TARTU ÜLIKOOL

MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduse instituut
Tarkvarasüsteemide õppetool
Informaatika eriala

Veiko Sang

Eesti veebi serveripoolse keskkonna uuring

Magistritöö

Juhendaja: Anne Villems

Tartu 2004

Sisukord

Sissejuhatus

Veebi muutumine ülipopulaarseks äri-, meelelahutus ja infokeskkonnaks on kahtlemata põhjustatud dünaamilise veebi arengust. Kui staatilise veebi korral koosneb veebisait kindlaksmääratud veebilehekülgedest ning veebilehele muudatuste sisse viimiseks tuleb inimesel käsitsi olemasolevaid veebifaile muuta, siis dünaamilise veebi korral toimub veebilehekülgede loomine automaatselt veebiserveris töötavate spetsiaalsete programmide abil, mis moodustavad veebilehe sisu olemasolevate tekstifailide põhjal, andmebaasis olevatest andmetest ja/või programmide enda automaatsest väljundist (nt kellaaeg). Tänasel päeval ei ole suured ja populaarsed saidid juba ammu enam staatilised. Serveripoolsete programmidega on võimalik luua interaktiivseid veebisaite (Internetipank, infootsing saidil, veebipõhine foorum, mitmesugused portaalid), mis vastavalt kasutaja sisendile genereerivad veebilehe koos vajaliku infoga. Käesolevas töös mõeldakse dünaamilise veebisaidi all just sellist saiti, mille funktsionaalsuse tõstmiseks kasutatakse serveripoolseid programme, mis on veebilehe sisu genereerimiseks reaalajas käivitatavad saidikülastajate poolt või millele kasutaja saab edastada oma sisendi läbi veebivormi, küpsiste või veebiaadressis sisalduvate parameetrite.

Seoses järjest efektiivsemate ja lihtsamalt kasutatavate serveripoolsete veebitehnoloogiate esile kerkimisega muutuvad saidid üha enam dünaamilisemateks, võimaldades maksimaalselt ära kasutada mugavust, mida veeb pakub inimestevaheliseks suhtluseks ja võimalikult värske info (uudised, hinnad, ilm) saamiseks. Kindlasti pakub paljudele Interneti kasutajatele ja veebiarendajatele, sealhulgas töö autorile, huvi kui kaugele on veeb oma arengus dünaamilisuse suunas liikunud ning missuguseid serveripoolseid tehnoloogiaid ja mis ulatuses kasutatakse? Käesoleva töö autoril ei õnnestunud kogu veebist leida ühtegi uurimust serveripoolsete veebiprogrammeerimise vahendite kasutuse kohta, mis annaks tõepärase pildi **paljude erinevate** serveripoolsete tehnoloogiate omavahelisest jaotusest ja kasutuspopulaarsusest ning arvestaks samal ajal ka veebisaitide **kogu HTML** (*Hypertext Markup Language*) **sisuga**, et täpsustada vastava tehnoloogia tegelikku kasutust saidi loomisel. Sellest tuleneb antud töö idee, mille põhieesmärk on läbi viia HTTP (*Hypertext Transfer Protocol*) päiseinfost ja saitide kogu sisust lähtuv mahukas uuring serveripoolsete tehnoloogiate populaarsuse kohta, sest siiani on vastavates uuringutes sõltumatult kasutatud kas HTTP päiseinfot või saitide esilehel sisalduvaid veebiaadresse.

Antud töö on oma olemuselt edasiarenduseks A. Sibola poolt 2000. aastal läbi viidud Eesti

veebimaastiku lihtsale uuringule [1]. Edasiarenduse eesmärgiks on paremini ja täpsemalt kajastada Eesti veebi serveripoolset keskkonda, arvestades tulemuste väljastamisel lisaks HTTP päiseinfole ja domeeninimepõhisusele ka IP-aadressipõhisusega, veebilehekülgede sisuga, serveris asuvate saitide (virtuaalserverite) arvuga ning suuremate saitidega (lehekülgede arvu järgi). Valimiks on käesolevas uurimuses võetud Eesti veebiserverid ja saidid, mis annab võimaluse võrrelda Eestit muu maailma kohta eksisteerivate (küll pealiskaudsemate) uurimuste tulemustega serveripoolsete tehnoloogiate valdkonnas.

Töö üldisem eesmärk on anda ülevaade kogu serveripoolsest keskkonnast, milles staatilised saidid ja veebiinfosüsteemid töötavad. Serveripoolse keskkonna komponentideks loetakse antud töös veebiserveri tarkvara, veebiserveri operatsioonisüsteemi, serveripoolseid programme (erinevad tehnoloogiad dünaamilise veebi loomisel) ja andmebaasiserveri tarkvara.

Antud töö materjal eeldab lugejalt mõningaid eelteadmisi klient-server süsteemide toimimisest, HTTP-protokollist, märgendikeelest HTML ja veebiprogrammeerimisest. Dünaamilise veebi olemusest ning selle loomiseks kasutatavatest põhilistest tehnoloogiatest annab ülevaate koos praktiliste näidetega töö autori poolt kirjutatud bakalaureusetöö [2].

Käesoleva uuringu tulemused on mõeldud eeskätt veebiarendajatele, kes saavad ülevaate veebiinfosüsteemide keskkondade populaarsusest praegusel hetkel avalikus veebis. Eriti sobivad töös esitatud tulemused uutele veebiarendajatele, kes alles tutvuvad neile võõra valdkonnaga ning teevad oma esimesi valikuid veebiarendusvahendite valikul. Tulemustest on näha, millist tarkvara seoses veebiga kasutatakse ja kui suures ulatuses. Tulemused kajastavad olukorda Eesti veebis 2004. aasta aprilli seisuga.

Töö koosneb kahest peatükist. Esimeses peatükis vaadeldakse erinevaid võimalusi ja aspekte serveripoolse keskkonna hindamisel ning kirjeldatakse detailselt käesoleva uurimuse metoodikat ning reaalseid puudusi. Teises peatükis esitatakse eelnevalt kirjeldatud meetodite abil saadud tulemused ning võrreldakse neid nii omavahel kui ka olemasolevate teiste analoogiliste uurimuste tulemustega. Põhilisteks tulemusteks on saitide jagunemine dünaamilisteks ja staatilisteks ning erinevate serveripoolsete tehnoloogiate – PHP, *JavaServer Pages* (JSP), *Active Server Pages* (ASP), *Internet Server Application Programming Interface* (ISAPI), *Common Gateway Interface* (CGI) jpt – kasutuspopulaarsus veebi loomisel. Lisaks on serveripoolse keskkonna määratlemisel esitatud tulemused andmebaasiserverite (Oracle,

MS SQL Server, MySQL, PostgreSQL jt), veebiserveri tarkvara (Microsoft, Apache jt) ja operatsioonisüsteemi tüüpide (Unix, Windows jt) populaarsuse kohta.

Käesoleva töö lisas (CD) on toodud uuringu käigus kasutatud ja loodud andmefailid ning andmete kogumiseks, töötlemiseks ja analüüsimiseks koostatud käsureaskriptid keeles PHP. Kõik failid, millele antud töö tekstis viidatakse, asuvad töö lisas.

1. Serveripoolse keskkonna hindamise metoodikad

Selles peatükis antakse ülevaade serveripoolse keskkonna automaatseks uurimiseks töös kasutatavatest metoodikatest, viidatakse olemasolevatele tulemustele, tuuakse välja kirjeldatud meetodite võimalikud puudused ning esitatakse autoripoolsed täiendused, mis võimaldavad tegelikku olukorda paremini kajastada. Samuti kirjeldatakse detailselt käesoleva uuringu iseärasusi ja metoodikat. Kas esitatud autoripoolsetel täiendustel on tulemustele ka mingi mõju, selgub teises peatükis, kus on toodud Eesti veebisaitidel põhinenud uuringute tulemused, mille saamiseks on kasutatud samal valimil põhinevaid mitmeid erinevaid metoodikaid.

1.1. HTTP-protokollil põhinev metoodika

Kõige parem ja informatiivsem vahend veebimaastiku automaatsete uuringute teostamiseks on veebiprotokoll HTTP, mille alusel on klientidel võimalik veebiserveritega suhelda ning küsida vajalikku infot (veebilehekülge). Vastavalt HTTP spetsifikatsioonile [3] koosneb kliendi päring ja serveri vastus kahest osast: päisest ja kehast. Päis koosneb erilisest esimesest reast ning sellele järgnevatest päiseridadest, milles asuvale infole antakse tähendus päisevälja nimetusega. Esimene päiserida peab HTTP teates kindlasti eksisteerima, sest klient esitab selle kaudu oma soovi (tavaliselt veebilehekülje päring) serverile ning server teatab päise esimese rea kaudu nõutud ressursi staatuse (päring täideti edukalt, tekkis viga, vaadata mujalt aadressilt jpt).

Olgu näiteks huvipakkuvaks saidiks www.linnakivi.ee, siis selle saidi juurkataloogi (/) HTTP päiseinfo (käsk `HEAD`) saamiseks võib vastava serveri pordile number 80 saata lihtsa HTTP päringu, mis on toodud joonisel 1.

```
HEAD / HTTP/1.0
Host: www.linnakivi.ee
```

Joonis 1. HTTP päring veebilehe päiseinfo saamiseks.

Päiseväljaga `Host` määratakse konkreetse saidi nimi. Klient võib oma HTTP päringusse lisada veel paljusid teisi päiseridasid, et informeerida serverit vastuse saatmisel. Näiteks väljal `User-Agent` edastatakse klienti identifitseeriv info (Mozilla, Internet Explorer), väljal

`Accept` kirjeldatakse dokumenditüübid, mida klient oskab interpreteerida (tekstifail, pildifail, PDF-fail), väljal `Accept-Language` loetletakse keeled (eesti, inglise), milles võiks eelistatult tagastatav dokument olla kirjutatud.

