse genereerijaid eesmärgiga toota võrgus müra ning kattevarju rünnet teostavale poolele. Antud tegevus muudab õppuste keskkonna realistlikumaks ning kaitsvale poolele keerukamaks, kuna ilma võrguliikluse genereerijata tekitaksid kaitsvale poolele liiklust vaid ründavad ning hindavad osapooled.

See muudab ründajate elimineerimise lihtsamaks ning õppused ei pruugi täita oma eesmärki. Võrguliikluse juurdetekitamine loob ründajatele kattevarju ning kaitsval poolel on nende eraldamine muust liiklusest raskendatud.

Võrguliikluse genereerija peab olema suuteline muutma tekitatava liikluse mahtu ning hulka. On tähtis, et võrguliikluse tekitaja töötaks kogu õppuse vältel. Süsteemi mittetöötamine võib panna võistlejad ebavõrdsesse olukorda. Näiteks kui liikluse genereerija toodab ühe kaitsva meeskonna taristu pihta suuremahuliselt liiklust, muutub rünnete tõrjumine ning punktide saamine raskendatuks. Vastupidiselt võib mõne meeskonna pihta tehtav liiklus olla väga väike ning muudab taristu kaitsmise ning punktide saamise lihtsaks.

Hetkel on kasutusel keskset serverit kasutav robotvõrk(vt Joonis 4 [8]), mis arendati Erki Naumanise magistritöö „*Centrally Managed Network Traffic Generation For Cyber Security Exercises*“ raames. Ühendus keskserveri ning robotite vahel toimub läbi krüpteeritud IRC protokolli. Lahendus lubab roboteid kekselt hallata ning anda neile käsklusi, mille sisuks on sihtmärgid, ning nende pihta tehtava võrguliikluse maht. [8]



Joonis 4. Kasutusel olev võrguliikluse genereerija

Juhtsõnum, mida jagatakse robotite vahel, sisaldab endas loodava võrguliikluse profiile ning võrke, millele genereeritav võrguliiklus suunatakse. Näited *webtraffic*'u (vt Joonis 5) ning DNS juhtsõnumi (vt Joonis 6) sisust:

```
load: 0
target:
- domain: http://traffic3.ex/
  logsteps: []
- domain: http://news.ex/
  logsteps: []
- domain: http://redmine.bluex.ex/
  logsteps: []
- domain: http://shop.bluex.ex/
  logsteps: []
- domain: http://wiki.bluex.ex/
  logsteps: []
- domain: http://mail.bluex.ex/
  logsteps: []
- domain: http://www.bluex.ex/
  logsteps: []
```

Joonis 5. *Webtraffic*'u juhtsõnum

```
load: 0
target:
- {host: ftp.estpak.ee., rr: A}
- {host: ex., rr: MX}
- {host: itcollege.ee., rr: A}
- {host: itcollege.ee., rr: MX}
- {host: www.itcollege.ee., rr: A}
- {host: 202.194.40.193.in-addr.arpa, rr: PTR}
- {host: itcollege.ee., rr: NS}
- {host: ex., rr: NS}
- {host: traffic2.ex., rr: A}
- {host: traffic2.ex., rr: AAAA}
- {host: cucm-mgmt1.ex., rr: A}
- {host: cucm-mgmt1.ex., rr: AAAA}
```

Joonis 6. DNS juhtsõnum

3 Analüüs

Analüüsi käigus vaatleb autor LS-i näitel, töös lahendatavat probleemi. Järgnevalt tutvustab autor robotvõrkude olemust, nende levimist ja kasutusvõimalusi. Robotvõrkude topoloogia ning kommunikatsiooni osas vaadeldakse erinevaid robotvõrkude arhitektuure ning neis kasutatavaid kommunikeerumise viise. Probleemi analüüsist lähtuvalt püstitatakse nõuded, millele lahendus peab vastama ning leitakse sobiv arhitektuur.

3.1 Probleemi analüüs

Hetkel kasutusel oleva lahenduse juures tekitab probleeme ühenduse alaline või ajutine kadumine roboti ning keskserveri vahel, näiteks 2014. aasta LS-i alguses oli võrguliikluse genereerimiseks ~750 robotit, õppuste lõpuks oli neist kättesaadavad ning juhitavad ~350 robotit. Alles oli jäänud ~47 protsenti robotitest.

Taolise olukorra võivad põhjustada mitmed tegurid. Näiteks taristulised probleemid, kaitsva poole poolt masinatele tehtud muudatused (resolv.confi ümberkirjutamine, ühenduste piiramine) või operatsioonisüsteemidest tingitud erisused, näiteks saadab keskserver robotitele välja uue komplekti käsklusi. Robotid, mis kasutavad Linuxi operatsioonisüsteemi, saavad selle kätte ning uuendavad oma käskluste baasi ära. Samas robotid, mis kasutavad Microsoft Windowsi, ei saa käsklusi uuendatud, sest operatsioonisüsteemis toimunud uuendused ei lase uusi käsklusi rakendada ning kehtima jäävad vanad käsklused, mis ei ole enam kooskõlas robotvõrgu uue tegevusplaaniga ning robotid muutuvad võrgu silmis juhitamatuks. 2013. aasta LS-i ajal seiskus õppuste läbiviimine 40 minutiks, sest võrguliiklus oli õppuste keskkonna, Gameneti, võrgu üle koormanud. Selle põhjuseks oli osa roboteid, kellel oli vana korralduse järgi ülesanne tekitada suures mahus võrguliiklust. Uue korralduse tulles ei suutnud robotid käsklust omastada ning jätkasid võrguliikluse genereerimist vanal viisil. Sellega korraldasid nad võrgus põhimõtteliselt DDoS ründe.

Samuti tekitavad robotvõrgus juhitamatult käituvad robotid ebavõrdseid olukordi õppustel osalevate meeskondade või võistlejate vahel (vt Võrguliikluse genereerija).

Õppustel saavad kõrgemad punktid need, kes suudavad edukamalt enda taristu pihta tehtavaid ründeid ennetada, tõrjuda ning nendest taastuda. Ilma võrguliikluse genereerijata tekitaks kaitsva poole pihta liiklust ainult hindajad ning ründavad osalised. Selline olukord muudaks Sinise meeskonnal rünnete takistamise ning seeläbi punktide saamise lihtsamaks.

Võrguliikluse genereerija kasutamine tekitab kattevarju, mis ei luba kaitsval poolel enam nii lihtsalt ründajat tuvastada ning muudab õppused rohkem väljakutseid pakkuvamaks. Samuti muudab see kaitsvatel meeskondadel punktide saamise keerulisemaks. Juhitamatu robotvõrk võib tuua meeskondade või võistlejate vahele ebavõrdsust. Näiteks kui meeskonna A taristu pihta tehakse robotite juhtimatuse tõttu suuremahulisemalt liiklust kui meeskonna B taristu pihta, on meeskond A ebavõrdsemas seisus.

Praegune probleem on ebavõrdsus õppustel osalevate meeskondade vahel ning õppuste keskkonna vähene sarnanemine realistlikule olukorrale, mis on tingitud kasutatava robotvõrgu juhitamatusest.

3.2 Robotvõrgu olemus

"*A botnet is comparable to compulsory military service for windows boxes*" – Stromberg [12] [3]

Botiks[4] nimetatakse arvuteid või mobiilseid seadmeid, mis on nakatunud pahavaraga, mis muudab nad erinevaid käsklusi täitvaks „robotiks“. Taolised seadmed moodustavad omavahel võrgu, mida kutsutakse robotvõrguks või botnetiks. Võrgu tööd juhib tavaliselt hierarhiliselt kõrgemal olev server, mida nimetatakse kesksserveriks (C&C). Olenemata robotvõrgu tüübist, juhib kõiki robotvõrke inimliides, keda nimetatakse *botmaster'*iks *või botherder'*iks. Robotvõrku kasutatakse peamiselt pahatahtlike ülesannete läbiviimiseks, mille ülesanne on teenida *botmaster*'i huve või vajadusi. [13] ,[14] ,[15]

[3] „Robotvõrk on võrreldav arvutite kohustusliku sõjaväe teenistusega.“

[4] Bot on lühend sõnast „robot“.

Peamised tegevused, mille jaoks robotvõrke kasutatakse, on informatsiooni kogumine, hajustöötlus, küberpettused, pahavara levitamine, poliitiliselt motiveeritud kräkkimine, peale sunnitud turundamine ning võrguliikluse segamine. [15]

3.2.1 Robotvõrgu elutsükkel

Robotvõrgu elutsükkel koosneb tavaliselt viie faasist: esialgne nakatumine, teise järgu nakatumine, ühendamine, pahatahtliku käsu saatmine ja käskluste uuendamine ning haldamine. [16]

Esimeses etapis skaneerib ründaja sihtmärgiks valitud võrku ning proovib sealt leida masinaid, mida erinevaid meetodeid kasutades nakatada. Peale esimese faasi õnnestumist järgneb teise järgu nakatamine, mille käigus paigaldatakse nakatunud masinasse kestakood, mille abil robotvõrgu kahendkood laetakse alla ning paigaldatakse. Tavaliselt saadakse see mõnest konkreetsest kohast, kasutades kas FTP, HTTP'd või P2P. Kahendkoodi paigaldamisega muudetakse nakatunud arvuti "robotiks", mille kaudu teostatakse pahatahtlikke tegevusi. Pahavara, mis arvutisse on paigaldatud, käivitub igal taaskäivitumisel. [16]

Ühenduse faasis loob paigaldatud pahavara ühenduse keskserveriga. Selle kaudu saab arvuti osaks robotvõrgust. Peale seda hakkab arvuti täitma keskserveri poolt tulevaid käsklusi. *Botmaster* kasutab keskserverit, levitamaks käsklusi kogu robotvõrgule. Selle abil saab *botmaster* juhtida suurel hulgal pahatahtlikkusele kallutatud arvuteid korraga. Viimases elutsükli faasis toimub nakatunud arvutite haldamine ning käskluste uuendamine. [16]

Aeg-ajalt lastakse nakatunud arvutil alla laadida uuendatud robotvõrgu spetsiifilist tarkvara. See sisaldab robotvõrgus toimunud uuendusi ning parandusi. Uuendused võivad sisaldada uusi funktsionaalsusi või uusi avastamisevastaseid meetmeid. Lisaks võidakse uuendusi kasutada ka serveri migratsiooniks, mille käigus viiakse robot üle teise keskserveri alla. [16]

Oma robotvõrgu nähtamatuse säilitamiseks ning kaitseks, kasutavad *botmaster*'id dünaamilist DNS-i, mille puhul on hõlbus uuendada ning vahetada serveri asukohta. Näiteks kui võimud on avastanud keskserveri , saab dünaamilist DNS kasutades seada

üles uue keskserveri, millel on sama nimi, kuid teine IP-aadress. Aadressi muudatus levib kohe botidele ning see lubab robotvõrgul tegevust jätkata. [16]

3.2.2 Robotvõrgu levimine

Robotiks muutvat pahavara levitatakse aktiivset või passiivset meetodit kasutades [15] .

Aktiivsed meetodid

Aktiivne levimine toimub ilma kasutaja sekkumiseta. Kõige levinum meetod selleks on skaneerimine. Selle ajal otsib pahavaraga nakatunud arvuti nõrgalt turvatud arvuteid ning seda ära kasutades hangib arvuti administraatori privileegid ja hõivab privileege kasutades sihtmärgiks oleva arvuti. Sellele järgneb kestakoodi paigaldamine, vastava tarkvara alla laadimine ja paigaldamine ning keskserveriga ühenduse loomine. [15]

Passiivne meetod

Passiivne robotvõrgu pahavara levimine nõuab inimteguri kaasamist [15] .

Tahtmatu allalaadimine: kasutaja arvuti võib pahavara paigaldada, kui ta satub mõnele veebilehele, mis on kaaperdatud või spetsiaalselt loodud pahavaraga nakatamiseks. Need veebilehed sisaldavad aktiivsisu, mis automaatselt laadib alla ning nakatab kasutaja arvuti, muutes selle robotiks. [15]

Nakatunud andmekandja: kasutades mõnda andmekandjat, näiteks USB-põhine andmekandja või CD-ROM. Kirjeldatud vahendeid kasutades võib arvuti nakatuda ka võrgus olemata. [15]

Sotsiaalvõrgustikud

Robotiks muutvat pahavara võib levitada ka läbi sotsiaalvõrgustike. Klikkides mõne usaldusväärse, kuid kaaperdatud konto peal, võib see suunata kasutaja veebilehele, millel olles laetakse alla ning paigaldatakse kasutaja arvutisse aimamatult pahavara. Teine populaarne viis on levitada pahavara läbi e-kirja manuste. [15]

3.2.3 Robotvõrkude kasutamine

Robotvõrke kasutatakse tavaliselt pahatahtlikel või omakasupüüdlikel eesmärkidel [15] .

Informatsiooni kogumine

Informatsiooni kogutakse finantsilise või teabelise kasu saamiseks. Robotvõrku kuuluvad robotid kasutavad nakatunud masinast tundliku informatsiooni kätte saamiseks erinevaid viise. Informatsioon, mida nad koguvad, võib olla krediitkaardi või pangakonto numbrid, kasutajanimed, salasõnad vms. Robotvõrke kasutatakse ka luure eesmärgil informatsiooni kogumiseks. Sellise tegevuse sihtmärgiks võivad olla erinevad riigid või organisatsioonid. Näiteks Aurora-nimeline robotvõrk oli spetsialiseerunud, kogumaks informatsiooni Google'i kohta. [15]

Hajustöötlus

Lauaarvutid kasutavad tavaliselt ära ainult 5% oma arvutusvõimsusest [17] . Robotvõrgus kasutatakse maksimaalselt ära kogu robotvõrgustatud arvutite arvutusvõimus, et hajutatud viisil jagada ning hoida faile, muukida salasõnu või teostada teisi tegevusi [15] .

Küberpettused

Küberpettuseks nimetatakse internetis tehtavat toimingut, mille abil teenitakse tulu ebaõiglasel viisil. Küberpettusteks kasutatakse ka robotvõrke. Nende abil koguvad *botmaster*'id krediitkaardi numbreid või salasõnu. Samuti võimaldavad robotvõrgud läbi viia sobinguid internetimängudel ning võrguhääletustel, kasutades selleks „tellimusroboteid", mis tegutsevad *botmaster*'i huvisid silmas pidades. Robotvõrke kasutatakse ka klikipettuste läbiviimiseks. Robotid suunatakse klikkama *pay-per-click* reklaamidel, kallutades sellega klikkide arvu *botmaster*'i huvides. [15]

Pahavara levitamine

Robotvõrke kasutatakse ka teiste pahavarade levitamiseks. Robotvõrgu kaudu pahavara levitamine võib olla selle võrgu peamine funktsioon, kuid samas võidakse ka robotvõrke samal otstarbel ajutiselt välja rentida. [15]

Kübervõitlus

Kübervõitlus nimetatakse tegevust, mille raames segab või kahjustab üks riik teise riigi vara, sisenedes viimase võrku ning arvutitesse. Viimastel aastatel on suurenenud robotvõrkude kasutamine küberrünnetes. Küberruumi peetakse riikide jaoks tähtsaks strateegiliseks punktiks, mille kontrollime annab teise ees olulise eelise. [15]

Antud kontseptsioon tõestas ennast 2007. aasta kevadel, kui Eesti veebilehti tabas väidetavalt Venemaa poolne ummistusrünne [18] .