Serveri HTTP vastus joonisel 1 toodud päringule on esitatud alljärgnevalt joonisel 2. Serveri vastuse esimene rida (päiserida) sisaldab kolmekohalist HTTP olekukoodi, mis kirjeldab kliendi poolt nõutud ressursi staatust: koodid kujul 2xx näitavad faili olemasolu ja vastuse saatmise õnnestumist, koodid 3xx teavitavad klienti lehekülje asumisest teisel aadressil, mis antakse kaasa tavaliselt päiseväljaga `Location`, koodid 4xx väljendavad veaolukorda, kui nõutud faili ei eksisteeri (404, 410), faili näitamine on keelatud (403) või failile ligipääsemiseks on vaja kasutaja autentimist (401), koodid 5xx kirjeldavad serveri sisemisi veaolukordi, kui süsteem pole tehniliselt võimeline vastama või on vigaselt konfigureeritud. Olekureale järgneb hulgaliselt päiseridu, mis on seotud vastava dokumendi aegumisega (`Expires`, `Cache-Control`, `Pragma`, `Last-Modified`), dokumendi sisu kirjeldusega (`Content-Type`, `Content-Length`), veebiserveri ja selle lisafunktsionaalsuse identifitseerimisega (`Server`), töötava veebirakendusega (`Set-Cookie`, `X-Powered-By`) jpm.

```
HTTP/1.1 200 OK
Date: Thu, 25 Mar 2004 08:40:59 GMT
Server: Apache/1.3.29 (Unix) mod_ssl/2.8.16 OpenSSL/0.9.6m PHP/4.3.4
X-Powered-By: PHP/4.3.4
Set-Cookie: sess_admin=09f3b0d504385e26e15494d5bb9f584e; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Connection: close
Content-Type: text/html
```

Joonis 2. Veebiserveri HTTP vastus päiseinfo päringule.

Veebiserverid võivad väljastada HTTP päiseridadel veebipõhiste uuringute jaoks meeldivalt palju infot. Näiteks HTTP vastuse päisest piisab enamasti selleks, et teada saada valitud veebisaidile vastavat veebiserverit, sellele veebiserverile installeeritud lisafunktsionaalsust (s.h. serveripoolseid skriptivahendeid), veebiserveri operatsioonisüsteemi ning saidil kasutatavaid küpsiseid. Ülaltoodud päringutulemusest saab välja lugeda, et kasutatav veebiserver on Apache, mis töötab Unix-laadsel operatsioonisüsteemil, veebiserverile on

lisatud HTTPS (*HyperText Transfer Protocol Secured*) protokolli toetus ning PHP interpretaator. Kõik see info asub väljal `Server`. Lisaks on näha, et kasutuses on mingi serveripoolne programm (oletatavalt PHP rakendus), mis kasutab oma töös sessiooniküpsist nimega `sess_admin` (väljalt `Set-Cookie`). Peale HTTP standardsete päiseväljade (`Date`, `Server`, `Expires`, `Cache-Control`, `Content-Type` jpt) võib serverisse installeeritud tarkvara lisada veebiserveri väljundisse mistahes enda spetsiifilisi päiseridu. Näiteks reklaamib PHP interpretaator (ka ASP) ennast väljal `X-Powered-By`. Osa sisuhaldustarkvara kasutab enda reklaamimiseks päisevälju `X-Content-Parsed-By`, `Generator` jt, kust on võimalik sisuhaldustarkvara spetsiifiliselt välja lugeda kasutatav serveripoolne skriptivahend.

Konkreetse veebilehekülje HTML-sisu saamiseks on kasutatav HTTP käsk `GET`. Näiteks saidi www.rajaleidja.ee juurkataloogile vastava HTML-lehekülje saab klient kätte, kui serveri pordile numbriga 80 edastada joonisel 3 toodud päring.

```
GET / HTTP/1.0
Host: www.rajaleidja.ee
```

Joonis 3. HTTP päring saidi esilehe saamiseks.

Vastuseks sellele päringule annab veebiserver nii HTTP päise kui ka HTTP kehas paikneva HTML lähteteksti (Joonis 4), mille põhjal saab teha märgendipõhiseid uuringuid (freimid, kliendipoolsed skriptid, Java rakendid) kui ka sisu indekseerimist ja veebiaadressidest uute saitide avastamist. Serveripoolsete tehnoloogiate määramise seisukohast on HTML-sisust kõige olulisemad saidisisesed veebiviited, mis paljastavad serveripoolsete programmide olemasolu ning üldjuhul ka nende liigi, nagu on näha joonisel 4 toodud veebiserveri väljundist, kus freimide sisu moodustatakse PHP-failidega:

```
HTTP/1.1 200 OK
Date: Thu, 25 Mar 2004 09:46:17 GMT
Server: Apache/1.3.27 (Unix)
Cache-Control: max-age=1
Expires: Thu, 25 Mar 2004 09:46:18 GMT
Last-Modified: Wed, 03 Dec 2003 15:06:48 GMT
ETag: "40829-3a1-3fcdfc08"
Accept-Ranges: bytes
Content-Length: 929
Connection: close
Content-Type: text/html

<html>
<frameset rows="91,*">
  <frame SRC="frametop.php" NAME="top">
  <frameset cols="196,*">
    <frame SRC="frameleft.php" NAME="menu">
    <frameset rows="27,*">
      <frame SRC="menusub.php" NAME="submenu">
      <frame SRC="sisu.html" NAME="cont" SCROLLING="auto">
    </frameset>
  </frameset>
</frameset>
</html>
```

Joonis 4. Veebiserveri HTTP vastus veebilehekülje päringule.

Nagu ülaltoodud veebiserverite väljunditest (joonised 2 ja 4) on näha, võivad nii HTTP vastuse päis kui ka saitide sisu sisaldada päris palju infot serveripoolsete tehnoloogiate (antud näidete puhul ainult PHP) kohta. Seega on HTTP päiseinfo ja saitide sisu üheks allikaks, mille alusel saab programselt hinnata serveripoolsete tehnoloogiate populaarsust. Veebiserveri tarkvara ja operatsioonisüsteemide populaarsuse hindamisel saab samuti kasutada HTTP vastuse päiseinfot.

Negatiivse poole pealt peab HTTP-põhise meetodi juures kahjuks märkima, et serverite administraatorid võivad lisada, kustutada ja muuta mistahes HTTP päisevälju. Samuti võib mistahes faililaiendiga või kataloogiga siduda suvalise serveripoolse programmeerimiskeele interpretaatori.

Serverite operatsioonisüsteemi määramiseks on välja töötatud ka palju keerulisemaid

meetodeid. Lisaks rakenduskihi protokollide (HTTP, FTP, Telnet) põhjal saadavale infole on võrguarvuti operatsioonisüsteemi võimalik määrata ka madalama taseme protokollide (TCP) abil, mida võimaldavad teostada mitmed eriprogrammid: Nmap, SIRC, Queso jt [4, vt link „OS detection“]. Käesolevas uuringus ei kasutatud operatsioonisüsteemi tuvastamisel Nmap-taolist programmi, kuna pole garanteeritud, et tuvastatud operatsioonisüsteem on seotud veebiserveriga, vaid võib olla mistahes teise vahemasina (nt ruuteri) operatsioonisüsteem. Paljud sisevõrgud on kaitstud Nmap-taoliste võrguskaneerimisprogrammide vastu ning seetõttu võib väljastatud tulemus veebiserveri operatsioonisüsteemi kohta olla vale või suures ulatuses ebatäpne. Näiteks on juhtumeid, kus Nmap pakub ühele IP-aadressile vastavaks operatsioonisüsteemiks samal ajal nii Linuxit kui Windowsi.

1.2. Veebiotsingu saitidel põhinev serveripoolsete programmide kasutuse määramine

Lisaks eelnevalt vaadeldud HTTP-protokolli kasutamisele saab serveripoolsete tehnoloogiate populaarsust väga lihtsal meetodil [5] mõõta, kasutades veebiotsingu saite. Veebirobotid indekseerivad Internetilehekülgi ning igale leheküljele vastab Internetiaadress, mis salvestatakse otsingumootori andmebaasi. Populaarsemad veebiotsingu saidid (Google, Alta Vista, Northern Light, Neti jt) võimaldavad kasutajatel teostada otsingut terves Internetiaadressis (mitte ainult faililaiendiosas) sisalduvate sõnade alusel ning väljastavad muuhulgas ka leitud lehekülgede koguarvu. Selline tehnika annab võimaluse kokku lugeda veebilehekülgi, mis on seotud otsitava serveripoolse tehnoloogiaga. Näiteks soovides teada saada lehekülgede arvu, mis on tõenäoliselt loodud JSP tehnoloogiat kasutades, otsime Internetiaadressidest sõna „jsp“, mis on JSP tehnoloogiat kasutavate failide tüüpiline laiend. Analoogilises seoses on laiendid php, phtml, php3 PHP-ga ja asp, aspx ASP-ga. Võrreldes otsingumootori poolt tagastatud lehekülgede arvu vaatluse all olnud tehnoloogiate korral, saamegi nendevahelise populaarsuse jaotuse. Positiivne asjaolu siinjuures on see, et otsingumootor ei otsi etteantud sõna kui alamsõne Internetiaadressides, vaid kui (erisümbolitega `&?=,-./` jms) eraldatud sõna, seega näiteks otsingusõnale „asp“ vastavad URLid üldjuhul ei sisalda otsingusõnale „aspx“ vastavaid URLe. Lisaks HTTP-protokolli abil saadud serveripoolsete tehnoloogiate saidipõhistele populaarsuse tulemustele, esitatakse käesolevas uuringus võrdluseks ka veebiotsingu saitide põhjal saadud tulemused, mis näitavad populaarsust veebilehekülgede arvu järgi.

Veebiotsingu saitide kasutamine annab ilmselgelt ebatäpseid tulemusi, kuna otsitav sõna (meie mõttes faililaiend) võib paljudel juhtudel sisalduda ka URLide teistes komponentides –

serveri nimes, kataloogide ja failide nimedes, päringusõnes (*query string*) – ning üldse mitte tähendada vastavat serveripoolset tehnoloogiat (*asp* tähendab inglise keeles haavapuud, aspisrästikut, asparagiinhapet, kui ka akronüümi terminist *Application Service Provider*). Paljud dünaamilised saidid kasutavad faililaiendit html, millega loodud leheküljed jäävad antud juhul täielikult vaatluse alt välja. Oluline puudus otsingumootoripõhise tulemuse puhul on ka see, et otsingus osalevad ainult need veebilehed, mis on veebiroboti poolt indekseeritud. Siia alla ei kuulu leheküljed, mis nõuavad veebivormide täitmist ning kliendipoolsete tehnoloogiate (JavaScript, Shockwave jt) tundmist. Samuti ei arvesta veebirobotid kõikide parameetreid sisaldavate URLidega — pikkade päringusõnedega ja paljude parameetritega URLe ignoreeritakse [6]. Firma BrightPlanet andmetel [7] on nähtamatu (indekseerimata) veebi suurus koguni 500 korda suurem, kui veebiroboti poolt indekseeritud veeb.