Teise näitena võib tuua Stuxneti-nimelise robotvõrgu, mis häiris Iraanis uraani rikastamisel kasutatavaid tsentrifuuge juhtivaid kontrollereid [15]

Pealesunnitud turundus

Internetiturundus on tõestanud, et on oma kiiruse ja odavuse tõttu efektiivsem kui traditsioonilised turunduse meetodid. Kuid mõned turundajad kasutavad seda ära. Kasutades rämpsposti, hüpikaknaid vms, sunnivad turundajad kasutajatele peale mittevajaliku reklaami. Viimastel aastatel on ette võetud mitmeid samme sellise tegevuse vähendamiseks või tõkestamiseks. Üheks võimalikuks meetmeks on näiteks e-posti serveri *blacklist*'i panemine. [15]

Meetmetele vastukaaluks on spämmerid oma tegevuse jätkamiseks kasutusele võtnud robotvõrgud. *Botmaster* saadab robotitele laiali nimekirjad mailiaadressidega, millele rämpsposti saatma hakatakse. Sellisel jagatud viisil saatmine muudab robotvõrgu avastamise väga keeruliseks, sest iga robot vastutab ainult väikese osa väljasaadetavate e-kirjade eest. [15] Näiteks 2011. aastal deaktiveeritud Rustocki robotvõrgus oli Microsofti andmetel ligi 1.3 miljonit unikaalse IP-ga robotit, kes tegelesid rämpsposti saatmisega. [19] ,[20]

Võrguliikluse häirimine

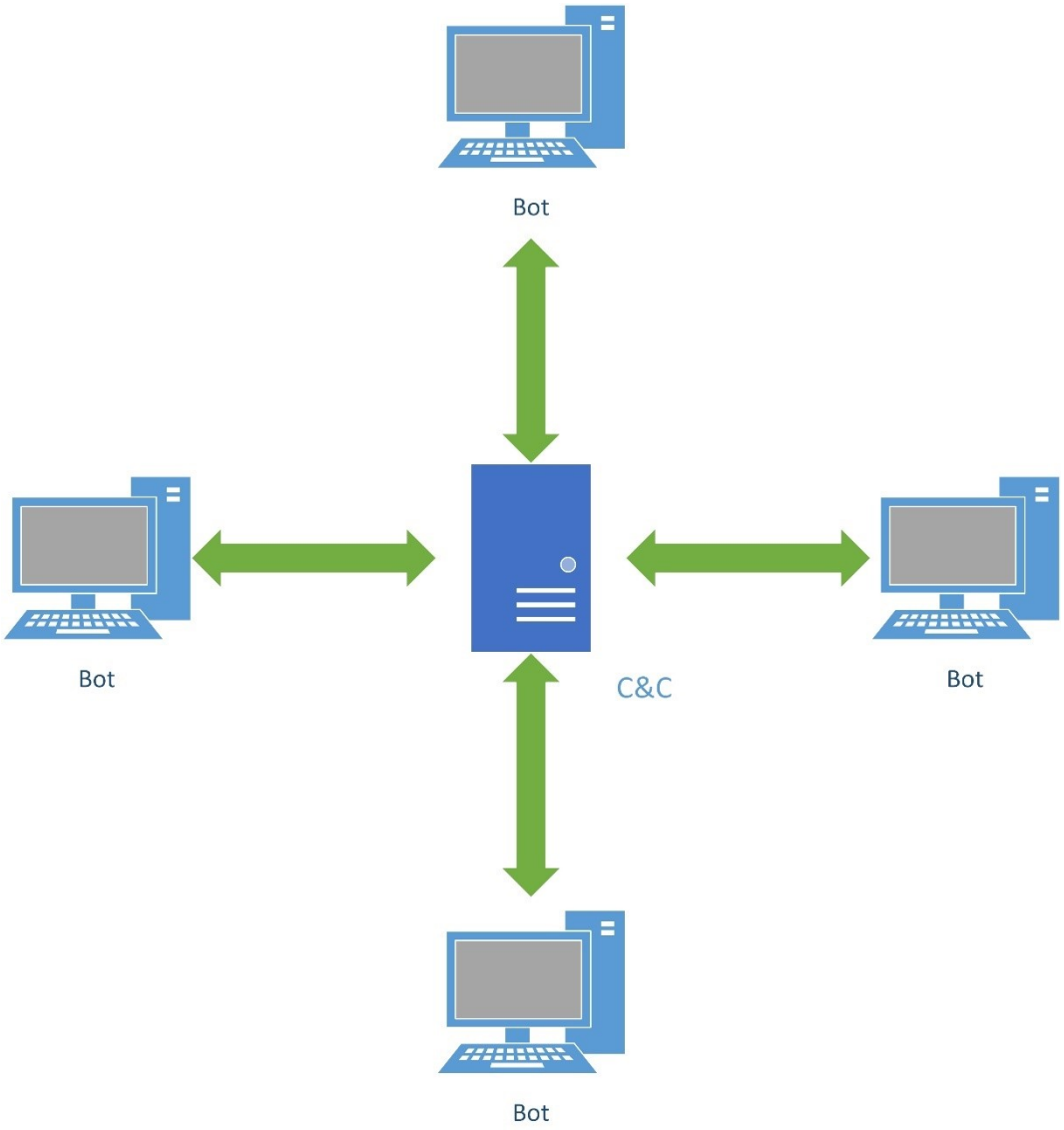
Robotvõrgus olevate tuhandete robotite võimsust ühendades on võimalik ummistada ligipääs erinevatele internetiteenustele. Selle näiteks on IRC võrkudele suunatud kloonirünnakud, mis koormavad üle võrguressurssi, tõkestades võrgu kasutamise. Sarnaselt viiakse läbi ka DDoS rünnakuid. Tuhanded robotid saadavad sihtmärgi pihta lühikese aja jooksul suure hulga päringuid. Tekkinud ülekoormuse tõttu teenus seiskub ning muutub kasutajatele kättesaamatuks. Ummistusrünnakuid kasutatakse küberväljapressimiseks. Suured firmad ning veebilehed on pigem nõus maksma lunaraha kui riskima võimalusega, et teenuse või veebilehe kättesaamatuse tõttu kaotatakse usaldust või tulu. [15]

3.3 Robotvõrkude topoloogiad

Robotvõrgud jagunevad arhitektuuri poolest viieks: tähekujuline, multiserveriline, mitmekihiline, ebakorrapärane ning hübriidne. Nendest esimesed kolm kasutavad oma tegevuse koordineerimiseks keskserverit. Viimased kaks, ebakorrapärane ning hübriidne lahendus, keskserverit ei kasuta, vaid saavad korraldused otse *botmaster*'ilt. Kasutatava robotvõrgu disain valitakse vastavalt omaniku vajadusele ning riskialdisusele. Omaniku jaoks peab arhitektuur olema vastuvõtlik igasugustele vastumeetmetele ning varjama maksimaalselt oma tegevust See lubab robotvõrgu paremat käideldavust ning suuremat kasu omanikule. [21] ,[22]

3.3.1 Tähekujuline arhitektuur

Levinuim ning tänu oma lihtsusele siiamaani laialdaselt kasutuses olev robotvõrgu arhitektuur on tähekujulise arhitektuuriga robotvõrk (vt Joonis 7). Antud lahenduses suhtlevad robotid ühe keskserveriga, mille käest nad saavad käsklused ning juhised. See lubab keskserveril olla kõigi oma robotitega ühenduses samal ajal. Tähekujulise arhitektuuri peamine puudus on toetumine ühele serverile. Selle seiskumine toob kaasa robotvõrgu osalise või täieliku juhitamatuse ning töövõime languse. [21] ,[22]

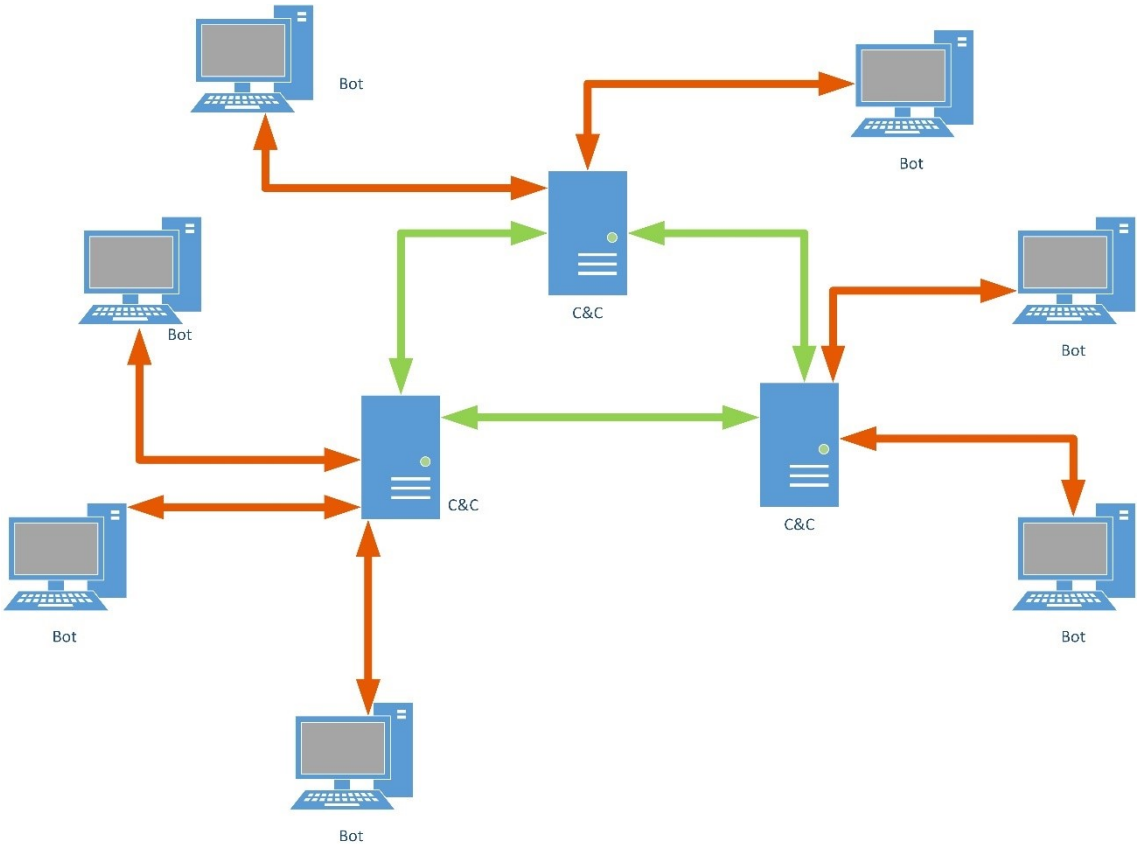


Joonis 7. Tähtvõrguline robotvõrk

3.3.2 Multiserveriline arhitektuur

Multiserveriline arhitektuur (vt Joonis 8) on loogiline täiendus tähekujulisele arhitektuurile. Multiserverilise lahenduse puhul annavad robotvõrgus robotitele käsklusi edasi mitu keskserverit. Samal ajal vahetavad ka keskserverid omavahel informatsiooni. See on mõistlik lahendus juhuks, kui üks serveritest peaks lakkama töötamast või eemaldataks täielikult võimude poolt. Sellisel juhul suudab robotvõrk ikkagi oma tööd jätkata, erinevalt tähekujuliselt arhitektuurist. Arhitektuuri puuduseks on võrreldes tähekujulise arhitektuuriga planeerimisele ning ülesseadmisele kuluv aeg. Positiivse omadusena saab multiserverilise lahenduse plokke jagada väiksemateks, tähekujulise arhitektuuriga robotvõrkudeks. [21]

Robotite geograafilisest asukohast lähtuvalt on võimalik robotvõrk jagada nii, et sama regiooni robotid saavad oma käsklused keskserverilt, mis asub nendega samas regioonis. See lubab neil kiiremini kommunikeeruda ning liiklust optimeerida, erinevalt tähtkujulisest robotvõrgust, kus kogu võrgu peale on ainult üks server. Lisaks aitab selline jaotus muuta robotvõrgu vastuvõtlikumaks juhul, kui võimud peaks andma korralduse mõne keskserveri väljalülitamiseks. [21]

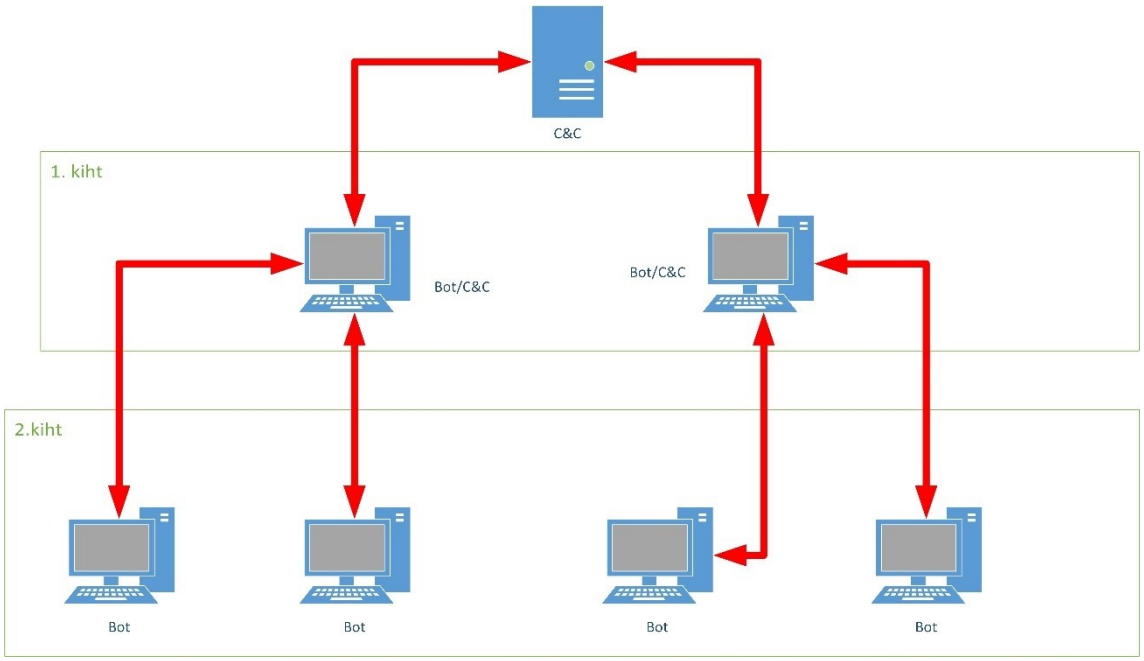


Joonis 8. Multiserveriline robotvõrk

3.3.3 Mitmekihiline robotvõrk

Mitmekihiline või hierarhiline arhitektuur (vt Joonis 9) on moodustatud mitmest keskserverist, mis on paigutatud topoloogiliselt kihiti. Iga saadetud sõnum läbib mitu kihti. Iga kiht käitub talle järgnevale kihile kui proksiserver. Selle tulemusena ei ole ükski robot teadlik terve robotvõrgu asukohast ning ei oma informatsiooni, kus on talle saadetud sõnumi algallikas. See muudab keskserveri asukoha väljauurimise ning sulgemise raskemaks. Samuti lubab selline seadistus robotvõrgu jaotada osadeks ning kasutada neid erinevates ülesannetes. [21]

Ainukeseks puuduseks antud arhitektuuri juures on sõlmedevahelise kommunikatsioonikiirus. Tänu mitmele kihile jõuavad sõnumid kohale viivitusega ning see muudab raskeks läbi viia tegevusi reaalajas. [22] ,[23]



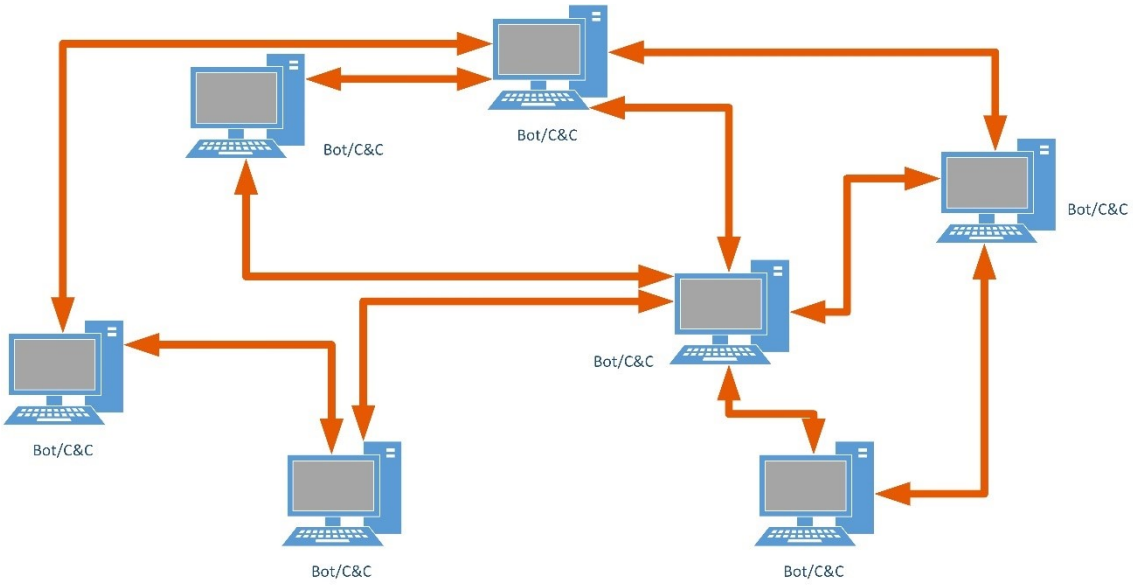
Joonis 9. Mitmekihiline robotvõrk

3.3.4 Ebakorrapärane arhitektuur

Ebakorrapärase arhitektuuri juures ei kasutata keskset juhtserverit (vt Joonis 10). Käsklused viiakse robotvõrku otse läbi *botmaster*'i ning robotid levitavad seda omavahel edasi. Käskluste omavahel jagamiseks kasutavad robotid P2P protokolle. [21]. Keskse serveri puudumisel ning robotitevaheliste kommunikatsiooniteede paljususe tõttu on robotid raskesti välja uuritavad. Samas, jälgides ühe roboti suhtlemist teistega, on võimalik välja uurida teised robotvõrgu liikmeid. [21]

Antud robotvõrk suudab oma tugevuse säilitada, kui üks robotitest peaks lõpetama töötamise. Keskserveri puudumise suurim tugevus seisnebki selles, et *botmaster* võib ennast ühendada võrdvõrgu kaudu roboti külge ning anda uue käskluse. Kuna selleks võib olla ükskõik milline võrku kuuluv robot, on *botmaster*'i jälitamine peaaegu võimatu. Suurimaks puuduseks antud robotvõrgu juures võib elastsusele vaatamata lugeda keerukat arhitektuuri ning kommunikeerumise teede paljususest tulenevat käskluste edastamise kiirust. Lisaks eelmainitule võib kasutatavast marsruutimise protokollist

olenevalt võrdvõrk mõraneda, näiteks ei ole enam võimalik ühe roboti kaudu kätte saada ülejäänud robotvõrgus olevaid roboteid. [22]



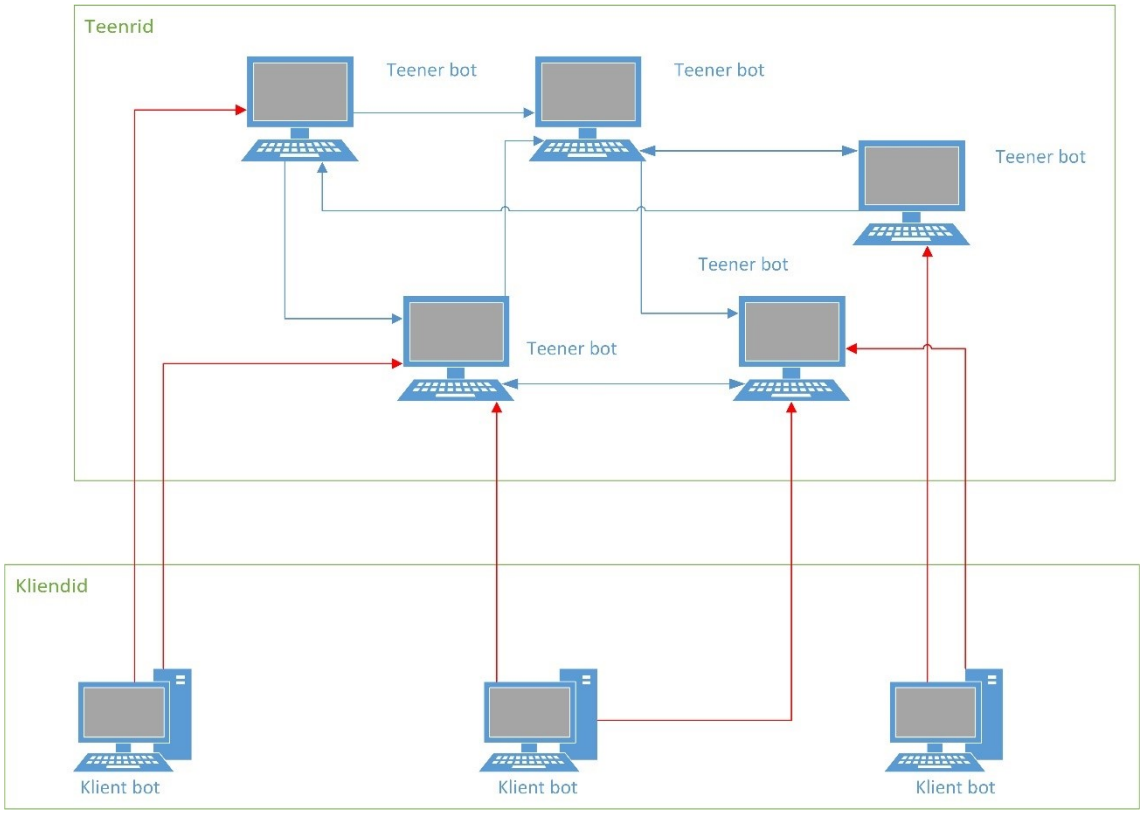
Joonis 10. Ebakorrapärane robotvõrk

3.3.5 Hübriidarhitektuur

Hübriidne topoloogia ühendab endas nii detsentraliseeritud kui ka tsentraliseeritud robotvõrkude arhitektuuride osad (vt Joonis 11). Selle lahenduse puhul grupeeruvad robotid kaheks: teener- ning klientrobotiteks. Esimesse gruppi kuuluvad robotid käituvad nii serveri kui ka kliendina. Neil peab olema staatiline ja avalik IP-aadress ning neile peab ligi saama avalikust internetist. Teise gruppi kuuluvad robotid, millel on dünaamiline ja privaatne IP-aadress ja mis asuvad tulemüüri taga nii, et neile ei saa ligi pääseda avalikust võrgust. Teist gruppi nimetatakse kliendiks, sest nad ei võta vastu sissetulevaid ühendusi. Ainuke ühendus luuakse teener- robotitega, kes on nende *peer*'ide nimekirjas. Kõik robotid, olenemata grupist, võtavad aktiivselt ühendust teener-tüüpi robotitega, et saada uusi korraldusi. [24] ,[25]

Hübriidtopoloogiaga robotvõrgu tugevusteks võib lugeda *botmaster*'i võimekust juhtida käskluste voogu ning kasutada tehtavates rünnetes täpselt vajaminevat arvu roboteid. Seda võimaldab hübriidse robotvõrgu omadus saata käsklusi läbi teener robotite, mitte otse klient robotite. Lisaks vähendab taoline puhver tõenäosust, et *botmaster*'i avastab mõni meepurk-tüüpi robot. Teiseks tugevuseks on, et robotvõrgus olevad teener robotid kasutavad staatilist IP-aadressit, suurendades sellega robotvõrgu stabiilsust, usaldatavust

ning töökindlust. Arvestades neid tugevusi, on sellist arhitektuuri kasutavat robotvõrku raskem monitoorida, kaaperdada või sulgeda. [26]



Joonis 11. Hübriidrobotvõrk

3.4 Robotvõrkude kommunikatsioon

Robotvõrgu osad kasutavad omavahel kommunikeerumiseks erinevaid protokolle ning mooduseid. Neist levinumad on IRC ja HTTP protokolli kasutamine. Viimasel ajal on kasvavas trendis robotvõrgud, mis kasutavad P2P protokolle. Samuti kasutavad robotvõrgud kommunikeerumiseks alternatiivsemaid mooduseid.

3.4.1 Internet Relay Chat

Internet Relay Chat (IRC) on rakenduse kihis kasutatav ning TCP/IP-il põhinev avatud protokoll. Tema eelkäijaks on *Bitnet Relay Chat*. [27] ,[28]

IRC on tekstipõhine telekonverentsisüsteem, mis töötab klient-server arhitektuuril ning on tänu sellele suuteline töötama paljudes erinevates masinates. Tüüpiline IRC süsteem koosneb keskesest serverist, mis on sõlmpunktiks klientidele või teistele serveritele. Serveri ülesanneteks on sõnumite edastamine ning teised funktsioonid. [29]

Server on IRC-s punktiks, millega kliendid ühenduvad, et üksteisega rääkida. Serverid võivad ühineda ka üksteisega, moodustades sellega IRC võrgustiku. Ainuke lubatud võrguseadistus, mis IRC-le lubatud on, seisneb sellest, et iga server näib võrgupuus ülejäänud võrgule kui keskne sõlmpunkt. [29]

Klient: IRC kontekstis nimetatakse kliendiks kõiki neid osalisi, kes on ennast serveriga ühendanud, kuid ei ole serverid. Üksteise eristamiseks on igal kliendil unikaalne kasutajanimi. Lisaks kasutajanimele peab server omama kõikide klientide kohta järgmist informatsiooni: kliendi võõrustaja tegelikku nime, kliendi kasutajanime võõrustajas ja serveris, mille külge klient on ühendatud. Klientidel on olemas ka eraldi klass nimetusega operaatorid. Nende klientide ülesandeks on IRC võrgus korda hoida. [29]

Kanal: gruppi, kus on üks või rohkem sõnumeid saatvat klienti, nimetatakse kanaliks. Klientide poolt saadetud sõnumid on vastavale kanaline adresseeritud. Kanal luuakse niipea, kui esimene klient sellega ühendub, ning lakkab töötamast niipea, kui viimane klient lahkub sellest. Senikaua, kuni kanal töötab, võib iga klient sellele kanalile viidata, kasutades selleks kanali nime. Kanali nimed on sõnetüüpi ning võivad koosneda kuni 200 sümbolist. Kanali nimed peavad hakkama sümboliga & või #. Lisaks sellele ei tohi kanali nimi sisaldada koma ega tühikut. [29]

Serverid ja kliendid saadavad üksteisele sõnumeid, mis võivad, kuid ei pea vastust saatma. Kui sõnum sisaldab korrektset käsklust, peaks klient saama selle kohta vastuse. Kuid soovitatav ei ole oodata kaua, sest kliendi ja serveri ning sellele vastupidine kommunikatsioon on sisuliselt asünkroonne. [29]

Iga IRC sõnum sisaldab endas kolme peamist osa: valikulist eesliidet, käsklust ning kuni 15 käskluse parameetrit. Eesliides, käsklus ning kõik parameetrid on eraldatud ASCII süsteemis tühikuga (0x20). [29]

IRC sõnumite näiteid:

Ühendumine kanaliga:

/join#demokanal [30]

Tegevuse tegemine (lehvitamine):

/me waves hello [30]

Privaatvestluse alustamine:

/msg <kasutajanimi> Tere, alustame vestlust! [30]

Hüüdnime vahetamine:

/nick <uus soovitud nimi> [30]

Populaarsemad IRC võrgud kasutajate arvust lähtuvalt on freenode, IRCnet, QuakeNet, EFnet, Rizon, Undernet, ChLame, IRC-Hispano, OFTC ja LinkBr, mille kasutajate arv jääb 8000 ja 60 000 vahele. [27] ,[31] ,[32]

3.4.2 HTTP

HTTP on rakenduse kihis kasutatav protokoll, mida kasutab WWW(World Wide Web). HTTP määrab, kuidas sõnumeid koostatakse ja edastatakse ning millised tegevusi peavad veebiserverid ning veebilehitsejad tegema vastuseks erinevatele päringutele. [33]

HTTP on TCP/IP-il põhinev kommunikatsiooniprotokoll, mida kasutatakse erinevate andmete edastamiseks WWW-sse. HTTP kasutab vaikimisi porti TCP 80, kuid võib kasutada ka teisi porte. Protokoll kasutab päring/vastus loogikat, mis põhineb klient-server arhitektuuril, kus veebilehitsejad, robotid ja otsingumootorid käituvad kui kliendid ning veebiserver kui server. Kliendi ja serveri vaheline tööprotsess koosneb kahest osast: päring ja vastus. [34]

Päring tuleb kliendi poolelt. Klient saadab serverile päringu, mis koosneb päringu meetodist, URI-ist ja protokolli versioonist[5]. Sellele järgneb MIME-tüüpi sõnum, mis koosneb päringu teisendajatest, kliendi informatsioonist ning sisust. [34]

[5] (HTTP'l on kasutusel kaks versiooni, HTTP/1.0 ja HTTP/1.1. Esimene loob uue ühenduse iga päringu/vastuse vahetuse korral. Teise puhul võidakse sama ühendust kasutada rohkem kui ühe päringu/vastuse puhul). [34]

Vastus: tuleb serveri poolelt. Vastuseks saadab server seisundi rea, mis koosneb sõnumi protokolli versioonist ning õnnestumise või veateatest. Sellele järgneb MIME sõnum, mis koosneb serveri ja metainformatsioonist ning võimalusel ka sisust. [34]

HTTP on omaduste tõttu lihtne, kuid võimekas võrguprotokoll. Nende omaduste hulka kuuluvad:

1. staatuseta olek. Iga käsklus teostatakse iseseisvalt, teadmata eelnevaid käsklusi. Selle tõttu on raske luua kasutaja sisendile intelligentselt reageerivaid veebilehti. Seda vajakajäämist üritatakse lahendada, kasutades selle tehnoloogiaid, näiteks ActiveX, Java ja Javascript. Samuti kasutatakse kasutaja tegevuste salvestamiseks „küpsiste“ abi; [34]

2. andmetüübist sõltumatus. HTTP kaudu saab saata palju erinevat tüüpi andmeid. Tähtis on vaid see, et mõlemad, nii serveri kui kliendi pool oskab andmeid töödelda ning määraks ära korrektse MIME tüübi; [34]

3. ühendusteta olek. Peale veebilehitseja poolt tehtud päringut lõpetab klient (veebilehitseja) ühenduse serveriga ning jääb ootama vastust. Server töötleb päringut, taastab ühenduse kliendiga ning saadab omapoolse vastuse tagasi. [34]

3.4.3 P2P protokollide perekond

Võrdvõrk on detsentraliseeritud kommunikeerumise mudel, milles kõik osapooled on õiguste poolest võrdsed. Erinevalt klient-server mudelist, kus rollid on ära jaotatud, võib iga võrdvõrgu liige käituda nii serveri kui ka kliendina. [35]

Võrdvõrgu süsteeme saab kasutada anonüümseks marsruutimiseks, massiivsete paralleelarvutuste jaoks, andmesalvestuseks ning muudeks funktsioonideks. Suurem osa võrdvõrgu programmidest on mõeldud meedia jagamiseks ning tänu sellele seostatakse neid autoriõiguste rikkumise ning tarkvarapiraatlusega. Tavaliselt lubab võrdvõrgu rakendus kasutajal hallata suurel hulgal erinevaid parameetreid, mis on seotud võrdvõrgu kasutamisega, nt. ühenduste hulk väljuva ning sissetuleva liikluse korral, milliste süsteemide külge ennast ühendada, milliseid teenused pakkuda ning palju süsteemset ressurssi toimingute jaoks loovutada. Samas on olemas ka süsteeme, mis lihtsalt loovad ühenduse mõne aktiivse sõlmega ning annavad kasutajale väiksel hulga võimalusi seadete muutmiseks. [35]

Kuigi võrdvõrgu kontseptsiooni on juba uuritud alates ARPANET'i aegadest, võttis laiem üldsus võrdvõrgu kasutusele alles 1990-ndate aastate lõpus, kui esile kerkisid muusikajagamise rakendused Napster ning talle järgnenud Gnutella. [35]

Traditsiooniliselt kasutatakse võrdvõrke erinevate failide jagamiseks, näiteks dokumendid, audio- ja videofailid, elektroonilised raamatud. Viimaselt aegadel on sinna lisandunud ka rakendused, mis lubavad võrdvõrku kasutades läbi viia erinevaid ülekandeid, võrgumänge ning konverentse. [36]

Võrdvõrgud jagunevad struktuuri poolest kaheks: Struktureeritud ning mittestruktureeritud.