1.3. Serveri portidel põhinev andmebaasi tarkvara tuvastamine

Dünaamiliste saitide loomise protsessis kasutatavate andmebaasiserverite kohta ei ole avalikke automaatseid uurimusi töö autorile teada. Taoliste uurimuste vähesuse peamine põhjus seisneb selles, et andmebaasiserverite kasutust või mitte kasutust ning andmebaasisüsteemi tarkvara on väljastpoolt serverit vaatlevatel võõrastel praktiliselt võimatu teada saada. Kasutades ära asjaolu, et üldjuhul installeeritakse tarkvara vaikimisi parameetreid kasutades, siis avaneb üks võimalus mingisugusegi ettekujutuse saamiseks andmebaaside kasutatavuse valdkonnas. Nimelt osutub siin kasulikuks vaikeparameetriks pordinumber, millele konkreetne andmebaasiserver klientide päringuid ootab.

Arvuti port saab olla avatud või suletud olekus. Kui port on avatud, siis kuulab seal mingisugune server/teenus, millega on kliendil võimalik suhtlemiseks luua ühendus. Kui port on suletud, siis seal mingit teenust installeeritud pole ning suhtlusühendust pole võimalik luua. Portide avatust/suletust saab kontrollida mitmete programmidega (Telnet, Nmap), mis üritavad luua ühendust etteantud pordil potentsiaalselt eksisteeriva serveriga. Näiteks kui pordiga 3306 õnnestub luua ühendus, siis tõenäoliselt on tegemist MySQL andmebaasiserveriga, kuna MySQL serveri vaikimisi installatsioonil seatav port on 3306.

Kirjeldatud meetod on väga lihtne, kuid annab paraku ebatäpseid tulemusi, kuna turvalisuse kaalutlustel ei panda missioonikriitiliste infosüsteemidega seotud andmebaasiservereid üldjuhul vaikimisi määratud pordile ega veebiserveriga samasse masinasse, samuti ei pruugi andmebaasile olla võimaldatud väljast juurdepääsu ning kaitseks võib ees olla tulemüür, mis

blokeerib kahtlased või võõrad ühenduskatsed. Loomulikult võib olla mistahes (mitte andmebaasi tarkvaraga seotud) teenus seatud kliente teenindama suvalisele pordile (kaasa arvatud tuntud andmebaasiserverite pordile). Seega andmebaasiserverite portide skaneerimine avastab eelkõige selliseid andmebaasisüsteeme, mis on seotud väiksemate ja vähem turvakriitiliste veebiinfosüsteemide projektidega.

Lisaks üldistele portide skaneerimisprogrammidele on olemas ka konkreetsest andmebaasist lähtuvad programmid, mis lisaks pordi avatusele kontrollivad, kas vastav teenus avatud pordil on tegelikult andmebaasiserveriga seotud, saates pordile andmebaasi tarkvara spetsiifilisi päringuid. Näiteks Oracle andmebaasisüsteemi eksisteerimist saab kontrollida TNS (*Transparent Network Substrate*) protokolli alusel [8].

1.4. Täiendused teadaolevatele HTTP-põhistele uurimismeetoditele

Siin jaotises käsitletakse autorile teadaolevaid HTTP-põhised tulemusi veebi serveripoolse keskkonna hindamisel. Veebimaastiku automaatsele kaardistamisele orienteeritud firmadest on tuntumad Netcraft ja E-Soft. Mõlemad avaldavad igakuiselt HTTP päiseinfo põhiseid tulemusi veebiserveri tarkvara kasutatavuse kohta veebisaitide serveerimisel [9; 10], esitades tulemused domeeninimepõhiselt. Nendest tulemustest saab välja lugeda, millist veebiserveri tarkvara üldse kasutatakse ning kui paljud saidid sellel tarkvaral töötavad. Lisaks avaldab E-Soft igakuiselt tulemusi mitmete kliendipoolsete tehnoloogiate (CSS, JavaScript, Java, Flash), küpsiste ning Apache'le installeeritud moodulite populaarsuse kohta.

Serveripoolsete programmide jaotuse kohta on avaldatud vaid üksikuid tulemusi. Kõige rohkem tulemusi serveripoolsete programmide turuosa jaotuse kohta on autorile teadaolevalt avaldanud Netcraft, kuid siiani on need olnud väga spetsiifilised: 1) ainult JSP kohta, lähtudes IP-aadressidest [11; 12]; 2) ainult Apache veebiserverite HTTP päiseinfot domeeninimepõhiselt arvesse võttev tulemus (PHP, Perl) [13]; 3) HTTP päiseinfo põhine PHP ja ASP vaheline võrdlus, lähtudes domeeninimedest [14]; 4) ainult Windows operatsioonisüsteemiga veebiserveritel põhinev uuringutulemus PHP, CFML (*ColdFusion Markup Language*) ja JSP kohta [15]; 5) valitud tehnoloogiad (ASP, JSP, CFML, Lotus Notes) saidi esilehel sisalduvate URLide põhjal, grupeerides IP-aadresside järgi [16].

Paraku ei ole käesoleva töö autoril õnnestunud leida ühtegi sellist uuringut serveripoolsete programmide kasutuse kohta, mis arvestaks samal ajal **kõikvõimalike** programmide

tüüpidega (s.h. CGI programmid) ning kasutaks andmeid samal ajal nii HTTP päisest kui ka saitide **kogu** sisust. Saitide sisuga arvestamine on oluline serveripoolsete programmide **tegeliku** kasutuse määramisel veebilehekülgede genereerimisprotsessis. Kui HTTP päiste põhjal veebiserver toetab PHP-d, siis see veel ei tähenda, et PHP-d saidi juures kasutatakse. Ainult HTTP päiseid arvestav tulemus serveripoolsete skriptide populaarsuse hindamisel näitab seda, mis vahendeid on saitidel põhimõtteliselt võimalik kasutada. Kahjuks ei tule nendest tulemustest välja, kas ja milliseid vahendeid üldse kasutatakse. Näiteks kui mingile serverile on installeeritud Apache veebiserver, mis HTTP päises avaldab infot PHP ja Perli interpretaatorite olemasolu kohta, siis ainult HTTP päiseinfo põhise uurimuse kohaselt lähevad kõik saidid, mis sellel Apache veebiserveril töötavad, arvesse nii PHP kui ka Perli populaarsuse hindamisel, kuigi ükski sait ei pruugi kumbagi vahendit kasutada. Vaatamata mingi serveripoolse tehnoloogia olemasolule või sisseehitatud toetusele serveris, ei tähenda see veel seda, et saidi loomisel vastavaid serveripoolseid programme üldse kasutatakse. Serveripoolsete programmide tegelik kasutamine avaldub saitide sisust veebivormide, tuntud faililaienditega või parameetritega URLide olemasolust HTML lähtetekstis.

Lisaks on saitide sisu põhjal võimalik avastada serveripoolseid programmeerimisvahendeid, mille kasutamine HTTP päises ei avaldu (vt jaotises 1.1. joonisel 4 esitatud veebiserveri väljundit PHP avaldumise kohta ainult HTML-sisust).

Saitide sisu puhul ainult esilehega arvestamine annab oluliselt vähem informatsiooni saitide dünaamilisuse kohta kui kogu sisuga arvestamine, kuna laialdaselt kasutatakse saidi juurkataloogile esitatud päringute ümbersuunamisi teisele asukohale ning seega esilehelt on üldjuhul leitav vaid üks veebiaadress (HTTP päiseväljalt `Location`). Käesolev uuring näitas, et 30% dünaamilistest saitidest ei paljastanud kasutatavat serveripoolset tehnoloogiat esilehel sisalduvates veebiaadressides.

Seetõttu on käesolev uurimus serveripoolsete programmide populaarsuse hindamisel keskendunud kogu saidi sisu analüüsimisele (kombineerides tulemusi HTTP päiseinfoga), võimaldades täpsemaid tulemusi võrreldes ainult esilehe sisu või ainult HTTP päiseridade arvestamisega.

Eesti veebimaastikul veebiserveri tarkvara, serverite operatsioonisüsteemide ning serveripoolsete skriptide populaarsuse hindamiseks A. Sibola poolt tehtud uuringu tulemused [1] arvestasid ainult HTTP päiseinfoga. Kuna need tulemused olid ka domeeninimepõhised

(mitte IP-aadressipõhised), siis tekib veebiserveri tarkvara tulemuste suhtes (samamoodi ka operatsioonisüsteemide korral) kahtlus, kui hästi need tulemused ikkagi kajastavad tegelikku turuosa jaotumist veebiserverite tootjate (Apache, Microsoft, Sun jt) vahel. Domeeninime- ja IP-aadressipõhise analüüsi erinev olemus seisneb selles, et ühe IP-aadressiga (füüsilise serveriga) võib siduda piiramatu hulga domeeninimesid (virtuaalservereid, saite). Oletame, et valimis on kaks serverit, millest ühele on installeeritud Apache veebiserver, millel töötab 200 väikest saiti, ja teisele on installeeritud Microsofti veebiserver, millel töötab üks suur sait, siis domeeninimepõhine tulemus näitab olematut turuosa Microsoftile ning praktiliselt 100%-list turuosa Apache'le. Samas näitaks IP-aadressipõhine tulemus võrdset jaotust kahe konkurendi vahel. Seega on IP-aadressipõhine tulemus parem kriteerium operatsioonisüsteemide ja veebiserveri tarkvara turuosa määramisel, kuna arvestab füüsilisi servereid ning ei loe sama installatsiooni mitu korda nagu domeeninimepõhine tulemus.

Domeeninimepõhine tulemus ei pruugi väga hästi kajastada tegelikku olukorda [17], kuna tulemusi dikteerivad veebiserveriteenuse pakkujad, mille korral ühel füüsilisel serveril võib paikneda tuhandeid virtuaalservereid. Seega loetakse ühte füüsilist tarkvarainstallatsiooni nii mitu korda, kui palju asub veebiserveril erinevaid saite. Seetõttu lisatakse domeeninimepõhistele tulemustele käesoleva uuringu käigus ka vastavad IP-aadressipõhised tulemused.