Struktureeritud võrdvõrgu lahendus. Selles süsteemis on sõlmedevaheline ühendusete arv fikseeritud ning osapooled säilitavad endas informatsiooni andmete kohta, mida nende naaber osapool hoiab. See on efektiivne lahendus, sest andmete päringut on võimalik kohe suunata sinna, kus vastavad andmed asuvad, olenemata nende rohkusest. [36]

Süsteemis olevate osapoolte indekseerimiseks on kõige levinum viis DHT. (*Distributed Hash Tables*). Sarnaselt tavalise räsitabeliga, pakub DHT teatmeotsingut võtme ja väärtuse paaride abil. Igal võrdvõrku kuuluval liikmel on võimalik see paar endale hankida. Taolist tehnoloogiat esindavad Chord, Pastry, Tapestry, Kademlia ning CAN protokollid. [36]

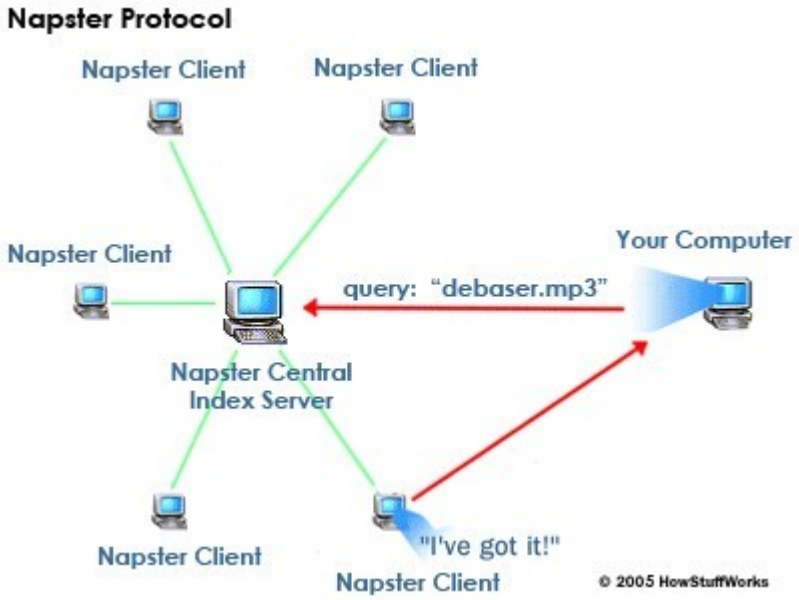
Kademlia on rakendatud näiteks populaarse Bittorrenti protokolli juures, lubades faile jagada ka ilma jälgurit kasutamata [37] .

Struktureerimata võrdvõrgud. Antud struktuur jaguneb omakorda kolmeks alaarhitektuuriks: tsentraliseeritud, hübriidseks ning detsentraliseerimata võrdvõrguks. Erinevalt struktureeritud võrdvõrgust on ühendused osapoolte vahel loodud ühetasandiselt või hierarhiliselt. Leidmaks võimalikult palju osapooli, kellel on vajaminevaid andmeid, kasutab struktureerimata võrdvõrk erinevaid tehnikaid nagu, ummistamine, "*random walking*" ning laiendamine. [36]

Tsentraliseeritud struktureerimata võrdvõrk.

Antud arhitektuur kasutab osapoolte indekseerimiseks ning kogu robotvõrgu käivitamiseks keskset serverit. Masin hoiab endas andmebaasi, milles asub informatsioon kõikide osapoolte ning nende poolt jagatavate andmete kohta. Lahenduse suurimaks puuduseks on sõltumine kesksest serverist. Selle mahakukkumisel lakkab kogu võrdvõrk töötamast. Probleemiks saab lugeda ka antud võrdvõrgu liikluse haldamisest ning liikluse kogusest tulenevaid kitsaskohti. Sellele olukorrale pakuks lahendust taristu laiendamine, mis toob endaga kaasa lisakulutusi. [36]

Sellist arhitektuuri kasutas Napsteri-nimeline muusikajagamise rakendus [38] (vt Joonis 12 [39]).



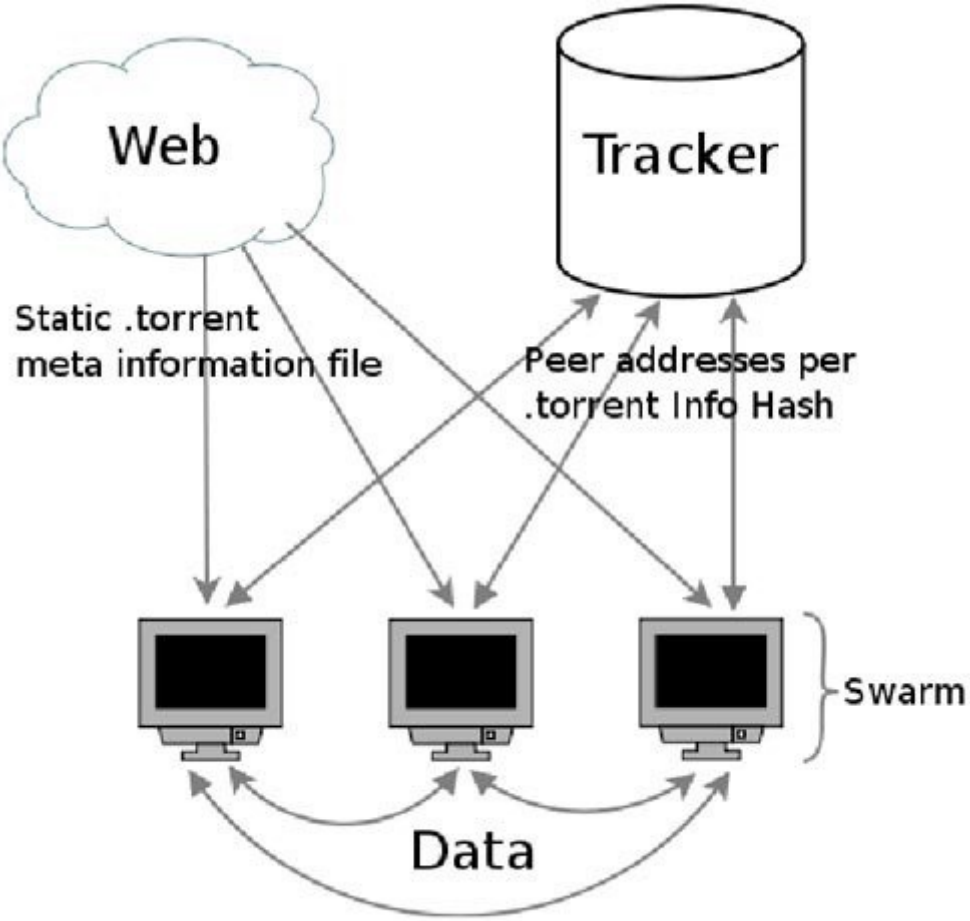
Joonis 12. Napsteri protokoll

Antud arhitektuuri, tsentraliseeritud struktureerimata võrdvõrgu, alla arvatakse ka Bittorrenti protokoll. Bittorrent töötab P2P baasil. Faili jagatakse nii, et algupärane jagaja saab ribalaiuse kasutust vähendada, samas jõuda sama arvu inimesteni. [40]

Sama faili soovivatest *peer*'idest moodustatakse parv. Fail jagatakse juppideks, mis jaotatakse erinevate *peer*'ide vahel ära. *Peer*, kes on alla laadinud terve faili, võib teha selle kättesaadavaks ka teistele parve liikmetele, muutudes sellega *seeder*'iks. Parve tugevuse määrab ära kättesaadavus, mis Bittorrenti mõistes tähendab tervete failide komplektide arvu, mis parve kaudu kättesaadavad on. Kui vastav väärtus on 1 või suurem, on selle parve kaudu võimalik kätte saada kogu vajaminev fail. Üksteise leidmiseks

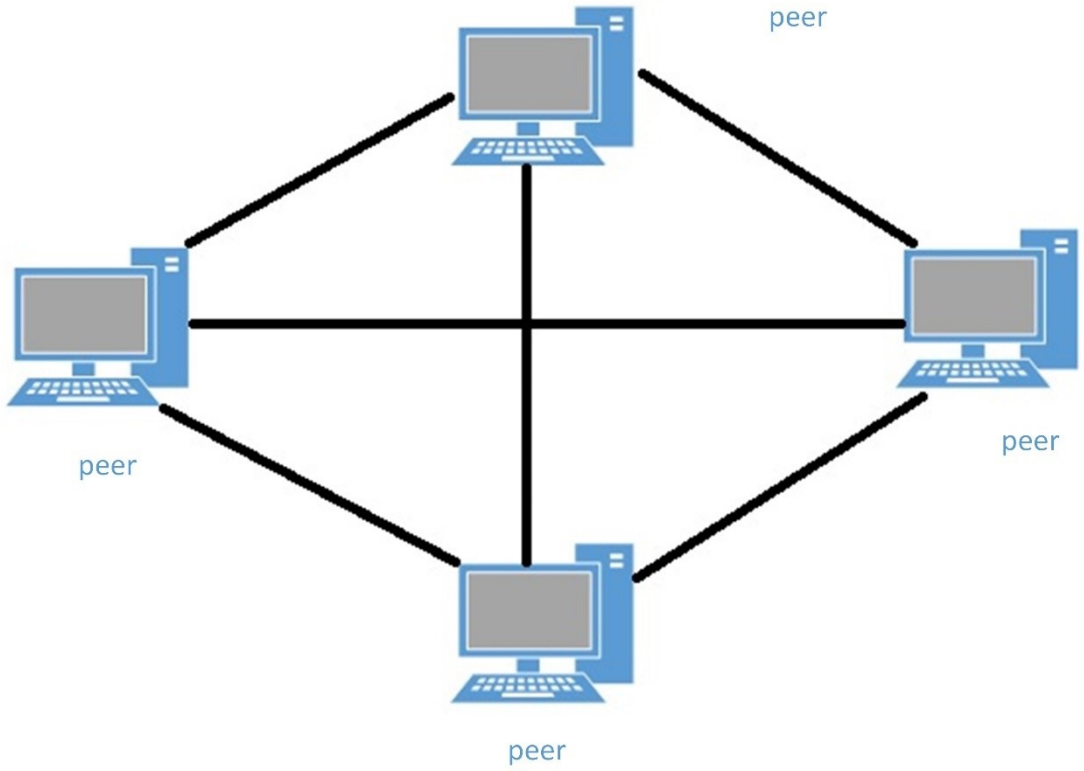
kasutatakse jälgurit, kuhu külge *peer*'id saavad ühenduda ning kätte saada teiste *peer*'ide IP-aadressid. Iga parve jaoks peab jälgur salvestama *peer*'i IP-aadressi ning pordi numbri, mida jagatakse teiste, parve kuuluvate *peer*'idega (vt Joonis 13 [41]). [40] ,[42]

Põhiline informatsioon: parve ning faili puudutav metaandmestik on kirjas torrent-failis. Selle faili loob parve initsiaator. Parvega liitumiseks käivitab uus osapool torrent-faili ning meta-andmestiku kasutades, liitub parvega. Iga torrent-fail on varustatud ka 160-bitise SHA1 räsiga. Selle räsi abil on võimalik saada kätte parve info, kui torrent-fail ei ole (mingil põhjusel) kättesaadav. [42]



Joonis 13. Bittorrent

Detsentraliseerimata võrdvõrk on oma arhitektuurilt ülekatte võrk (vt Joonis 14). Võrgu otspunktid moodustavad omavahel osapoolte paari, kelle vahel luuakse TCP-ühendus. Kuna antud võrgustruktuur on ehituselt ühetasandiline, on kõik osapooled omavahel võrdsed. Detsentraliseerimata võrdvõrgus ei ole kasutusel tsentraalset serverit vms taristut. [36]

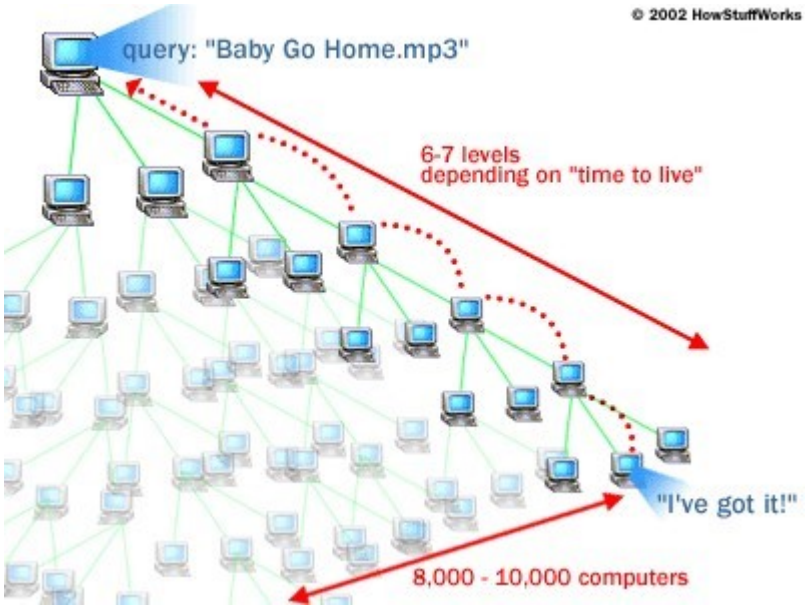


Joonis 14. Detsentraliseeritud võrdvõrk

Detsentraliseeritud struktueerimata võrdvõrgu lahendust kasutab Gnutella-nimeline protokoll (vt Joonis 15 [39]). Võrguga liitumiseks, ühendab kasutaja ennast kõigepealt mõne teadaoleva algosapoolega. Järgnevalt annab algosapool informatsiooni ühe või rohkema osapoole kohta, kes asub antud ülekatte võrgus. Informatsiooni hulka kuuluvad kasutatavad pordid ning IP-aadressid, mida osapooled kasutavad. Gnutella näitel on osapooled teadlikud ainult oma naaberosapooltest. Ülekatte võrgus on osapooled omavahel ühenduses läbi ühise "joone". Päringud, mis võrku tulevad, on osapoolte vahel jagatud. Selleks kasutatakse erinevaid "üleujutuse" metoodika variatsioone. [36]

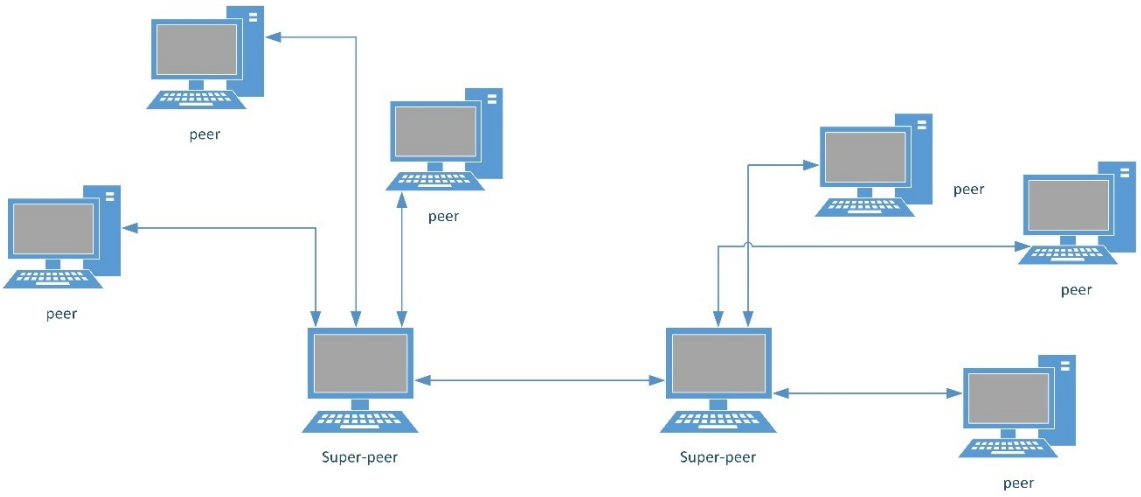
Osapool, kes on huvitatud teatud andmetest, saadab vastava päringu oma võrdvõrgus olevatele naabritele. Järgnevalt saadavad naabrist osapooled selle omakorda edasi oma naabritele. Protsess jätkub, kuni päring ületab otsingu limiidi, mis on määratud *Time-To-Live* loenduriga. Osapool, kellel on päringus soovitud andmed olemas, saadab sellekohase

vastuse päringu tegijale. Vastuse saatmiseks kasutatakse vastupidist päringut, mis kasutab esimeses etapis loodud TCP ühendusi. Peale vastuste saamist valib päringu tegija välja ühe osapoole, kelle käest ta otsest TCP ühendust kasutades andmed alla laadib. Kuigi Gnutella tööpõhimõte on lihtne, kõrgelt detsentraliseeritud ning ei nõua osapooltel andmete asukoha säilitamist, peetakse tema suurimaks nõrkuseks mitteskaleerumist. See tuleneb osapooltevaheliste päringute arvu lineaarsest kasvamisest koos tehtud päringute arvuga, mis omakorda kasvab võrdvõrgu suurenemisega. Teiseks puuduseks loetakse võimalust, et päringu teinud osapool ei pruugi saada vastust, sest tema palutud andmed on vähelevinud ja raskelt kättesaadavad. [36]



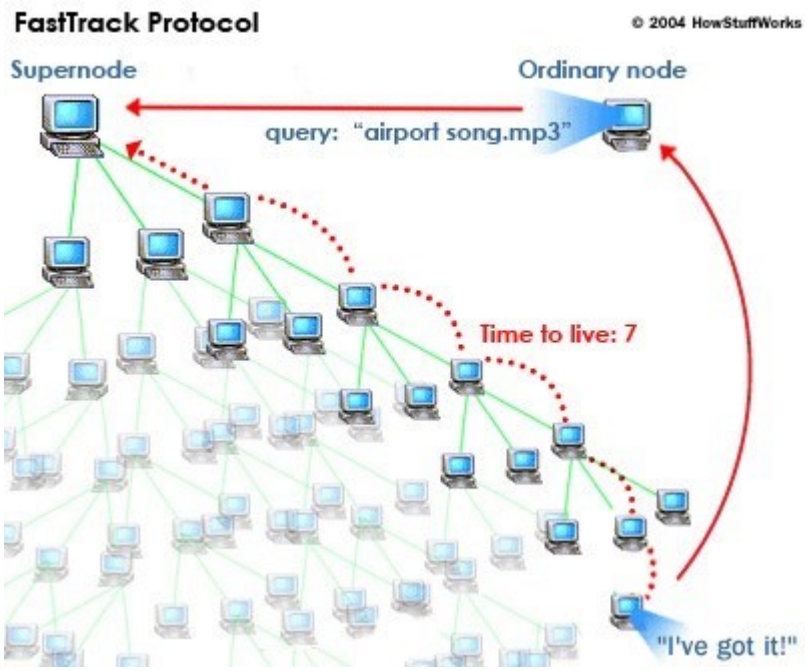
Joonis 15. Gnutella arhitektuur

Hübriidne struktueerimata võrdvõrk lubab enda struktuuris taristulisi sõlmi, mida nimetatakse *super-peer*-ideks (vt Joonis 16). Selline lahendus moodustab hierarhilise ülekattega võrdvõrgu ning on vastuseks puhta struktureerimata võrdvõrgu puhul esinenud skaleerumise probleemidele. Hübriidses võrdvõrgus võivad osapooled rolle vahetada, näiteks võib tavaline osapool muutuda *super-peer*-ideks ning osaleda võrdvõrgu töö koordineerimises. [36]



Joonis 16. Hübriid võrdvõrk

Hübriidset ning struktureerimata võrdvõrgu lahendust kasutab KaZaA-nimeline rakendus, mis põhineb FastTracki nimelisel protokollil (vt Joonis 17 [43]). Antud võrdvõrk kasutab spetsiaalselt loodud *super-peer*'i, kellel on suur ribalaius, suur andmesalvestusvõimekus ning suur andmetöötlusvõimekus. [36]



Joonis 17. FastTrack protokoll

Peer'i liitumisel võrdvõrguga ühendatakse ta kõigepealt *super-peer*'i külge. Seejärel annab ühendatud *peer* informatsiooni selle kohta, milliseid andmeid ta tahab teistele *peer*'idele jagada. *Super-peer*'i roll seisneb selles, et ta hõlbustab otsingut, luues ning hallates andembaase, mis sisaldavad informatsiooni *peer*'ide ning nende poolt jagatava

andmete kohta. Andmebaasis olevat informatsiooni kasutab ta *peer*'ide tehtud päringutele vastamiseks. [36]

Sarnaselt tsentraliseeritud ehitusele etendab *super-peer* keskse serveri rolli, kuigi seda ainult määratud *peer*'idele. Osapooled loovad omavahel ülekatte võrgu, mis muudab andmete otsingu palju efektiivsemaks. Päringut tehes suunatakse päring kõigepealt *super-peer*'ile, kes sarnaselt detsentraliseerimata võrguta annab selle edasi oma naaber *super-peer*'ile. Järgnevalt saadab *super-peer* päringu tegijale vastusena nimekirja *peer*'idest, kes vajaminevaid andmeid jagavad. [36]

Hübriidvõrgud ei kasuta enam ühte, keskset serverit, vaid on seal oleva andmebaasi ära jaotanud paljude *super-peer*'ide vahel. Sellest tulenevalt on ka *super-peer*'ide poolt hallatavate andmebaaside suurused suhteliselt väikesed, sest nad hoiavad seal ainult endaga seotud *peer*'ide informatsiooni. Antud lahenduse suurimaks negatiivseks küljeks on keerukus, lisaks sellele on *super-peer*'idel on rohkem kohustusi kui tava osapooltel ning nad võivad seetõttu saada antud võrdvõrgu lahenduse pudelikaelaks. [36]

Peatükkides „Robotvõrkude topoloogiad" ning „Robotvõrkude kommunikatsioon" kirjutatu on kokku võetud järgneval leheküljel asuvasse tabelisse (vt Tabel 1).

Tabel 1. Robotvõrkude võrdlus

Arhitektuur	**Levinud kommunikatsiooni-protokollid**	**Plussid**	**Miinused**
Tähekujuline	IRC,HTTP [24]	Kiirus keskserveri ning robotite vahel. [21]	Keskserveri seisund määrab robotvõrgu funktsioneerimise. [21]
Multiserver	IRC,HTTP [24]	Mitme keskserveri olemasolu lubab robotvõrgul tööd jätkata ka mõne serveri langemisel. Samuti lubab lahendus geograafiliselt optimeerida robotvõrgu tööd. [21]	Vajab pikemat planeerimist [21]
Mitmekihiline	IRC,HTTP [24]	Mitmekihiline olemus ei lase ühe roboti „kättesaamisel“ kätte saada kõiki roboteid ning suure tõenäosusega ei luba ka leida keskserverit. Saab kasutada eri osadena või sektsioonidena ning erinevateks ülesanneteks. [21]	Arhitektuuri kihilisusest tulenev käskluste viivitus [21]
Ebakorrapärane	P2P protokollide perekond [24]	Hajususe ning keskserveri puudumise tõttu on väga vastupidav avastamisele ning kinnipanemisele [21]	Robotite omavahelise suhtlemise monitoorimine suudab tuvastada robotvõrku kuuluvad liikmed. *Ad hoc* viisil kommunikeerumine botide vahel võib tuua kaasa kõrge käskluste viivituse aja. [21]
Hübriid	P2P protokollide perekond, HTTP, IRC [24]	Võrgu ülesehitus ei lase keskserverit kergelt avastada. Kasutatav *peer*'ide süsteem lubab piirata lõimete arvu ning avastamise korral ei leita üles kõiki roboteid. [26]	Keerulisem struktuur kui teistel arhitektuuridel. Keerulisem hallata. [36]

3.4.4 Alternatiivsed kommunikatsioonikanalid

Lisaks traditsioonilistele kommunikatsioonikanalitele kasutavad robotvõrgud ka alternatiivsemaid meetodeid, näiteks piltide metaandmestik, tekstifailid ning sotsiaalvõrgustikud.

JPG-formaadis pildid. Seda tüüpi failid sisaldavad metaandmeid, mis on tuntud ka EXIF nime all. Digitaalsed kaamerad kasutavad seda tüüpi andmeid, säilitamaks informatsiooni erinevate pildiparameetrite kohta, milleks on näiteks pildi tegemisel kasutatud säriaeg, ISO ning ava suurus. Neid andmeid saavad kasutavad ka robotvõrgu koosseisu kuuluvad robotid. Teades JPG faili asukohta ning kasutades sealt saadud metaandmeid, saab robot võtta ühendust keskserveriga ning saada korraldusi järgmisteks tegevusteks. Selle tõttu, et enamik tulemüüre lubab väljuvaid HTTP Port 80 ühendusi, jääb roboti ning keskserveri vaheline suhtlust täiesti märkamatuks. [44]

Microsoft Word failid. Microsoft Wordi *.docx fail on tegelikult kokku pakitud erinevatest failidest, millest üks sisaldab XML metaandmeid. Sarnaselt JPG faili lahendustega kasutab robot, robotvõrguga ühenduse saamiseks teatud kohta üles pandud Wordi faili ning sellega kaasa tulevat metaandmestiku. Sarnaselt JPG näitega lubab enamik tulemüüre väljuvaid HTTP port 80 ühendusi ning keskserveri ja roboti vaheline suhtlus jääb jällegi märkamatuks. [44]

Linkedin.com staatuse sektsioon ning Twitteri postitused. Käskluste edastamiseks robotvõrku võib kasutada ka LinkedIn keskkonnas olevat libakontot. Robotvõrgu keskserver saab kasutada libakonto seisundi sektsiooni, edastamaks robotitele perioodiliselt uusi käsklusi. Tänu Linkedini omadusele lühendada staatuse sektsiooni postitatud URL-i aadresse tekitavad nad robotvõrgu käsklustele veel rohkem kattevarju. [44] Sarnaselt LinkedIn'i näitega kasutavad robotvõrgud suhtlemiseks ka Twitteri libakontosid ning nende alt tehtud postitusi [45] . Kasutades LinkedIni- või Twitteri-suguse veebisaidi ning HTTP Port 80 usaldusväärsust, jätab robotvõrgu erinevate poolte vaheline suhtlus võrguliikluse monitoorijatele neutraalse ning usaldusväärse mulje. [44]

3.5 Nõuded

Selles peatükis kirjeldatakse loodavale süsteemile esitatud nõudeid ning nende kaalukust. Nõuded jagunevad oma kaalukuselt kahte gruppi: soovituslik ning kohustuslik. Soovituslike nõuete täitmine ei ole lahenduse vastuvõtmise juures määrav, kuid tõstab lahenduse väärtust. Kohustuslikute nõuete täitmine on vajalik, sest nende nõuete täitmisega tagatakse lahenduse põhifunktsioonid ning vajalik võimekus. Nõuete kaalukust kasutatakse testimise peatükis (vt Testimine), otsustamaks lahenduse sobivust.

3.5.1 Lahenduse töökindluse tõstmiseks tuleb rakendada liiasust (N1)

Loodava lahenduse töötamine ei tohi sõltuda ainult ühe osa töötamisest. Antud töö kontekstis ei tohi robotvõrgu funktsioneerimine oleneda ühest osapoolest, vaid peab töötamise säilitamiseks rakendama liiasust. Juhtiva keskserveri käideltamatuse esinemisel peab uue lahenduse arhitektuur säilitama robotvõrgu funktsioneerimise. Keskserveri langemise korral võtaks võrgu juhtimise ülesande üle järgmine osapool. Praegu õppustel kasutatava lahenduse funktsioneerimine oleneb ainult ühest keskserverist ning selle langemisel muutub kogu robotvõrk juhitamatuks. Küberõppuste edukaks läbiviimiseks on vajalik, et loodav lahendus oleks kättesaadav kogu õppuste vältel, mis muudab antud nõude kohustuslikuks.

3.5.2 Lahenduse harude vahel liikuvad andmed peavad olema verifitseeritud (N2)

Lahenduse harude vahel liikuvad juhtsõnumite vahetus peab olema verifitseeritud, et oleks tagatud nende terviklus ja autentsus. Selle nõude täitmine on kohustuslik tagamaks, et käsklused erinevate tegevuste rakendamiseks tulevad robotvõrgu haldaja poolt, mitte kolmandatelt osapooltelt.

3.5.3 Võimalus lahenduse muutmiseks (N3)

Loodava lahenduse osad peavad olema loodud OSI (*Open Source Initiative*) ühilduva litsentsi all. Lahendust hakkavad kasutama LS, Küberolümpia ning Eesti Infotehnoloogia Kolledži virtuaallaborid. Nõude muudab kohustuslikuks õppuste ning mainitud keskkondade muudatused ning sellest lähtuvalt ka muudatuste vajadus lahenduses.

3.5.4 Töötamine ette antud operatsioonisüsteemidel(N4)

Loodav lahendus peab toetama erinevaid operatsioonisüsteeme. Sinna hulka kuuluvad Microsoft Windowsi versioonid 7, 8 ning 10, Apple OS X ning levinumad Linuxil põhinevad distributsioonid (Ubuntu kehtiv LTS versioon, Debiani viimane stable versioon). Need operatsioonisüsteemid on õppustel kasutustel ning lahenduse tugi neile on kohustuslik.

3.5.5 Juhtsõnumi liikumine peab toimuma ühtset keskserverit kasutamata (N5)

Lahenduses peab juhtsõnumivahetus toimuma ühtset keskserverit kasutamata. Sõnumivahetus toimub robotvõrgus robotite vahel, mis käituvad nii käsutäitjana kui ka keskserverina. Roboti ülesandeks on tekitada vastavalt käsklustele liiklust sihtmärgi pihta ning samal ajal levitada sama käsklust teistele robotvõrku kuuluvatele robotitele. Nii toimiva lahenduse eelis on sõltumatus keskse serveri funktsioneerimisest. Samuti tõuseb robotvõrgu töökindlus võrreldes hetkel kasutusel oleva lahendusega. Näiteks kui keskserverit kasutataval robotvõrgul elimineeritakse keskserver, muutub see robotvõrk juhitamatuks. Antud nõude kohustuslikus tegemine ning lahenduses rakendamine muudab robotvõrgu töökindlamaks, sest robotvõrku kuuluvad robotid suudavad täita mõlemat rolli, liikme langemisel tema ülesanded üle võtta ning robotvõrgu töö säilitada.

3.5.6 Peab suutma hallata ~ 1000 võrguliikluse tekitajat (N6)

Lahendus peab võimaldama võrguliikluse voo sisse- ja väljalülitamist vastavalt vajadusele. Samuti peab lahendus olema suuteline jooksvalt muutma võrguliikluse voo suurust ning sihtmärke [8] . Nõude täitmine lahenduses on kohustuslik, balansseerimaks võrguliikluse hulka erinevate kaitsva poole meeskondade taristute vahel. Samuti on tähtis võimalus lülitada liiklusvoog välja erinevate tegevuste ning olukordade tõttu, milleks on näiteks ülekoormuse tekkimine või hooldustööd taristus. Hallatavate robotite arv on LS-i näitel aasta-aastalt kasvamas. 2013. aastal oli robotvõrgus liiklust tekitamas 500 robotit. 2014. aasta õppustel oli robotite arv tõusnud 750-ni ning 2015. aastal oli liikluse tekitajate arv ~1000.

3.5.7 Loodud lahendus peab olema kergesti integreeritav olemasoleva lahendusega (N7)

Antud robotvõrku arendatakse kogu aeg edasi, lisades sinna uusi nõudeid ja protokolle. Sellega seoses on robotvõrgu koodibaas pidevas muutumises ning seda tuleks iga uue funktsionaalsuse lisandumisel integreerida ning testida. Seda saab teostada ainult robotvõrgu koodibaasi haldaja. Seetõttu on eelistatum lahendus, mis vajab minimaalset olemasoleva süsteemi muutmist ning on kergelt integreeritav. Loodav lahendus peab olema integreeritav olemasolevasse süsteemi soovituslikult 40 töötunniga. Arvestades õppuste ning töötubade planeerimiseks ning ettevalmistamiseks kuluvat aega, ei ole oluline, et nende sündmuste taustal toimuv lahenduse integreerimine kestaks kauem kui soovituslik aeg.