Et anda parem ülevaade kogu valimit (serverid, saidid) hõlmavate tulemuste kujunemisest, esitatakse uuringu tulemused ka järgmiste eritunnustega valimite põhjal: suurimad serverid (saitide arvu järgi) ehk hostinguserverid, suurimad saidid (lehekülgede arvu järgi) ja erisaidid (üks sait tervel serveril). Suurimate serverite arvestuse kaudu avalduvad veebiserveriteenuse pakkujate eelistused hostingukeskkonna loomisel. Erisaitidega arvestamise puhul kaovad valimist aga veebiserveriteenuse pakkujate serverid, mis reeglina mõjutavad väga oluliselt domeeninimepõhiseid tulemusi, kuna suur hulk saite töötab identses keskkonnas, kus saitide loomisel puudub serveripoolsete tehnoloogiate valikuvabadus. Erisaitide grupp on eriline veel selles mõttes, et sisaldab ainult neid saite, mille teenindamiseks on mõeldud kogu server. Siia alla kuuluvad üldjuhul saidid, mis on valminud eriprojektide tulemusel, mis peavad toime tulema suure koormusega ning mis nõuavad serverilt palju ressursse. Tulemused, mis põhinevad ainult suurematel saitidel, võivad oluliselt erineda kõikide saitide põhjal saadud tulemustest, kuna kõrvaldavad vaatluse alt tühjad saidid, mõneleheküljelised saidid, veebiserverite poolt vaikelehekülgedega täidetud saidid ja muud olematu tähtsusega saidid. Viimast oletust tulemuste erinevuses toetab teatud määral USA tuhande kõige edukama firma

veebisaitide analüüs [18], mis näitab veebiserveri tarkvara turuosas (Microsoft 54%, Apache 20%) vastupidist olukorda võrreldes kõiki saite arvestava Netcrafti tulemusega (Apache 67%, Microsoft 21%) [9].

1.5. Käesoleva uuringu metoodika

Läbiviidud uurimus on täisautomaatne, st uuritavate objektide kohta info saamisel pole vaja inimese osavõttu. Mitteautomaatse (telefoni kõne, küsitlus paberil, meilivorm) ja poolautomaatse (veebivorm) uurimusvormi kasuks ei otsustatud, kuna uuringut on vaja teostada mitmeid kordi, tulemusi on vaja saada kiirelt ja võimalikult paljudest kohtadest (kõik veebisaidid Eestis), inimeste kontaktandmete leidmine pole automatiseeritav, vastaja ei pruugi olla pädev, inimestel on vähe aega võõraste asjadega tegelemiseks ja nad ei pruugi üldse vastata.

Käesolev uuring Eesti veebi serveripoolse keskkonna kohta põhineb HTTP-protokollil. Uuringu põhieesmärk on hinnata serveripoolsete tehnoloogiate populaarsust, kasutades samal ajal erinevaid tulemuse esitamise meetodeid: ainult HTTP päis, HTTP päis + HTML-sisu, domeeninimepõhisus, IP-aadressipõhisus, veebiotsingu saidid, eritunnustega valimid. Vaatluse alla võetakse kõik dünaamilise veebi loomiseks kasutatavad erinevad tehnoloogiate kategooriad [1]: CGI programmid, veebiserverite API (*Application Programming Interface*) rakendused ja serveripoolsed skriptivahendid.

Et vähendada teadmatust Eesti serveripoolsete tehnoloogiate kasutuse kohta, siis on uuringu valimiks võetud Eesti veebiserverid. Samuti annab see hea võimaluse täheldada suundumusi Eesti veebis, võrreldes praegusi tulemusi A. Sibola nelja aasta taguste tulemustega [1]. Lisaks on Eesti piisavalt väike, et võtta arvesse suurt enamust Eesti serveritest ning tänu Neti veebiotsingu saidile, kust on võimalik leida Eesti veebiserverite ja saitide nimekiri [19], on lihtsalt lahendatav ka valimi probleem.

1.5.1. Valim

Uurimusse võetud veebisaidid ei ole suvalised ja tundmatud, vaid on sellised, mida võib pidada piisavalt tuntuteks, et vähemalt Neti otsingumootori andmebaasis on need olemas. Uurimuse jaoks läks lõplikult arvesse 62% nendest Eesti veebisaitidest, mis on loetletud Neti veebiotsingu saidil [19]. Järgnevalt on käesolevas jaotises kirjeldatud täpsemalt, millised saidid ja veebiserverid jäid uuringutuelmustes arvestamata.

Neti Eesti veebisaitide nimekirjast [19] on välja jäetud saidid, mille nimi ei alga sõnega „www.“. Peamine põhjus, miks on arvestatud ainult sõnega „www.“ algavaid saite, on duplikaatide eemaldamine valimist. Väga paljudel juhtudel on identsed need saidid, mille nimedest üks algab sõnega „www.“ aga teine ei alga ehk www.host.com=host.com. Tihti on saidiga www.host.com samasugust sisu ja ülesehitust omavad ka järgmised saidid: w.host.com, ww.host.com, wwww.host.com, web.host.com, w3.host.com, www2.host.com, portal.host.com, home.host.com jms. Saidid, mille nimi ei alga sõnega „www.“, ei pruugi olla ka avaliku veebi osa (intranet.host.com, mail.host.com, webmail.host.com jt) või ei olegi üldse mõeldud tavapäraseks veebisaidiks (ns.host.com, cache.host.com, wap.host.com, ftp.host.com jt). Duplikaatide eemaldamise eesmärgil on valimist välja jäetud ka mõned www-ga algavad saidid, kui nad asuvad samal IP-aadressil ning nende nimed erinevad vaid esimese taseme domeeni poolest, st saidiga www.host.ee on loetud samaks ka saidid www.host.com, www.host.net, www.host.lt jne.

Paraku jääb ülalkirjeldatud domeeninimepõhisest analüüsist väheseks, et valimist täielikult eemaldada samasisulised saidid, sest on olemas veel mitmeid veebisaite (nt www.cut.ee=www.kliinikum.ee), mille nimed on küll erinevad, kuid omavad põhimõtteliselt sama sisu. Neid saite ei ole uuringu valimist eemaldatud, kuna neid on piisavalt vähe ning nende kindlaks tegemine ei tagaks 100%-list õnnestumist, sest samade dünaamiliste saitide sisu võib erineval ajahetkel olla erinev.

Neti veebisaitide nimekirjas teeb ülalkirjeldatud puhastustööd PHP skript wwwServerid.php, mis saab ette HTML-faili netiWWWserverid.html ning tekitab faili serveriteIP.txt, mis antakse edasises ette käesoleva uuringu veebirobotile. Nimetatud failid asuvad käesoleva töö lisas.

Käesoleva uuringu tulemusi esitatakse ka tuhande Eesti suurima saidi alusel. Suurimate saitide hulka loetakse antud juhul need saidid, mis Neti veebiroboti poolt indekseerituna omavad kõige enam lehekülgi. Vastav pingerida (töö lisas olev fail netiWWWserverid_by_size.txt) on leitav Neti veebiotsingu saidilt [19], sorteerides saite mahu järgi. Kahjuks ei kuulu Neti suurimate saitide hulka absoluutselt kõik suurimad saidid, vaid ainult need, mis on veebirobotisõbralikud, st saidisisesel navigeerimisel on kasutatud vähe või mitte üldse JavaScripti või Flashi. Suurimate saitide arvestus on eriline ka selle poolest, et enamus kõige külastatavamaid saite [20] kuulub samuti veebirobotisõbraliku tuhande suurima saidi hulka.

Erisaitideks loetakse antud juhul neid saite, mis käesolevas uuringus arvesse läinud saitide hulgas (serveriteIP.txt) asuvad vastaval IP-aadressil ainukesena. Suurimate serverite hulka loetakse antud uurimuses need IP-aadressid, millele Neti veebiserverite info (serveriteIP.txt) põhjal vastab saitide arv, mis on piisav saja suurima serveri hulka jõudmiseks.

Käesolevas uuringus ei läinud arvesse veebiserverid, mis veebiroboti tööajal olid maas, ning veebisaidid, mida enam ei eksisteerinud, mis suunasid kasutaja esilehelt mingile teisele saidile (duplikaatide vähendamise võimalus), mis ei reageerinud veebiroboti päringutele, mis andsid vastuseks HTTP veateate (olekukoodidega 403, 404, 410, 503), mis küsisid saidi esilehel kasutajanime/parooli (HTTP põhiselt) või mis olid väikese sisuga (alla 200 sümboli). Väikese sisuga saitide välja jätmine oli põhjustatud massilisest olematute saitide arvust (ca 2000), mille korral saiti tegelikult ei eksisteerinud, kuid veebiserver andis vastuseks ikkagi normaalsena näiva lehekülje (olekukood 200), mis sisaldas lihtsalt infot, et saiti ei eksisteeri. Sellega seoses jäi uuringust välja ka enamus alles valmimisjärgus olevatest saitidest, mis uuringu hetkel sisaldasid vaid lühidat teadet saidi peagi valmimise kohta (*under construction*).

1.5.2. Käesoleva uuringu veebirobot

Käesoleva uurimuse jaoks on hangitud ainult sellist informatsiooni, mis on kõikidele Interneti kasutajatele avalikult igal hetkel kättesaadav. Andmed on kogutud HTTP-protokolli abil. Valimiks olevad saidid asuvad failis serveriteIP.txt, kus on saidid grupeeritud vastavalt oma IP-aadressidele. Kõikide saitide läbimiseks on töö autori poolt koostatud spetsiaalne veebirobot (laeAndmed.php), mis alustab iga saidiga suhtlust vastavale IP-aadressile pordile number 80 saadetava HTTP päringuga, mis on toodud joonisel 5.

```
GET / HTTP/1.0
Host: www.host.ee
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; VeebiUuringuRobot;
     http://math.ut.ee/~veikos/uuring/)
Accept: text/*,application/x-httpd-php
Accept-Language: et, en
Connection: close
```

Joonis 5. Veebiroboti esimene päring etteantud saidile.