3.6 Lahenduse arhitektuur

Lahendusele esitatud nõuetest lähtuvalt ei tohi arhitektuur toetuda kesksele serverile ega sellest sõltuda, mistõttu ei saa lahenduses kasutada täht-, multi- ega hargvõrgulist robotvõrgu arhitektuuri. Sobilikeks arhitektuurideks jäid ebakorrapärane ning hübriidne robotvõrgu arhitektuur. (vt Tabel 1)

Mõlemad, nii hübriidne kui ka ebakorrapärane robotvõrgu arhitektuuri realisatsioon kasutatavad sõnumivahetuses P2P protokollide perekonda kuuluvaid liikmeid. Mõlema robotvõrgu arhitektuuri puhul paiknevad liikmed hajutatult, muutes sellega nende avastamise keeruliseks. Samuti on mõlema robotvõrgu arhitektuuri puhul on täidetud nõue, mille järgi peab lahendus olema keskserveri pidevast toimimisest sõltumatu.

Erinevalt hübriidsest arhitektuurist on ebakorrapärane arhitektuur lihtsa ülesehitusega ning tulenevat sellest ka kiiremini üles seatav. Tulenevalt oma lihtsusest on ebakorrapärane arhitektuur ka mobiilsem ning paremini hallatav. Arhitektuuri lihtsusest tingitult valis autor lahenduse realiseerimiseks ebakorrapärase arhitektuuri kasutava robotvõrgu.

Lahenduse realiseerimiseks kaaluti ning võrreldi kahte varianti. Esimene neist on ebakorrapärase arhitektuuriga robotvõrgu andmevahetuse integreerimine olemasolevasse lahendusse. Teiseks variandiks on eraldiseisva juhtsõnumi edastamise süsteemi loomine, kasutades varem valmis olevaid tarkvarakomponente.

Ebakorrapärase arhitektuuriga robotvõrgu lahenduse olemasolevasse lahendusse integreerimise suurimaks puuduseks on olemasoleva lahenduse muutmiseks vajalik ajaline ressurss. Ajakulu tekitavad olemasoleva süsteemi tundmaõppimine ning lisatava funktsionaalsuse arendamine. Mõlema tegevuse peale võib kuluda hinnanguliselt üks kuni kaks kuud, mis on väga suur ajakulu ning ei luba täita nõuet N7 (vt 3.5.7). Samuti võib olemasoleva lahenduse muudatus kaasa tuua muudatuse juhtsõnumi jagamiseks kasutatavas süsteemis.

Teine võimalus on jagada juhtsõnumit uue eraldiseisva lahenduse abil. Lahendus luuakse, kasutades valmis olevaid tarkvarakomponente. See toob kaasa suure ajavõidu, mida võib lugeda antud lahenduse suurimaks plussiks. Teiseks on analoogse arhitektuuriga lahendused kasutusel ka teistes valdkondades, näiteks faili jagamisel erinevatel torrentilehekülgedel. Samuti on iseseisev lahendus vastuvõtlikum muudatustele, mis võivad kaasneda kogu süsteemi muutumisega.

Lahenduse realiseerimiseks valiti teine variant, milleks on uue eraldiseisva lahenduse loomine. Eelised lahenduse olemasoleva süsteemi integreerimise ees on eelkõige realiseerimiseks kuluv aeg. Esimese variandi puhul oleks ajakulu hinnanguliselt mitu kuud, samas uue lahenduse arhitektuur lubaks lahenduse realiseerida hinnanguliselt paari nädalaga. Esimese variandi puhul oleks vaja ebakorrapärasel arhitektuuril põhinev andmevahetus arendada nullist. Enne arendust kulub aega ka olemasoleva süsteemi tundmaõppimiseks. Teise variandi puhul komplekteeritakse lahendus valmisolevatest ning laialdaselt kasutusel olevatest tarkvara komponentidest, mis lubab kokku hoida testimisele kuluvat aega. See muudab lahenduse loomise protsessi tunduvalt kiiremaks ning tõstab töökindlust.

Uues lahenduses kasutatakse juhtsõnumi saatmiseks Bittorrent protokolli. Torrentfaili võrku paiskamine toimub sama protokolli kasutava jälgurarvuti abil, mille ülesandeks on *peer*'ide omavaheline ühendamine. Kui klientarvuti käivitab torrent faili, teeb arvuti päringu erinevates arvutites paikenva faili kohta. Antud arvutid saadavad päringu teinud arvutile igaüks osa failist, mis lõppkokkuvõttes moodustab päritud faili tervikuna. [46]

Torrentfaili alglaadimiseks ning allalaadimiseks kasutatakse Transmissioni-nimelist Bittorrent klienti. Loodud lahenduses kasutatakse jälgurarvuti loomiseks Bitstormi-nimelist avatud lähtekoodiga tarkvara. Võrreldes teiseks variandiks olnud opentrackeri

tarkvaraga oli Bitstormi eeliseks stabiilsem versioon, mis openTrackeril puudus. Samuti on Bitstormi paigaldus on suhteliselt lihtne ning kiire. Kasutatava Bitstormi versiooni juures on võimalik salvestada informatsiooni kuni 1000 *peer*'i kohta. Nõuetest lähtudes on see kogus piisav. Bitstormi järgmine versioon kasutab *peer*'ide haldamiseks MySQL andmebaasi ning lubab hoiustada informatsiooni kordades rohkemate *peer*'ide kohta. [47]

Täitmaks juhtsõnumite verifitseerituse nõuet, kasutatakse lahenduses GnuPG-nimelist krüpteerimise tarkvara. Jälgurarvuti signeerib GnuPG abil jagatava juhtsõnumi ning klient-arvuti, saanud faili kätte, verifitseerib selle. Verifitseerimiseks kasutab klient arvuti varem saadetud jälgurarvuti avalikku võtit.

Kõik lahenduse realiseerimiseks kasutatavad tarkvarakomponendid on tehtud OSI ühilduva litsentsi all, millega täidetakse ära ka nõue N3 (vt 3.5.3).

4 Teostus

Teostuse osas kirjeldab autor lahenduse prototüübi abil teostatavaid etappe, mis on vajalikud tulemuse saavutamiseks ning lahenduse nõuetele vastamise kontrollimiseks. Teostuse etapid on jaotatud ettevalmistavateks ning rutiinseteks tegevusteks.

Lahenduse nõuetele vastamist hinnatakse testimise osas. Testid ning nende tulemused on kirjeldatud testimise peatükis (vt Testimine).

4.1 Ettevalmistavad etapid

Ettevalmistavate etappide alla kuuluvad jälgur- ning klientarvutile vajalike tarkvarakomponentide paigaldamine, verifitseerimiseks vajaminevate võtmete genereerimine jälgurarvuti poolt ning avaliku võtme jagamine ning paigaldamine klientarvutile.

4.1.1 Jälgurarvuti seadistamine

Juhtsõnumi ja andmete levitamiseks robotvõrku on kõigepealt vajalik paigaldada ühte arvutisse jälgurtarkvara, muutes arvuti jälgurarvutiks. Jälgurtarkvaraks on kasutusel vabavaraline Bitstorm ning torrentfaili levitamiseks kasutatakse Transmissioni-nimelist Bittorrent klienti.

Esmalt on vajalik alla laadida skriptid *tracker.sh* ning fail nimega *ui.php*, mis asuvad repositooriumis, mis asuvad internetiaadressil: https://github.com/osoomuk/P2PScript

Skript *tracker.sh* koosneb kahest osast. Skripti esimene osa kontrollib, kas masinasse on paigaldatud jälguri tarkvara. Kui tarkvara ei ole, paigaldatakse see automaatselt. Skripti teises osas kontrollitakse ning vajadusel paigaldatakse tarkvara, mille abil luuakse torrent fail ning käivitatakse faili jagamine teistele arvutitele.

Skripti allalaadimise õnnestumisel tuleb veenduda, et allalaetud skript ning php-fail asuvad ühes kaustas. Samuti võib tekkida vajadus failide ümbertõstmiseks. Failide ümbertõstmiseks saab kasutada käsurea käsklust või kasutada graafilist kasutajaliidest.

Käsurealt asukohta muutes tuleb liikuda asukohta, kus failid asuvad, ning sisestada järgmine käsklus:

```
mv tracker.sh ui.php <soovitud kausta asukoht>
```

Jälgur tarkvara paigaldamiseks tuleb liikuda kausta, kus alla laetud failid paiknevad. Käsureale tuleb sisestada käsklus

```
cd <skripte sisaldav kaust>
```

ning käivitada skript *tracker.sh* juurkasutaja õigustes, seejärel lisada sinna juurde torrenti abil jagatava kausta nimi, soovitud torrentfaili nimi ning jälguri aadress. Jälguri aadressiks on arvuti IP-aadress koos kasutusel oleva pordi numbri ning jälgurtarkvara skriptiga. Skript käivitatakse käsurealt järgneva käsklusega:

```
sudo bash tracker.sh <jagatavad failid> <loodavad torrent faili nimi> <jälguri aadress>
```

Näiteks:

```
sudo bash tracker.sh viirus.sig viirus.txt viirust.torrent https://192.168.0.101:80/ui.php
```

Tarkvara paigaldamine toimub ainult skripti algupärasel käivitamisel. Järgnevatel kordadel kasutatakse *tracker.sh* skripti ainult torrentfaili loomiseks.

4.1.2 Failide verifitseerimine

Tagamaks juhtsõnumi päritolu õigsust, peab torrentfaili jagaja selle signeerima ning vastuvõttev pool verifitseerima. See samm on vajalik veendumaks, et juhtsõnum pärineb *botmaster*'ilt, mitte kolmandalt osapoolelt. Eeltingimusena peab verifitseeriv arvuti (klient) omama *botmaster*'i avalikku võtit. Signeerimiseks ning verifitseerimiseks kasutatakse GnuPG-nimelist krüpteerimise tarkvara.

Serveripoolsed tegevused:

Botmaster'i-poolne võtme genereerimine:

gpg --gen-key

Avaliku võtme eksportimine:

gpg --export <botmaster'i meiliaadress> avalikuvõtmefail.asg

4.1.3 Robotvõrgu liikme seadistamine

Seadistamise faasis paigaldatakse klientarvutile juhtsõnumi allalaadimiseks vajalikud tarkvara- komponendid ning lisatakse jälgurarvuti poolt tulnud avalik võti, mida kasutatakse verifitseerimiseks.

Juhtsõnumi alla laadimiseks ning verifitseerimiseks on kõigepealt vaja alla laadida skript loading.sh, mis paigaldab eelpool mainitud tegevuste teostamiseks vajalikud tarkvara komponendid.

Skript *loading.sh* asub aadressil: https://github.com/osoomuk/P2PScript/loading.sh
Allalaadimiseks kasutada veebilehitsejat või käsurida:

**wget --directory-prefix=<soovitud allalaadimise koht>
https://raw.githubusercontent.com/osoomuk/P2PScript/master/loading.sh**

Liikuda asukohta, kus asub alla laetud skript ning käivitada see juurkasutaja õigustes:

sudo bash loading.sh

Jälgurarvuti avaliku võtme importimiseks on vajalik see kõigepealt alla laadida. Kasutada võib veebilehitsejat või käsurida:

wget --directory-prefix=<soovitud allalaadmise koht><interneti aadress, kus avalik võti asub>

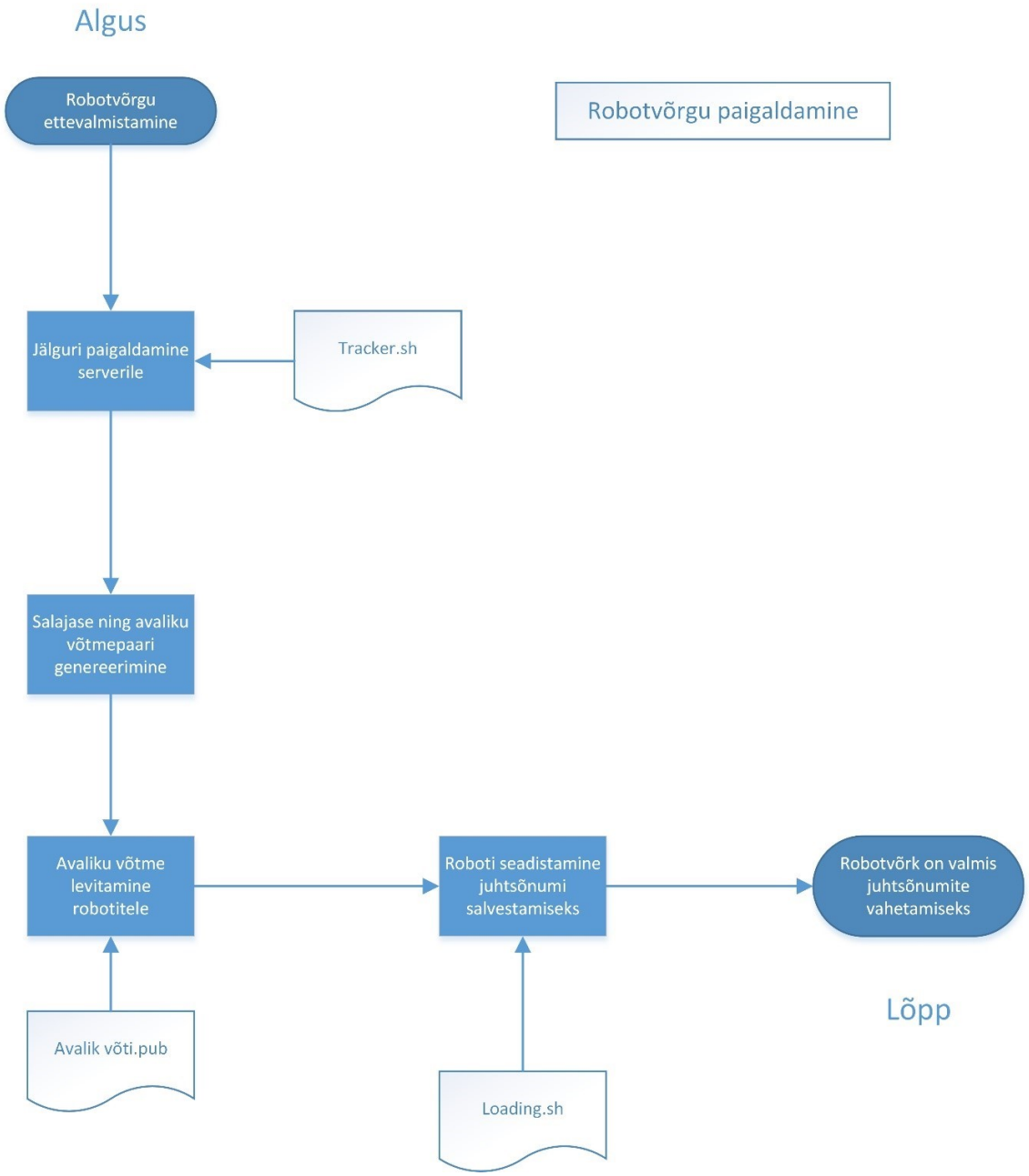
Avaliku võtme importimiseks tuleb käsurealt minna asukohta, kust avalik võti alla laaditi. Vajalikku asukohta liikumine:

cd /home/kasutajanimi/<asukoht, kuhu avalik võti salvestati>

Avaliku võtme importimine:

```
gpg --import <avalikuvõtmefail.asg>
```

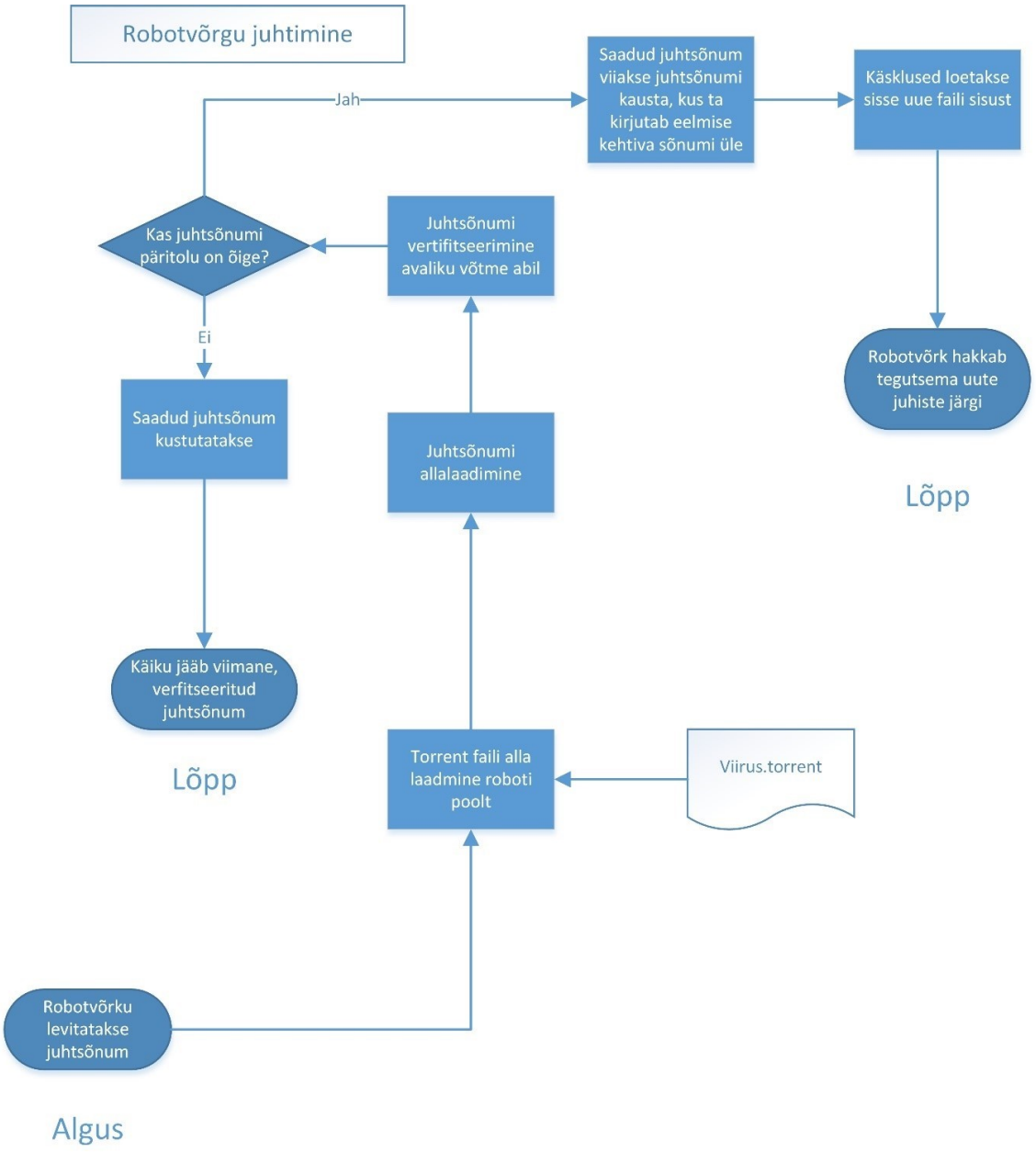
Robotvõrgu paigaldamine on visualiseeritud alljärgneval joonisel (vt Joonis 18):



Joonis 18. Robotvõrgu paigaldamine

4.2 Robotvõrgu juhtimine

Robotvõrgu juhtimise toimingute alla kuuluvad torrentfaili jagamine jälgurarvuti poolt, kliendipoolne juhtsõnumi allalaadimine, verifitseerimine ning juhtsõnumi õigesse kausta liigutamine. Rutiinsed toimingud on visualiseeritud alljärgneval joonisel (vt Joonis 19):



Joonis 19. Robotvõrgu juhtimine

4.2.1 Torrentfaili jagamine

Esimese toiminguna signeeritakse juhtsõnum. Signeerimiseks kasutatakse eraldiseisva signeerimise meetodit (*detached signature*), mis tähendab, et signeeritavast failist (juhtsõnumist) ning signatuurfailist luuakse omavahel seotud paar. Kui paari osad

salvestada erinevatesse kohtadesse, ei ole võimalik verifitseerimist teostada. Signeerimiseks tuleb minna käsureale ning sisestada käsklus:

```
gpg --output <signatuurfailinimi.sig> --detach-sig <signeeritav fail>
```

Näiteks:

```
gpg --output juhtsonum.sig --detach-sig juhtsonum.txt
```

Signeerimisel tekkinud failidepaari jagamiseks tuleb lisada nad torrentfaili. Selleks kasutatakse skripti *tracker.sh*, mida kasutati ettevalmistavas etapis (vt 4.1):

```
sudo bash tracker.sh <jagatavad failid> <loodava torrentfaili nimi>
<jälguriaadress>
```

4.2.2 Juhtsõnumi allalaadmine

Klient peab juhtsõnumi kättesaamiseks alla laadima jälguri poolt loodud torrentfaili. Kasutada võib selleks veebilehitsejat või käsurida:

```
wget --directory-prefix=<soovitud asukoht><torrenti URL>
```

Juhtsõnumi allalaadimiseks tuleb käivitada käsurealt skript *loading.sh*:

```
sudo bash loading.sh <torrent fail>
```

4.2.3 Juhtsõnumi allalaadmine

Peale juhtsõnumi allalaadimist toimub verifitseerimine. Selleks kasutatakse juhtsõnumiga kaasa tulnud signatuurfaili ning jälgurarvuti poolt tulnud avalikku võtit. Verifitseerimiseks tuleb liikuda käsureaga kausta, kuhu failide paar alla laaditi:

```
cd <asukoht, kuhu failide paar salvestati>
```

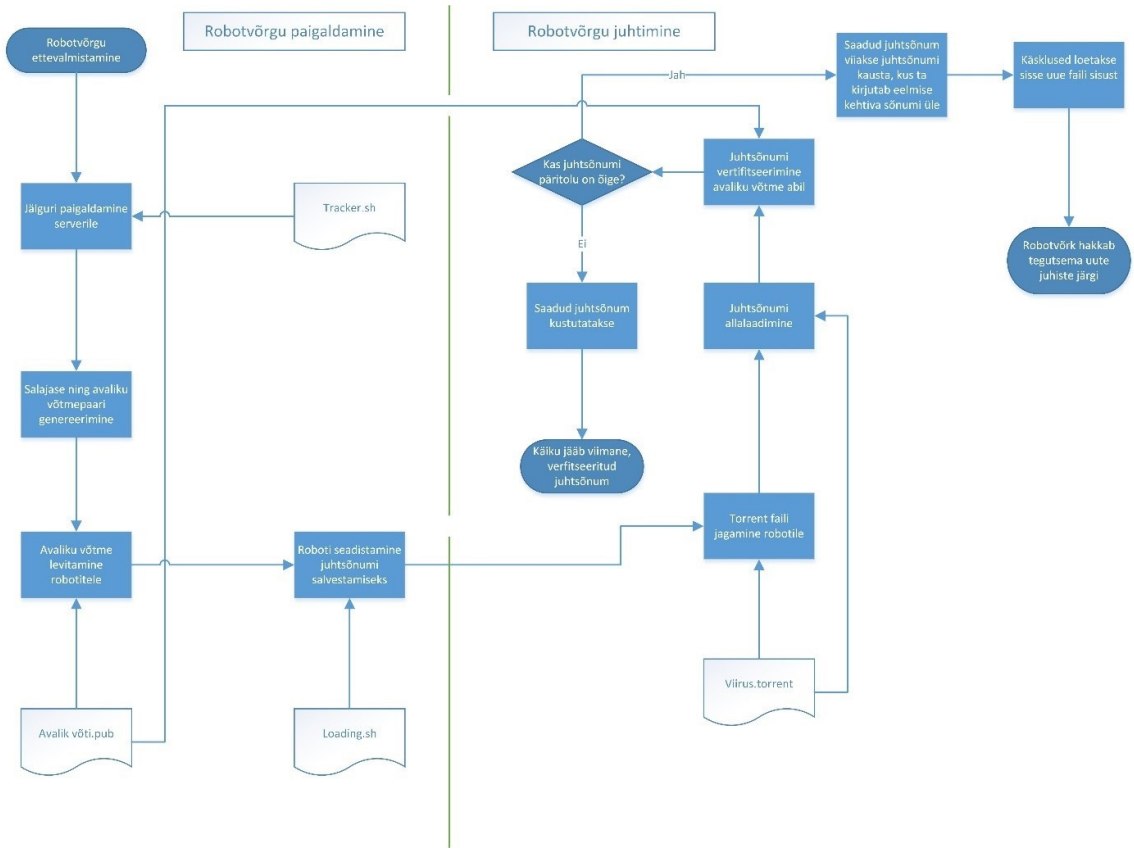
ning sisestada käsklus:

```
gpg –verify <signatuurfail.asg> <juhtsõnumi fail>
```

Juhul kui juhtsõnumi päritolu on õige, viiakse verifitseeritud juhtsõnum kausta, kus ta kirjutab üle eelmise, kehtiva juhtsõnumi. Käskluste lugemiseks loeb võrguliikluse tekitaja sama juhtsõnumi faili, milles on uued käsklused. Kui juhtsõnumi fail ei ole õiget päritolu,

siis antud juhtsõnum kustutatakse ning käiku jäävad käsklused, mis on saabunud viimase, verifitseeritud juhtsõnumiga.

Robotvõrgu paigaldamise ning juhtimise protsessid on visualiseeritud alljärgneval joonisel (vt Joonis 20):



Joonis 20. Robotvõrgu paigaldamine ning juhtimine

5 Testimine

Testimise peatükis hinnatakse loodud lahenduse vastamist esitatud nõuetele. Peatüki lõpus tehakse kokkuvõte testide ning hindamiste tulemustest ning otsustatakse lahenduse sobivuse üle töös käsitletud probleemi lahendamisel.

5.1 Nõude N1 testimine

Nõude N1 järgi (vt 3.5.1) peavad lahenduse tarkvarakomponendid olema dubleeritud. Nõudele vastamise kontrollimiseks loodi torrentfail, mida jagati jälgurarvutiga klient arvutile. Kui klient arvuti oli faili täies mahus alla laadinud, lülitati jälgurarvuti välja. Seejärel jagati sama torrent fail järgmisele arvutile ning vaadati, kas tal õnnestub faili allatõmbamine. Positiivse tulemuse korral loetakse test õnnestunuks ning nõue täidetuks.

Testimiste käigus õnnestus, kergete viperustega, faili jagamine kahe osapoole vahel ajal, kui jälgurarvuti oli välja lülitatud. Lahenduse vastamise nõudele N1 saab lugeda vaid osaliselt täidetuks, sest kuigi faili jagamine õnnestus, toimus see ainult ühe torrent faili raames. Uue sisuga faili jagamiseks on vaja arvutile paigaldada jälgurtarkvara, mille ülesandeks on teostada torrentfaili alglaadimine robotvõrku. Teine võimalus on välja lülitatud jälgurarvuti uuesti sisse lülitada. Nõude N1 testimise juures välja tulnud puudujääkide parandamine kuulub lahenduse juures teostatavate edaspidiste tööde hulka.

5.2 Nõude N2 testimine

Nõude N2 järgi (vt 3.5.2) edastatud juhtsõnum peab olema verifitseeritud. Nõudele vastamise testimiseks signeeritakse võrdvõrku paisatav põhifail jälgurarvuti poolt ning vastuvõetavates sihtkohtades kontrollitakse antud faili verifitseeritust vastavalt avalikule võtmele. Signeerimiseks kasutatakse GnuPG *detached* moodust. Antud moodus tekitab põhifailist signatuurfaili, mille kaudu saab kontrollida faili verifitseeritust. Signatuurfail ning põhifail moodustavad omavahel terviku. Paari ühe liikme puudumisel ei saa verifitseerimist teostada. Enne robotvõrgu rutiinset kasutamist saadetakse vastuvõtvale poolele signeerija avalik võti, mille abil verifitseerimine toimub.

Test loetakse õnnestunuks, kui saadetud põhifaili verifitseerimine õnnestub ning faili signeerijaks näidatakse olevat jälgurarvuti.

Läbiviidud testide tulemused näitasid, et lahenduses kasutatav verifitseerimise süsteem annab nõutud tulemuse ning on sobilik võrdvõrku paisatavate failide verifitseerimiseks.

5.3 Nõude N3 testimine

Nõude N3 (vt 3.5.3) järgi peab lahendus olema loodud tarkvarakomponentidest, mille litsents kuulub OSI (*Open Source Initiative*) alla. Lahenduses kasutatud vahenditest tehakse nimekiri ning vaadatakse, millist litsentsi need kasutavad ja kas need on OSI-ga ühilduvad.

Alljärgnevas tabelis (vt Tabel 2) on üles märgitud lahenduses kasutatud vahendid, kasutatav litsents ning see, kas litsents on OSI-ga ühilduv.

Tabel 2. OSI-ühilduvus

Tarkvara komponent	**Litsents**	**OSI-ühilduv**
GnuPG(GPG)	*GNU General Public License v3.*	Jah [48]
Transmission	*GNU General Public License v2.*	Jah [49]
Bitstorm	*GNU General Public License v3.*	Jah [49]
Rtorrent	*GNU General Public License v2.*	Jah [49]
Apache HTTP Server	*Apache License 2.0*	Jah [50]
PHP5	*PHP License v3.01*	Jah [51]

Tabelist on näha, et kasutatava tarkvarakomponentide litsentsid kuuluvad *Open Source Initiative* alla. Lähtudes tulemustest, võib lugeda nõude N3 (vt 3.5.3) täidetuks.

5.4 Nõude N4 testimine

Nõude N4 (vt 3.5.4) järgi peab lahendus toetama antud nõudes ette antud operatsioonisüsteeme. Nende alla kuuluvad Microsoft Windowsi versioonid 7, 8 ja 10, Apple OS X ning levinumad Linuxil põhinevad distributsioonid (Ubuntu LTS(*Long Term Support* (Pika toega)) versioon (14.04) [52] , Debiani viimane *stable* versioon(Debian 8) [53]). Lahenduse ja komponentide testimiseks kasutati Ubuntu versiooni 15.10, mis on viimane versioon enne aprillis 2016 ilmuvat uut Ubuntu LTS versiooni. Testimise tulemused näitasid, et lahenduses kasutatavad komponendid toetavad Ubuntu operatsioonisüsteemi. Saadud tulemuste järgi võib nõude pooleldi täidetuks lugeda, sest testimise skoopi ei kuulunud loetletud Microsoft Windowsi versioonid ning Apple OS X. Microsoft Windowsil töötava lahenduse arendamine ning rakendamine kuulub edasipidiste teostatavate tööde hulka ning valmib tulevikus korraldatavate küberõppuste ajaks.

5.5 Nõude N5 testimine

Lahenduse vastamist nõudele N5 (vt 3.5.5) kontrollivad testid teostati nõude N1 testimise käigus. Lähtudes nõude sisust, mille järgi peab sõnumivahetus toimima ilma keskserverita (antud juhul jälgurarvutita), võib lahenduse vastamise nõudele N5 lugeda täidetuks.

5.6 Nõude N6 testimine

Nõue N6 järgi (vt. 3.5.6) järgi peab robotvõrk suutma hallata kuni 1000 võrguliikluse tekitajat. Kuna töö autoril ei olnud testimiseks kasutada antud koguses masinaid, kasutati testimiseks jälgurarvutit ning kahte klientarvutit. Kirjeldatud ülesehitusega teostati kõik etapid, mida on kirjeldatud teostuse peatükis (vt Teostus) ning tulemused olid positiivsed. Analoogseid lahendusi kasutatakse ka mujal maailmas, näiteks mängu uuenduste alla laadimiseks, Linuxi distributsioonide jagamiseks ning legaalseks filmide ning muusika jagamiseks [54] . Neid asjaolusid arvesse võttes võib lahenduse vastamise nõudele N6 osaliselt täidetuks lugeda. Lahenduse testimine suurema koguse võrguliikluse tekitajatega kuulub edaspidiste tööde hulka.

5.7 Nõude N7 testimine

Nõue N7 järgi (vt 3.5.7) peab loodav lahendus peab olema integreeritav küberõppuste süsteemi 40 töötunniga. Lahenduse prototüübi ülesseadmiseks kulus autori hinnangul ca üks nädal, mis jääb soovitusliku aja piiresse. Lõplikku hinnangut lahenduse vastamisele nõudele N7 ei saa anda, sest lahenduse integreerimist ning testimist koos küberõppustel kasutatavate süsteemidega ei teostatud.