Veebiserverite poolt saadud vastusest salvestatakse huvipakkuv osa HTTP päiseinfost ja saidi HTML-sisust ühte faili, et seda hiljem põhjalikult uurida. See fail on lisatud käesoleva töö

lisasse (andmed.xml). HTTP päiseosast pakuvad eeskätt huvi andmed, mis on väljadel `Server`, `Set-Cookie`, `X-Powered-By`, `X-Content-Parsed-By`, `Generator` ja `X-Accelerated-By`. Saitide sisust on olulised saidisisesed veebiviited, mis on olemas HTML-märgenditel `<form>`, `<a>`, `<area>`, `<frame>` ja `<iframe>`. Kui `<a>`-märgendi atribuudid sisaldavad JavaScripti, siis on uurimiseks salvestatud kõik atribuudid, kuna veebiviited võivad esineda ka JavaScripti lähtetekstis.

Saidisisesed veebiviited määratakse järgmiste kriteeriumide alusel. Saidiga kujul www.host.ee loetakse veebiroboti poolt samadeks saitideks need, mille nimi on kujul host.ee, portal.host.ee, web.host.ee või mille nimi algab w-tähega, millele järgneb suvaline arv w-tähti, millele omakorda järgneb suvaline arv numbreid ja sidekriipse (nt wwwww.host.ee, w3.host.ee, www-2.host.ee), ning need saidid, mis erinevad esimese taseme domeeni poolest ja asuvad samal IP-aadressil (nt www12.host.com, host.lt, w.host.net).

Veebirobot läbib igal saidil nii palju lehekülgi, kui palju leidub piisava erinevusega saidisiseseid veebiviiteid (maksimaalselt 100). Näiteks URLid kujul index.php?id=1 ja index.php?id=2 loetakse samadeks URLideks, samuti ka URLid /nimekiri1.html ja /nimekiri2.html. Seega veebirobot ei järgne kõikidele URLidele, vaid ainult neile, mis näivad piisavalt erinevad, sest sealt on suurem tõenäosus leida veel avastamata serveripoolseid tehnoloogiaid. Veebiaadresside piisavat (subjektiivselt määratud) erinevust testitakse PHP funktsiooniga `similar_text()`.

Kui dünaamilise saidi korral ei leitud üheltki läbi vaadatud lehelt veebiaadressi, mis sisaldaks tuntud dünaamiliste failide laiendit, siis otsitakse veebiaadresse ka URLilt /robots.txt [6], mis on standardne Interneti veebirobotite infofail, mille kaudu saidihaldajad saavad esitada nimekirja saidisisestest URLidest, mida veebirobotid antud saidi indekseerimisel peaks ignoreerima.

Käesoleva uuringu jaoks loodud spetsiaalne veebirobot on võimeline

- ✔ järgnema URLidele, mis asuvad saidi suvalisel sügavusel (suhteliste aadresside teisendamine absoluutseteks, mille korral arvestatakse ka `<base>`-elemendiga, mille kaudu on võimalik esitada HTML-dokumendi veebiaadressist erinev absoluutne veebiaadress, mida kasutada saidisiseste suhteliste URLide teisendamisel absoluutseks);
- ✔ kodeerima URLi päringusõnes leiduvaid erisümboleid (*url-encoding*);
- ✔ reageerima ühelt veebiaadressilt teisele ümber suunamistele, arvestades HTTP päistega

(`Location`, `Refresh`), HTML `<meta>`-elemendiga ning JavaScripti `location` objektiga;

- ✔ vahetama suhtlusporti (www.host.ee:81);
- ✔ koguma vajadusel andmeid ka HTTPS-protokolli alusel.

1.5.3. Andmebaasiserverite info kogumine

Töö autori poolt läbi viidud andmebaasiserverite populaarsuse uuringus testitakse tuntud andmebaasiserverite vaikimisi installatsioonil seatud porte. Portide avatust või suletust kontrollitakse iga füüsilise serveri korral programmiga Nmap [4], mis käivitatakse joonisel 6 toodud käsurea abil.

```
nmap -sS -p 1112,1114,1433,1521,2638,3050,3306,3307,4100,4333,5000,5432,7
200,21064,50000 IP-aadress
```

Joonis 6. Andmebaasi tarkvara uuringus kasutatud Nmap väljakutse.

Vastavalt tuntumate andmebaasiserverite tavalistele kuulamisportidele, vaadeldakse käesolevas uurimuses 15 pordinumbrit: mSQL (1112, 1114, 4333) [21], MS SQL Server (1433) [21], Oracle (1521) [8], Sybase (2638, 4100, 5000) [21; 22; 23], Firebird (3050) [24], MySQL (3306, 3307) [21], PostgreSQL (5432) [21], Adabas (7200) [25], Ingres (21064) [26] ja IBM DB2 (50000) [27]. Nmap väljastab iga pordi kohta olekuinfot: avatud (*open*), suletud (*closed*) või varjatud (*filtered*) olek. Avatud olek näitab, et vastava pordi kaudu saab sihtmasinasse luua ühendusi. Ühendusi pole võimalik luua suletud portidega, kuna seal ei eksisteeri ühtegi teenust. Varjatud olek tähendab, et Nmap ei suutnud vastava pordi olekut kindlaks määrata, sest pordi kaitseks on üles seatud mingid turvavahendid (nt tülemüür). Näiteks joonisel 7 esitatud Nmap väljundi põhjal saab tuvastada MySQL andmebaasiserveri tõenäolise eksisteerimise.

```
(The 12 ports scanned but not shown below are in state: closed)
Port        State       Service
1433/tcp    filtered    ms-sql-s
1521/tcp    filtered    oracle
3306/tcp    open        mysql
```

Joonis 7. Programmi Nmap väljundi näide.

Tulemuste analüüsimisel võetakse kõigepealt arvesse avatud pordid, kui avatud porte ei leidu,

siis arvestatakse varjatud portidega, kui avatud portidega. Suletud porte ei arvestata. Kui Nmapi poolt saadud info põhjal paistab masinas kasutuses olevat rohkem kui kaks andmebaasiserverit, siis loetakse tulemuseks, et andmebaasiserverit pole teada.

Nagu teises peatükis avaldatud tulemustest järeldub, annab kirjeldatud vaikeportide uurimine küllaltki ootuspärase hinnangu andmebaasisüsteemide kasutuspopulaarsuse kohta.

1.5.4. Kogutud info põhjal otsuste tegemine

Veebiroboti ja Nmapi poolt kogutud andmetest (andmed.xml) salvestatakse analüüsiskripti (analyys.php) abil tulemuste esitamiseks vajalik info andmebaasi MySQL (andmebaas.sql) ning töös esitatud tabelid ja ülevaated saadakse andmebaasipäringute (ABp2ringud.txt) abil. Andmeid (andmed.xml) analüüsiv PHP skript (analyys.php) teeb otsuseid kasutatavate serveripoolsete tehnoloogiate, operatsioonisüsteemide ja veebiserveri tarkvara kohta vastavalt etteantud kirjelduste failile fingerprints.txt. Kirjelduste faili andmed põhinevad tarkvara standardsel installatsioonil seatavatel väärtustel. See tähendab, kui HTTP päise `Server`-väljal on olemas märksõnad Apache, Unix ja PHP, siis see väljendab Apache veebiserverit koos PHP interpretaatoriga Unix-laadsel operatsioonisüsteemil ja mitte midagi muud. Samuti kui veebisaidil on Internetiaadress, mis viitab selle saidi failile laiendiga jsp, siis järelikult on saidi loomisel kasutatud serveripoolse tehnoloogiana JSP-d. Analoogilises seoses on laiendid php, phtml, php3 PHP-ga, asp ASP-ga, pl Perliga jne. Ühesõnaga kirjelduste failis on ära toodud, missugune info (HTTP päisest, URList) viitab vaikimisi installatsiooni korral missugusele veebiserverile, operatsioonisüsteemile või serveripoolsele tehnoloogiale.

Vastavalt kirjelduste failile omistatakse saidile mingisugus(t)e serveripoolse(te) tehnoloogia (te) kasutamine, kui sait on dünaamiline ehk kui saidi sisust järeldub, et saiti võidakse toetada mingit liiki reaalajas käivitatava(te) serveripoolse(te) programmi(de)ga.

Käesoleva uuringu käigus loetakse dünaamilisteks neid saite, mis

- ✔ kasutavad veebivorme, mis saadavad andmed samal saidil eksisteerivale aadressile, sisaldavad mittetühja `action`-atribuuti, sisaldavad vähemalt kahte `<input>`-elementi, ei sisalda `<form>`-elemendis JavaScripti. Loetletud kitsendused veebivormidega arvestamisel on tingitud kliendipoolsete veebivormide (kujunduse osana, JavaScripti rakendusena) mittearvestamise vajadusega;
- ✔ kasutavad parameetritega URLe, mis viitavad samal saidil eksisteerivale aadressile;
- ✔ omavad saidisiseseid URLe, mis sisaldavad tuntud serveripoolsete tehnoloogiatega

seotud sõnesid „.asp“, „.php“, „.phtm“, „.jsp“, „.jhtml“, „.do“, „servlet/“, „.pl“, „.cfm“, „.shtm“, „.py“, „.nsf/“, „.cgi“, „cgi/“, „cgi-bin/“ (vt täpsemalt kirjelduste failist fingerprints.txt);

- ✔ kasutavad küpsiseid, mille sihtdomeen on sama nagu saidil endalgi; erandina ei loeta dünaamilise saidi lahutamatuks osaks küpsise ASPSESSIONID* kasutamist, sest veebiserver Microsoft IIS kasutab seda küpsist mõningatel juhtudel ka staatiliste saitide korral, ning küpsiste EGSOFT_ID, WEBTRENDS_ID, SITESERVER, Apache, ApacheEE ja ETRACK kasutamist, kuna nende küpsiste kasutamine pole otseselt seotud saidi sisu genereerimisega, vaid kujutavad endast veebilogifailide analüsaatorite kasutajajälituse küpsiseid [28].

Kõikide arvesse läinud veebiaadresside, mis sisaldavad serveripoolsete tehnoloogiate tuntud faililaiendeid või katalooge, eksisteerimine on kontrollitud veebiserveri poolt tagastatava HTTP olekukoodi alusel. Mitteeksisteerivateks ressurssideks loetakse veebiaadressid, mis tagastasid olekukoodid 403, 404, 410 või 5xx.

Otsused selle kohta, missuguseid serveripoolseid tehnoloogiaid on saidi lehekülgede loomisel kasutatud, tehakse kuue komponendi alusel:

- ✔ saidisiseste URLide faililaiendid ja kataloogid;
- ✔ küpsised (päiseväljalt `Set-Cookie`);
- ✔ dünaamiliste vahendite reklaaminfo veebiserveri HTTP päises (väljad `X-Accelerated-By`, `X-Powered-By`, `X-Content-Parsed-By` ja `Generator`);
- ✔ veebiserveri poolt toetatud skriptivahendid (päiseväljalt `Server`);
- ✔ serveripoolsete tehnoloogiate vaikimisi indeksfailide (index.php, default.asp, index.pl, index.jsp jt) olemasolu ja veebiserveri vastused neile failidele esitatud päringutele. Indeksfailidega arvestatakse ainult siis, kui on eelnevalt teada, et sait on dünaamiline ning indeksfailide poolt pakutavate erinevate serveripoolsete skriptivahendite arv on 1 või 2. Kui dünaamilisel saidil kasutatavate serveripoolsete programmide kohta on midagi teada ülejäänud 5 komponendi poolt, siis arvestatakse ainult nende indeksfailidega, mille korral saadi veebiserverilt mitteveateade (HTTP olekukoodid 200, 301, 302). Kui serveripoolsete programmide kohta kohta ei ole ülejäänud 5 komponendil mingit infot või on teada, et kasutatakse CGI programme, siis arvestatakse kõikide indeksfailidega (eelistatult olekukoodidega 200, 301 või 302), mille korral on veebiserveri poolt saadud mingi vastus ning see ei ole olekukoodiga 404 (ei leidu);
- ✔ JavaScripti sisaldava `<A>`-märgendi atribuutide sisu, näiteks

```
<a href="javascript:gotomenu('/display.php?item=23');">.
```

Loetletud kuus infokomponenti aktiveerivad märksõnu, kui neile vastav info sobib märksõna aktiveeriva regulaaravaldisega. Need otsustusreeglid, mida andmete analüüsimisel kasutatakse asuvad failis fingerprints.txt. **Faililaiendi** [29; 30], indeksfailide ning JavaScripti komponendi märksõnad (info serveripoolsete programmide kohta) ja vastavad regulaaravaldised on toodud peale märksõnade seletusi alljärgnevalt.

Serveripoolsed tehnoloogiad on uuringu tulemustes tähistatud järgmiste märksõnadega: märksõnaga „cfml“ on tähistatud Macromedia ColdFusion tehnoloogia [31], milles kasutatav skriptikeel on CFML; „asp“ alla on kokku võetud Microsofti ASP ja ASP.NET tehnoloogiad [32; 33]; „jsp“ hõlmab Sun'i servlett ja JSP tehnoloogiaid [34; 35; 12]; „api“ tähistab veebiserveri API rakendusi, mis on veebiaadressides ära tuntavad paraku vaid Windows operatsioonisüsteemil töötavatel veebiserveritel [36; 37]; „cgi“ alla loetakse kõik CGI programmid [38; 39]; „foxweb“ tähistab Visual FoxPro põhises skriptikeeles FoxWeb kirjutatud veebirakendusi [40]; „lotus“ tähistab IBM Lotus Notes/Domino põhiseid avalikke veebirakendusi [41]; „Usertalk“ tähistab skriptikeelt, millega realiseeritakse serveripoolseid skripte veebiserveris Userland Frontier [42]; „nsp“ all on mõeldud Novell Script Pages (NSP) tehnoloogiat [43], „plsql“ tähistab Oracle'i poolt loodud programmeerimiskeele PL/SQL kasutamist [44]. Ülejäänud märksõnad – „php“ [45], „perl“ [46], „ssi“ [47], „python“ [48], „ruby“ [49] – tähistavad vastavas skriptikeeles realiseeritud serveripoolseid skripte või CGI programme (Perl, Python ja Ruby puhul).

```
cfml \.cfml?$|\.cfml?\W
asp \.aspx?$|\.aspx?\W|\.cs[?\/]|\.mspx$|\.mspx\W
php \.php\d?$|\.php\d?\W|\.phtml?$|\.phtml?\W|PHPSESSID|\.aw\W|automatweb=|\.cns$|\.cns\W
jsp \.jsp$|\.jsp\W|\.jhtml$|\.jhtml\W|jsessionid|\.do$|\.do\W|jserv\/|servlet\/
perl \.e?pl$|\.e?pl\W|^e_\w+\.html|\/e_\w+\.html|\/sympa\/\w
ssi \.sh?tml?$|\.sh?tml?\W
python \.py$|\.py\W
ruby \.rb$|\.rb\W
lotus \.nsf[?\/]
api \.idq$|\.idq[?\/]|\.dll[?\/]|\.idc$|\.idc[?\/]
plsql \/pls(ql)?\/\w
foxweb \.fwx$|\.fwx\W
cgi \.f?cgi$|\.f?cgi\W|cgi[\w-]{0,5}\/[\w?]|\.exe[?\/]
```

Reeglid on kujul <märksõna><tühik><regulaaravaldis>. Ülaltoodud reeglistikust

määrab esimene rida (reegel) näitena järgmist: kui veebiaadressis sisaldub nii URLi lõpus kui ka keskel sõne „.cfm“ või „.cfml“, siis selle põhjal võib järeldada Macromedia ColdFusion tehnoloogia (märksõna „cfml“) kasutust.

Küpsiste komponent annab serveripoolsete skriptide määramisel infot järgmiselt:

```
php PHPSESSID|automatweb|^php|eZSession
asp ASPSESSIONID|ASP\.NET_SessionId
jsp JSESSIONID|JSESSION_ID|jserv|servlet|jrunsessionid
cfm CFID|CFTOKEN
perl EMBPERL|^MailMan
```

Vastavalt veebiserveritesse sisseehitatud funktsionaalsuse põhjal on võimalik teha oletusi serveripoolsete tehnoloogiate kasutuse kohta. Veebiserverit identifitseerivast päiseinfost – väljalt **`Server`** – on võimalik saada serveripoolsete skriptivahendite uuringu jaoks infot järgmiselt:

```
php Apache.+(PHP|Midgard)|^IceWarp WebSrv\/[\d.]{3}$
asp ^Microsoft-IIS|^MicrosoftOfficeWebServer
jsp ^Netscape-Enterprise\/[\d.a sp]+$|^Apache[ -]Coyote\/[\d.]{3,5}$
jsp ^Apache Tomcat\/[\d.]{1,6} \(.+\)$|^IBM_HTTP_Server|^Resin\/[\d.]+$
jsp ^NetWare-Enterprise-Web-Server|^SilverStream Server\/[\d.]+$
jsp ^Netscape-FastTrack|^JavaWebServer\/[\d.]+$|^Tomcat Web Server
jsp ^Jetty|mod_webapp|ApacheJServ|JRun|Resin|^Oracle|mod_jk|mod_jserv
perl Embperl|mod_perl|Perl|^MiniServ|AxKit
python python|PyApache
ruby mod_ruby|Ruby
lotus ^Lotus-Domino(\/[\d.release-]+)?(\(Intl\))?$
UserTalk ^UserLand Frontier\/[\d.b]+-WinNT$
plsql mod_plsql
nsp Novell Script Pages
```

Veebiserveri päiseinfost leitavad **lisaväljad** (`X-Powered-By`, `X-Content-Parsed-By` jt) annavad teavet järgmiselt:

```
php ^PHP|^phpCMS|^Nucleus|^eZ publish$|Roadsend PHP SiteManager
asp ^ASP\.NET
python Python
perl ^Slash
```

Loetletud kuue komponendi poolt aktiveeritud märksõnade komplekti alusel otsustati serveripoolsete tehnoloogiate kasutus saidil reeglitega, mis on kujul:

<vahend(id)><tühik><komponentide info>|<komponentide info>|<komponentide info>|..., kus
<komponentide info>=
=;tehn_url;tehn_cookie;tehn_xpoweredheader;tehn_serverheader;tehn_indexfile;tehn_js;

Järgnevalt on esitatud mõned uurimuses kasutatud reeglinäidised, mis otsustasid serveripoolsete tehnoloogiate kasutuse dünaamiliste saitide korral, kus sidekriips (-) URLi komponendi koha peal tähendab tundmatut faililaiendit (üldjuhul html) sisaldavat parameetritega URLi:

```
php ;;;asp,php;asp;php;;|;;;;perl,php;php;php;|;-;;;php;;;
asp ;asp;;;perl,php;;;|;asp;asp;;asp;asp;;|;-;asp;;asp;;;
java ;;;;UserTalk;;java;|;java;;;java,perl;;;|;java;;;php;;;
perl ;perl;;;jsp,perl,php;;;|;perl;;;asp;;;|;;perl;;;;;
cfm ;cfm;cfm;;asp;;;|;;;;;cfm;;|;cfm;cfm;asp;asp;;;|;cfm;;;asp;;;
ssi ;ssi;;;cgi,perl,php;;;|;;;;;;ssi;|;ssi;;;php;;ssi;
python ;;;python;;;;|;;;python;perl,php;;;|;;;python;perl;;;
UserTalk ;;;;UserTalk;;;
lotus ;lotus;;;lotus;;;
cgi ;cgi;;;asp;;;|;;;;php;;cgi;|;cgi;;;perl,php,python;;;
asp,php ;asp;;php;asp;;;|;asp,php;;php;perl,php;;;|;asp,php;;php;asp;;;
perl,php ;perl,php;php;php;;;php;|;perl,php;;php;perl,php,python;;;
cgi,perl ;perl;;;perl;;cgi,perl;|;-,cgi;;;perl,php;;;|;-,cgi;;;perl;;;
java,php ;java,php;java;;java,php;;;|;java,php;;;perl,php,python;;;
perl,ssi ;perl,ssi;;;;;;|;perl,ssi;;perl;perl,php;;perl;
php,ssi ;php,ssi;;;perl,php,python;;;|;ssi;;;php;;php;|;;;php;php;php;ssi;
```

Esimest rida (otsustusreeglit) tõlgendatakse nii: dünaamilise saidi poolt kasutatavaks serveripoolseks tehnoloogiaks on PHP, kui on teada, et 1) sait on loodud kasutades PHP interpretaatorit ning kõikidele tehnoloogiatele vastavatest indeksfailidest annab server php-laiendi korral kõige positiivsema tulemuse, vaatamata sellele, et väidetavalt kasutatakse saidi loomisel ka ASP.NET tehnoloogiat, mis on serveris sisseehitatult ka toetatud; 2) serveri poolt on eriliselt toetatud Perl ja PHP ning samal ajal leidub php-laiend veebiviitadega kasutatavas JavaScripti lähtetekstis ja PHP indeksfail annab serverist kõige positiivsema tulemuse; 3) serveri poolt on eriliselt toetatud ainult PHP ning saidil kasutatakse parameetritega URLides tundmatut faililaiendit.