Testide tulemused on kirjas alljärgnevas tabelis (vt Tabel 3)

Tabel 3 Testide tulemused

Nõude number	Nõude täitmine	Tulemus
N1	Kohustuslik	Osaliselt täidetud
N2	Kohustuslik	Täidetud
N3	Kohustuslik	Täidetud
N4	Kohustuslik	Osaliselt täidetud
N5	Kohustuslik	Täidetud
N6	Kohustuslik	Osaliselt täidetud
N7	Soovituslik	Ei saanud hinnata

5.8 Edaspidised tööd

Tööle järgnevalt ning vastavalt testides välja tulnud puudujääkidele on enne lahenduse rakendamist küberõppustel vajalik teostada mõningad tööd:

1. nõude N1 testimisel esinenud vajadus lisada kõikidele robotvõrgu liikmetele jälgurarvuti funktsionaalsus;

2. lahenduse arendamine ning testimine, toetamaks Microsoft Windows, Apple OS X'i ning lisaks Google Android operatsioonisüsteeme;

3. lahenduse testimine nõudes N6 nõutud robotite arvuga. Käesoleva töö raames toimus testimine väikese arvu robotite vahel. Kuigi autori hinnangute kohaselt peaks süsteem töötama ka 1000 roboti puhul, on vajalik ikkagi testimine nõudes kirjutatud robotite arvuga. Testimine on võimalik järgmisel LS-il;

4. lahenduse integreerimine ning testimine koos küberõppustel kasutatavate süsteemidega;

5. uue torrentfaili päringu süsteemi arendus. Saamaks kätte uut juhtsõnumit, teevad võrgus olevad robotid automaatselt päringu, saamaks teada, kas robotvõrku on üles laetud mõni uut juhtsõnumi sisaldav torrentfail.

5.9 Testimise kokkuvõte

Testimise peatükis kirjeldati, kuidas teisiti loodud lahenduse prototüübi vastamist esitatud nõuetele. Testide raames testiti ning hinnati lahenduse vastamist esitatud nõuetele. Testide käigus kontrolliti lahenduse võimekust juhtsõnumi jagamisel keskserverist sõltumata, sõnumi verifitseerituse nõude täitmist ning kasutatavate tarkvarakomponentide litsentside ühilduvust *Open Source Inititative*'iga. Lisaks anti hinnang lahenduse sobivusele haldamaks *ca* 1000 võrguliikluse tekitajat ning lahenduse töötamisele erinevate operatsioonisüsteemidega.

Testide tulemused näitasid, et lahenduse prototüüp täidab osaliselt või täielikult kõik kohustuslikud nõuded, kuid lahendust on vaja veel arendada ning suures mahus testida, enne kui seda saab rakendada küberõppuste juures (vt Edaspidised tööd).

6 Kokkuvõte

Antud töö käigus otsiti lahendust, suurendamaks küberkaitseõppustel kasutatava võrguliikluse genereerija töökindlust. See aitab tõsta õppuste kvaliteeti ning toob õppuste keskkonna reaalsele olustikule lähemale.

Praegu kasutusel olev võrguliikluse genereerija kasutab tähestruktuurilist robotvõrku. Võrguliikluse generaatori suurimaks probleemiks on ühenduse kadumine kekserveri ning robotite vahel. See tekitab olukordi, kus käskluste kättesaamatuse tõttu koormab robotvõrk õppuste keskkonna liiklusega üle ning võrk lõpetab töötamise. Samuti muudab võrguliikluse genereerija juhtimatu käitumine õppuste hindamise ebavõrdseks.

Töö käigus uuriti ning analüüsiti erinevaid robotvõrkudega seotuid aspekte: erinevad arhitektuurid, kasutusalad, robotvõrgusisesed kommunikeerumise viisid ning vahendid. Iga arhitektuuri juures toodi välja selle tugevused ning nõrkused.

Järgnevalt uuriti erinevaid robotvõrkude poolt kastutatavaid nii tavalisi kui ka alternatiivsemaid kommunikeerumise viise. Teistest viisidest põhjalikumalt vaadeldi võrdvõrgulisel arhitektuuril põhinevaid kommunikeerumise viise, kus toodi välja erinevate viiside tugevused ning nõrkused.

Uue lahenduse loomiseks alustati kõigepealt nõuete püstitamisega. Järgnevalt valiti välja lahenduse arhitektuur, milleks sai eraldiseisva juhtsõnumi edastamise süsteemi loomine. Lahenduse arhitektuuri juures kirjeldati ka tarkvara komponente, millest lahendust luuakse.

Töö tulemusena valmis lahenduse prototüüp, mis võib peale edaspidist arendamist ning testimist võib leida rakendust küberõppuste juures.

Esmaste edasiarenduste hulgas on lahenduse muutmine automatiseerituks. Lisaks on vajalik lahenduse rakendamine ning testimine teistel nõuetes nõutud operatsioonisüsteemidel lisaks Ubuntule. Kogu lahendust on vaja testida suurema hulga robotite peal veendumaks, et töö tulemusena saadu sobib ning täidab oma eesmärki küberkaitseõppuste juures

Tööst tuli välja ka hulk robotvõrkude ning võrdvõrkudega seotud teemasid, mille edasi-uurimine ning rakendamine võib kasu tuua ka teistes valdkondades.

Summary

The aim of this thesis was to find a solution for increasing the reliability of the network traffic generator which is used in cyber defense exercises. Increased reliability will increase the quality of the exercises and brings it closer to real life situations.

Network traffic generator in use, uses architecture of the star-shaped botnet. The biggest issue with the generator is the loss of communication between central server and robot. It will create situations, where due to unavailability of orders, botnet overloads and takes down the exercise's gaming environment. Also, it will bring unfairness to the assessments.

In thesis different botnets and related aspects were studied and analyzed, including architectures, fields of use and ways of communications. During that, cons and pros of botnets architectures were brought out.

Next, different ways of communication used by botnets were studied. Including most common and more alternative ways. More profoundly, architectures of P2P-based ways were explored and studied, bringing out cons and pros.

For creating the new solution of network traffic generator, the requirements were established. Subsequently, the architecture of solution was chosen. Creating detached message sending system was most suitable architecture. Along with solution, the used software components for creating prototype were described.

As a result of the thesis, the prototype of the solution was created. After further developments and testing, it could be used in cyber-defense exercises.

Further developments include automatization of the solution. Necessary is implementing and testing of the solution on other operating systems, which were described in requirements, beside Ubuntu. The whole solution needs testing with larger amount of bots, to prove its suitability for cyber-defense exercises.

Also, several topics related with botnets and P2P turned up and further research of these can be useful, because they can be beneficial in other fields.

Kasutatud kirjandus

[1] NATO. Cyber Defence Exercise Locked Shields 2013: After Action Report 2013. https://ccdcoe.org/publications/LockedShields13_AAR.pdf, 2013. [WWW](11.10.2015)

[2] CCDCOE. Locked Shields 2014 After Action Report. https://ccdcoe.org/sites/default/files/documents/LS14_After_Action_Report_Executive_Summary.pdf ,2014. [WWW] (11.10.2015)

[3] Cyber Defence Exercises. [WWW] https://ccdcoe.org/event/cyber-defence-exercises.html (12.11.2015)

[4] Küberolümpia.[WWW] http://www.kyberolympia.ee/ (1.05.2016)

[5] Küberkaitse.[WWW] http://www.ttu.ee/sisseastujale/magistriope-2/erialad-10/infotehnoloogia-teaduskonna-erialad/kuberkaitse-2/ (06.05.2016)

[6] Cyber Security Engineering. [WWW] http://www.itcollege.ee/en/admission/ (06.05.2016)

[7] Geers, K. Strategic Cyber Security. Tallinn : CCD COE Publication, 2011.

[8] Naumanis, E. Centrally Managed Network Traffic Generation For Cyber Security Exercises. http://digi.lib.ttu.ee/i/file.php?DLID=1786&t=1 ,2014.[WWW](26.08.2015)

[9] Locked Shields 2015. [WWW] https://ccdcoe.org/locked-shields-2015.html (06.05.2016)

[10] HITSA. Küberolümpia 2015 võitis Jaanus Kääp IT Kolledžist [WWW] http://www.itcollege.ee/blog/2015/02/14/kuberolumpia-2015-voitis-jaanus-kaap-it-kolledzist/ (29.12.2015)

[11] Ernits, M, Tammekänd, J ja Maennel, O. i-tee : A fully automated Cyber Defense Competition for Students Categories and Subject Descriptors. – *Proc. ACM SIGCOMM (Poster/demo session)*. 2015, pp. 113-114. [WWW] http://conferences.sigcomm.org/sigcomm/2015/pdf/papers/p113.pdf (16.11.2015)

[12] Uses of botnets. [WWW] https://www.honeynet.org/book/export/html/52 (06.05.2016)

[13] What is a botnet? [WWW] https://www.f-secure.com/en/web/labs_global/botnets (16.09.2015)

[14] Rouse, M. botnet (zombie army) definition [WWW] http://searchsecurity.techtarget.com/definition/botnet (06.05.2016)

[15] Bano, S. A Study of Botnets: Systemization of Knowledge and Correlation-based Detection. http://www.cl.cam.ac.uk/~sk766/publications/ms_thesis_sheharbano.pdf ,2012. [WWW] (04.08.2015)

[16] Feily, M, Shahrestani, A ja Ramadass, S. A Survey of Botnet and Botnet Detection. – *Proceedings - 3rd International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2009*, pp. 268-273. IEEE ,2009.

[17] Pinto, J. Distributed & Grid Computing. [WWW] http://www.jimpinto.com/writings/grid.html (30.11.2015)

[18] Traynor, I. Russia accused of unleashing cyberwar to disable Estonia. [WWW] http://www.theguardian.com/world/2007/may/17/topstories3.russia (30.11.2015)

[19] Microsoft. Battling the Rustock Threat. https://www.microsoft.com/en-us/download/confirmation.aspx?id=26673 ,2011. [WWW] (22.09.2015)

[20] C11-0222 Microsoft Corporation v. John Does 1-11 controlling a computer cotnet thereby injuring Microsoft and its customers. http://blogs.technet.com/cfs-

file.ashx/__key/CommunityServer-Blogs-Components-WeblogFiles/00-00-00-82-95-DCU/2112.2011_2D00_02_2D00_09_2D00_Complaint.pdf, 2011. [WWW] (04.05.2016)

[21] Ollmann, G. Botnet Communication Topologies. https://www.damballa.com/downloads/r_pubs/WP_Botnet_Communications_Primer.pdf, 2011. [WWW] (29.09.2015)

[22] Gassen, J, Tiirmaa-Klaar, H, Gerhards – Padilla, E, Martini, P. Botnets: How to Fight the Ever-Growing Threat on a Technical Level. http://www.springer.com/cda/content/document/cda_downloaddocument/9781447152156-c2.pdf, 2013. [WWW](26.09.2015)

[23] DDos Definitions – DdoSPedia. [WWW] http://security.radware.com/knowledge-center/DDoSPedia/command-and-control-server/ (06.05.2016)

[24] OpenDNS. The Role of DNS in Botnet Command & Control. http://info.opendns.com/rs/opendns/images/OpenDNS_SecurityWhitepaper-DNSRoleInBotnets.pdf, 2011. [WWW] (27.09.2015)

[25] Wang, P, Sparks, S ja Zou, C.C. Zou. An Advanced Hybrid Peer-to-Peer Botnet. *IEEE Transactions on Dependable and Secure Computing*. 2010, 7 (2), pp. 113-127. [Online] IEEE (11.10.15)

[26] Roshan, M. A New Generation Peer-to-Peer Advanced Botnet. http://interscience.in/IJIC_Vol1Iss2/paper10.pdf ,2011. [WWW] (23.10.2015)

[27] Buse, J. W. TCP/IP Protocols: Internet Relay Chat (IRC). [WWW] http://www.linux.org/threads/tcp-ip-protocols-internet-relay-chat-irc.4993/ (30.11.2015)

[28] Oikarinen, J. Founding IRC. [WWW] http://www.mirc.com/jarkko.html (30.11.2015)

[29] Reed, D, Oikarinen, J. Internet Relay Chat Protocol [WWW] https://tools.ietf.org/html/rfc1459#section-1 (27.12.2015)

[30] IRC Information..... [WWW] http://www.ircbeginner.com/ircinfo/ircc-commands.html (27.12.2015)

[31] IRC Networks - Top 10 by comparison. [WWW] http://irc.netsplit.de/networks/top10.php (30.11.2015)

[32] IRC is dead, long live IRC. [WWW] http://royal.pingdom.com/2012/04/24/irc-is-dead-long-live-irc/ (30.11.2015)

[33] Beal, V. HTTP - HyperText Transfer Protocol [WWW] http://www.webopedia.com/TERM/H/HTTP.html (05.05.2016)

[34] HTTP – Overview. [WWW] http://www.tutorialspoint.com/http/http_overview.htm (14.11.2015)

[35] Rouse, M. peer-to-peer (P2P) definition [WWW] http://searchnetworking.techtarget.com/definition/peer-to-peer (04.05.2016)

[36] Park, H, Ratzin, R. I. ja van der Schaar, M. Peer-to-Peer Networks - Protocols, Cooperation and Competition. http://medianetlab.ee.ucla.edu/papers/chapter_P2P_hpark.pdf, 2011. [WWW] (01.11.2015)

[37] DHT Protocol. [WWW] http://www.bittorrent.org/beps/bep_0005.html#kademlia (01.01.2015)

[38] Tyson, J. How the Old Napster Worked. [WWW] http://computer.howstuffworks.com/napster.htm (31.12.2015)

[39] Brain, M. How Gnutella Works. [WWW] http://computer.howstuffworks.com/file-sharing3.htm (28.04.2016)

[40] The Basics of BitTorrent. [WWW] http://help.bittorrent.com/customer/en/portal/articles/178790-the-basics-of-bittorrent (01.01.2016)

[41] Singh, T. How BitTorrent becomes a DDoS Tool [Hacking]. [WWW] http://geeknizer.com/bittorrent-as-ddos-tool/ (01.01.2016)

[42] Heide, H.v.d. BitTorrent Monitoring and Statistics, https://www.os3.nl/_media/2011-2012/courses/rp2/p41_report.pdf, 2012 [WWW] (01.02.2016)

[43] Watson, S. How Kazaa Works. [WWW] http://computer.howstuffworks.com/kazaa3.htm (28.04.2015)

[44] Talamantes, J. Botnet Command and Control via Covert Channels. [WWW] http://www.redteamsecure.com/labs/post/28/Botnet-Command-and-Control-via-Covert-Channels (19.11.2015)

[45] Edwards, J. Twitter Appears To Be Fighting A Massive Botnet, Possibly Involving Millions Of Accounts. [WWW] http://www.businessinsider.com/twitter-fighting-a-massive-botnet-2014-3 (19.11.2015)

[46] Cohen, J. What is a torrent tracker? [WWW] https://www.quora.com/What-is-a-torrent-tracker/answer/Jimmy-Cohen (22.12.2015)

[47] Caprioli, P. Bitstorm: A lightweight Bittorrent tracker. [WWW] http://felten.se/announce/ui.php (22.12.2015)

[48] GNU General Public License, version 3 (GPL-3.0). [WWW] https://opensource.org/licenses/GPL-3.0 (17.12.2015)

[49] GNU General Public License, version 2 (GPL-2.0). [WWW] https://opensource.org/licenses/GPL-2.0 (17.12.2015)

[50] Apache License, Version 2.0. [WWW] https://opensource.org/licenses/Apache-2.0 (25.04.2016)

[51] The PHP License 3.0 (PHP-3.0). [WWW] https://opensource.org/licenses/PHP-3.0 (25.04.2016)

[52] Releases. [WWW] https://wiki.ubuntu.com/Releases (18.12.2015)

[53] Debian Releases. [WWW] https://www.debian.org/releases/ (18.12.2015)

[54] Hoffmann, C. 8 Legal Uses For BitTorrent: You'd Be Surprised. [WWW] http://www.makeuseof.com/tag/8-legal-uses-for-bittorrent-youd-be-surprised/ (01.01.2016)