Veebiserveri poolt saadud päiseinfo – väljalt **`Server`** – grupeeriti **operatsioonisüsteemi**

tüüpide alla järgmiselt [50; 51; 52]:

```
UNIX (Apache|Tomcat Web Server|Jetty|Rapidsite\/Apa).+(Unix|Linux|Midgard)
UNIX ^MiniServ|^Zeus\/[\d.]+$|^NCSA\/[\d.]+$|^Zope.+linux|^Boa|^CERN
UNIX ^tigershark|^CoffeeMaker\/[\d.]+.+\(Unix\)$|^publicfile$
Windows ^Microsoft-IIS\/[\d.]{3}$|Apache.+\(Win32\)|^Abyss.+\(Win32\)
Windows ^MicrosoftOfficeWebServer|^Ipswitch-IMail\/[\d.]+$
Windows ^SAMBAR[ \d.]+$|^Tomcat Web Server.+\(.+Windows.+\)
Windows ^OmniHTTPd\/[\d.]+$|^UserLand Frontier\/[\d.b]+-WinNT$
Windows ^Falcon Web Server$|^LiteServe\/[\d.]+$|^24Link\/[\d.]+$
Netware ^Netscape-FastTrack.+NetWare$|^Novell Script Pages$
Macintosh ^Apache.+\(Darwin\)
```

Alljärgnevalt on toodud mõningate populaarsemate **veebiserverite** grupeeringud tootjafirma järgi veebiserveri poolt saadud päisevälja **`Server`** alusel:

```
Apache ^(Embperl\/.+ |\.V.+)?Apache|^Tomcat Web Server|^Rapidsite\/Apa
Microsoft ^Microsoft-IIS\/[\d.]{3}|^MicrosoftOfficeWebServer
Netscape ^Netscape-Communications\/[\d.]+$|^Netscape-FastTrack
Netscape ^Netscape-Enterprise\/[\d.a sp]+
IBM ^IBM_HTTP_Server|^Lotus-Domino
```

Veebiserverite ja vastavate operatsioonisüsteemide IP-aadressipõhisel ja domeenipõhisel analüüsil on arvestatud võimalusega, et ühe IP-aadressi või saidi taga võib olla mitu erinevat veebiserverit ja operatsioonisüsteemi. Loomulikult võib ühel saidil olla kasutuses mitu erinevat serveripoolset programmeerimistehnoloogiat.

1.5.5. Veebiotsingu saitide kasutamine

Et hinnata veebiotsingu saitide alusel lihtsalt saadavaid tulemusi serveripoolsete tehnoloogiate populaarsuse kohta, on käesolevas uurimuses võrdluseks samaaegselt kogutud andmeid nii veebiserverite vastustest (HTTP-põhiselt) kui ka veebiotsingu saitidelt. Antud töös on kasutatud kahte veebiotsingu saiti: Eesti veebisaitidel põhinev Neti ja ülemaailmsetel saitidel põhinev Google. Et Google'i tulemustes kajastuksid Eesti veebisaidid, on otsingutulemusi filtreeritud ee-domeeniga.

Otsingumootori Google tulemused ee-domeeni kohta on saadud selle otsinguväljale sisestatud päringutega kujul: `inurl:<otsisõne><tühik>site:ee`.

Näiteks kõikide selliste veebiaadresside arvu saamiseks, mis sisaldavad sõna „php“ ning

paiknevad ee-domeenis, antakse päring „`inurl:php site:ee`“, mille vastus [53] läheb arvesse PHP populaarsuse hindamisel Eesti avalikus veebis.

Otsingumootori Neti tulemused on saadud selle otsinguväljale sisestatud päringutega kujul: `allinanchor:<tühik><otsisõne>`.

Näiteks saame uuringu jaoks teada, et päringule „`allinanchor: php`“ [54] leidub 115419 vastust, päringule „`allinanchor: asp`“ leidub 7553 vastust jne.

Kõik uuringus arvesse läinud päringud ning nende poolt tagastatud veebiaadresside arv on toodud töö lisas olevas failis google_neti_info.txt.

Populaarsusprotsentide leidmisel (jaotises 2.4. Tabel 4 neljas ja viies veerg) on arvestatud ka HTML-faili sisaldavate URLide arvuga.

1.5.6. Uuringu puudused

Serveripoolsete tehnoloogiate kasutuse kohta on vähe uurimusi, sest automaatsed uuringumeetodid annavad ebatäpseid tulemusi. Kui uurida kliendipoolseid vahendeid, siis sellest, kui saidi HTML-sisus leiduvad märgendid `<frame>` või `<frameset>`, saab kindlalt väita, et saidil kasutatakse freime; analoogselt märgendi `<script>` alusel saab 100% kindlalt tuvastada kliendipoolse skriptikeele kasutuse veebileheküljel. Serveripoolsete tehnoloogiate tuvastamisel ei ole tulemus kunagi 100% kindel, sest turvalisuse kaalutlustel ei ole soovitatav serveris toimuvat avalikustada. Seega tegelikkus võib olla hoopis midagi muud, kui HTTP päiste ja saitidel kasutatavate veebiviitade põhjal võib oletada. Serveri administraatorid ei ole üldjuhul huvitatud sellest, et teised teaksid, mis tarkvara on serveris kasutusel või sinna installeeritud. Serveri haldajad võivad väga lihtsalt varjata kasutatavat tarkvara: kustutades täielikult või muutes valeks info, mida veebiserveri väljundist võib igaüks lugeda. Seega mistahes automaatne uuring (sh Netcraft, E-Soft) serveris paikneva tarkvara kohta on ebatäpne ning põhineb infol, mida administraatorid välja paista lasevad. Kui veebiserveri päiseinfost saab välja lugeda, et tegemist on Apache'ga, ei tähenda see, et tegelikkuses ongi tegu veebiserveriga Apache (võib olla näiteks Microsoft IIS), kuna serveri administraatorid võivad veebiserveri päiseinfosse kirjutada suvalist teksti. Analoogiliselt mõjutab see ka operatsioonisüsteemi tüüpide määramist HTTP päise alusel. Samuti kui saidil olevad URLid on laiendiga php, ei tähenda, et reaalselt veebiserverisse on üldse PHP installeeritud, rääkimata selle kasutamisest veebilehe loomisel, kasutuses võib tegelikult olla

hoopis Python või ASP. Veebiserveri haldajad võivad mistahes laiendiga (ka laiendita) failile vastavusse seada mistahes programmeerimiskeele interpretaatori.

Ülalkirjeldatud puudustest tulenevad ebatäpsused eksisteerivad ka käesoleva uuringu tulemustes, mis kajastavad kahjuks mitte tegelikku hetkeseisu, vaid andmeid selle kohta, mida serverite administraatorid välja paista lasevad.

Kui välja paistab üks, aga tegelikkuses on olukord teine, siis selline seis võib tekkida ka paratamatult. Oletame, et algselt oli serveripoolne veebirakendus loodud ASP tehnoloogiat kasutades, kuid hiljem mindi üle PHP kasutamisele ning kirjutati ümber ka olemasoleva ASP rakenduse funktsionaalsus, siis selleks, et mujal saitidel paiknevad veebiviited antud saidile edasi töötaksid, tulebki olemas olnud asp-laiend siduda PHP interpretaatoriga.

Vaatamata serverist tuleva potentsiaalselt ebausaldusväärse info olemasolule, võib siiski eeldada (saab mingil määral kinnitust ka käesoleva uuringu tulemustest), et neid, kes kasutatava tarkvara valeinfoga varjamiseks spetsiaalseid samme ette võtavad, on väga väike osa, mis piisavalt suure valimi korral ei esita ebatõepärast pilti tegeliku olukorra kohta.

Käesolev uuring põhineb seega standardse installatsiooni eeldusel, et serverisse on tarkvara (veebiserver, programmeerimiskeele interpretaator, andmebaas) seatud standardset (vaikimisi) installatsiooni kasutades. See tähendab, et Apache vaikimisi installatsiooni korral on tema päiseinfost võimalik välja lugeda sõna „Apache“, operatsioonisüsteem (nt Unix, Linux, Win32) ja installeeritud moodulid (nt PHP, Perl, Python). Kasutatava serveripoolse programmeerimiskeskkonna vaikimisi installatsiooni korral saab dünaamilisel saidil kasutatava programmeerimisvahendi välja lugeda nii HTTP päiseinfost (päiseväljad `Server`, `X-Powered-By`, `Set-Cookie` jt) kui ka saidisisestest URLidest.

2. Uuringu tulemused

Käesolevas peatükis esitatakse antud töö autori poolt HTTP-protokolli põhjal läbi viidud standardse installatsiooni eeldusel tehtud uuringu tulemused Eesti avaliku veebi serveripoolse keskkonna kohta. Tulemustes kajastub serveripoolsete programmeerimistehnoloogiate, veebiserveri tarkvara, veebiserveri operatsioonisüsteemi tüüpide ja andmebaasiserverite kasutuspopulaarsus. Lisaks kogu valimi põhistele tulemustele on lisatud mitmed piiratud valimi – hostinguserverite, eriserverite ja suuremate saitide – põhised tulemused. Peatüki lõpuosas on esitatud mõningad uuringu tulemustest välja võetavad andmed serverite turvamise valdkonnast, eeskätt tarkvara varjamise poole pealt, ning on tutvustatud veebiserverite ja saitide ülesehituse iseärasustest tulenevaid võimalikke ebatäpsusi uuringu tulemustes.

Käesolev uuring toimus ajavahemikul **02.04.2004 kuni 04.04.2004**, mil veebirobot kogus andmeid ning suhtles veebiserveritega kokku umbes 35 tundi. Veebirobotile anti ette 2177 serverit ning 17 573 saiti, millest uurimuse jaoks läks lõpuks arvesse **1842 serverit** ja **14 317 veebisaiti**.

2.1. Kasutatav tähistus

Käesolevas peatükis esitatud tulemuste tabelites antakse arvandmetele tähendused järgmiste lühendite ja märksõnade abil (tabeli päises):

- dom – domeeninimepõhine, saitidepõhine, virtuaalserverite põhine;
- ip – IP-aadressipõhine, füüsiliste serverite põhine;
- dün – dünaamiliste saitide või dünaamilisi saite omavate serverite põhine;
- top1000saidid – lehekülgede arvu (>478) järgi tuhande kõige suurema saidi põhine;
- top100hosting – saitide arvu (>24) järgi saja kõige suurima veebiserveriteenust pakkuva serveri põhine;
- erisaidid – eriserverite/erisaitide (üks sait ühel serveril) põhine;
- *arv_tabeli_päises* – arv, mis näitab palju on vastavaid saite/IP-aadresse kokku;
- sidekriips (-) tabeli andmelahtris – vastavad andmed on küll olemas, kuid ei oma mõtet või pole tähtsad;
- arv null (0) tabeli andmelahtris – käesoleva uuringu valimist ei leitud vastavas kategoorias ühtegi esinemist uuritud omadusele.

Tabelites toodud protsentuaalsed numbrid võivad märkimisväärselt ületada kogusummas 100%, kuna ühel saidil võib olla kasutuses mitmeid serveripoolseid tehnoloogiaid. Samuti võib ühe saidi või ühe IP-aadressi taga olla mitu erinevat veebiserverit ja operatsioonisüsteemi.

2.2. Staatilised ja dünaamilised saidid

Kuna veebisaitide areng toimub dünaamilisuse suunas, siis on huvitav teada, mis seis on selles osas praegu Eesti veebis. Uurimuse kohaselt on 57% kõikidest saitidest dünaamilised (Tabel 1). Dünaamiliste saitide alla kuuluvad saidid, mis vähemalt ühel aadressil kasutavad mingit serveripoolset programmi. Siia kuuluvad lihtsat küsimuse või tellimuse esitamise veebivormi sisaldavad saidid kuni täielikult andmebaasipõhiste veebisaitideni välja. Kui vaadelda saite, mis asuvad üksinda kogu serveril (erisaidid), siis nende hulgas on dünaamilisi saite 66%. Tuhande kõige suurema saidi hulgas on serveripoolsete programmidega toetatud koguni 922 saiti. Staatilisi saite on protsentuaalselt (45%) kõige rohkem nende saitide hulgas, mis paiknevad jagatud serveritel virtuaalserveritena.

Tabel 1. Veebisaitide jaotus staatilisteks ja dünaamilisteks.

Saidi tüüp	dom, 14317	dom, ip, erisaidid, 1170	dom, top1000-saidid, 1001	ip, 1842	dom, top100-hosting, 10640	ip, top100-hosting, 102
dünaamiline	8166 (57%)	777 (66%)	922 (92%)	1359 (74%)	5851 (55%)	102 (100%)
staatiline	6151 (43%)	393 (34%)	79 (8%)	-	4789 (45%)	-

Kui vaadelda tulemusi IP-aadressipõhiselt, siis 74%-l füüsilistest serveritest kasutatakse koos veebisaitidega ka serveripoolseid programme. Suurematel virtuaalserveriteenust pakkuvatel serveritel kasutatakse serveripoolseid programmeerimistehnoloogiaid 100%-liselt.

2.3. Serveripoolsed tehnoloogiad

Autori poolt läbi viidud serveripoolse veebimaastiku uurimuse peaeesmärk ning kõige tähtsam tulemus avaldub tabelis 2, kus on esitatud dünaamiliste saitide poolt kasutatavad serveripoolsed tehnoloogiad ning nende kasutuspopulaarsus. Tabelis toodud loetelu erinevate serveripoolsete programmeerimisvahendite kohta on parasjagu nii täpne, kui seda HTTP-protokolli alusel on võimalik eristada. Tabelis 2 on esitatud kõik erinevad serveripoolsed tehnoloogiad, mis antud uuringu valimiks olnud saitidel tuvastati – kokku 15.

Spetsiaalselt veebirakenduste realiseerimiseks loodud programmeerimiskeel PHP on konkurentsitult kõige populaarsem vahend dünaamiliste veebisaitide loomisel. PHP-d kasutab 87% dünaamilistest saitidest. CGI programmid, millega sai alguse dünaamiliste veebilehekülgede areng, leiavad tänapäeval kasutust veel 6%-l dünaamilistest saitidest. Nii CGI skriptidena kui ka HTML-faili sisestatuna kasutatakse programmeerimiskeelt Perl 5,3%-l dünaamilistest saitidest. Kuna reeglina kasutatakse CGI programmide realiseerimisel Perli, siis võib Perli tegelik osakaal olla 5,3%-st mõnevõrra suurem, kuna paljudel CGI programmide failidel kasutatava cgi-laiendi alusel ainuüksi ei piisanud vastava programmeerimiskeele määramiseks. PHP põhikonkurendiks loetavat Microsofti ASP tehnoloogiat kasutas vaid 4,3% dünaamilistest saitidest. ASP märksõna alla on käesolevas uuringus grupeeritud nii klassikalise ASP kui ka selle edasiarenduse ASP.NET tehnoloogiate kasutamine. Nagu CGI puhul nii on ka ASP rakenduste korral programmeerimiskeele valik vaba ning see valik ei avaldu veebiviitades kasutatavast faililaiendist (asp, aspx), kuid üldjuhul realiseeritakse ASP skripte keeltes VBScript või Jscript ning ASP.NET veebirakendusi programmeerimiskeeltes Visual Basic või C# [32]. Suhteliselt vana tehnoloogiat SSI (*Server-Side Includes*), mis loodi esimese edasiarendusena CGI programmidele ning mis võimaldas SSI lähteteksti sisestada HTML-faili, kasutatakse praegu veel 2,4%-l saitidest. Väga võimsaks peetavat programmeerimiskeelele Java orienteeritud JSP/Servlett tehnoloogiat kasutatakse Eestis vähem kui 1%-l üldkasutatavatest dünaamilistest saitidest.

Kui vaadelda tulemusi tuhande suurima saidi põhjal, siis on olukord serveripoolsete tehnoloogiate kasutuse osas praktiliselt sama, mis kõikide saitide arvestuses. Oluliselt erinev jaotus aga ilmneb erisaitide põhjal saadud tulemusest. Nimelt PHP kasutuspopulaarsus on siin langenud 65%-le ning teisena järgneval ASP-l on tõusnud 16%-le. Veelgi märgatavama tõusu on teinud JSP, mida dünaamilistel erisaitidel kasutatakse 5,5%.

Kõikide saitide arvestuses saadavaid tulemusi dikteerivad selgelt veebiserveri teenuse pakkujad, kelle serveritel asub suur enamus saite ning pakutavateks serveripoolseteks tehnoloogiateks on üldjuhul vaid PHP ja CGI/Perl. Suurimatel hostinguserveritel kasutatakse PHP-d koguni 93%-l dünaamilistest saitidest.

IP-aadressipõhisest tulemusest on näha, et PHP-d kasutatakse 73%-l füüsilistest serveritest, kus leidub dünaamilisi saite, järgnevad CGI 17%, ASP 13%, Perl 12%, SSI 6,3% ja JSP 4,7%.

JSP ja ka Perli kasutuspopulaarsus võib olla mõnevõrra suurem kui siintoodud numbrid näitavad, sest määramata serveripoolsete tehnoloogiatega saitide hulgas olid HTTP päiseinfo põhjal põhilised kandidaadid kasutatava serveripoolse vahendi määramisel just JSP ja Perl. Peamine põhjus, miks ei õnnestunud määrata serveripoolset tehnoloogiat, oli html-laiendi kasutamine saidisisestel dünaamilistel URLidel. Positiivse tulemusena oli neid saite, kus ei õnnestunud määrata kasutatavat serveripoolset tehnoloogiat, vaid 0,5% kõikidest dünaamilistest saitidest. Erisaitide korral oli vastav näitaja 1,5%

Tabel 2. Serveripoolsete tehnoloogiate kasutus dünaamilistel veebisaitidel.

Vahend	dom, dün, 8166	dom, dün, top1000saidid, 922	dom, ip, dün, erisaidid, 777	dom, dün, top100hosting, 5851	ip, dün, 1359
PHP	7105 (87%)	789 (86%)	503 (65%)	5419 (93%)	993 (73%)
CGI	509 (6,2%)	86 (9,3%)	95 (12%)	297 (5,1%)	228 (17%)
Perl	430 (5,3%)	79 (8,6%)	46 (5,9%)	265 (4,5%)	158 (12%)
ASP	347 (4,3%)	42 (4,6%)	123 (16%)	75 (1,3%)	176 (13%)
SSI	199 (2,4%)	32 (3,5%)	18 (2,3%)	139 (2,4%)	86 (6,3%)
JSP	77 (0,94%)	16 (1,7%)	43 (5,5%)	2 (0,03%)	64 (4,7%)
pole teada	43 (0,53%)	7 (0,76%)	12 (1,5%)	24 (0,41%)	27 (2%)
CFML	36 (0,44%)	3 (0,33%)	8 (1,0%)	26 (0,44%)	12 (0,88%)
Lotus	27 (0,33%)	4 (0,43%)	19 (2,5%)	0	23 (1,7%)
Python	21 (0,26%)	6 (0,65%)	7 (0,9%)	0	14 (1,0%)
API	13 (0,16%)	5 (0,54%)	10 (1,3%)	0	12 (0,88%)
UserTalk	8 (0,10%)	1 (0,11%)	0	0	1 (0,07%)
PL/SQL	4 (0,05%)	2 (0,22%)	2 (0,26%)	0	4 (0,29%)
FoxWeb	2 (0,02%)	0	0	0	2 (0,15%)
NSP	1 (0,01%)	0	1 (0,13%)	0	1 (0,07%)
Ruby	1 (0,01%)	0	1 (0,13%)	0	1 (0,07%)

Üldkasutatavas veebis suhteliselt harva esinevad serveripoolsed vahendid, millega luuakse dünaamilisi veebilehti on

- CFML (*ColdFusion Markup Language*) – Macromedia ColdFusion tehnoloogiaga loodavate veebirakenduste märgendipõhine skriptikeel, mille erimärgendid lisatakse HTML-faili ning mille abil on võimalik lihtsalt luua andmebaasipõhist veebi. Veebiarenduses kasutatakse veel teisigi analoogilisi märgendipõhiseid kommerts skriptivahendeid, näiteks LDML (*Lasso Dynamic Markup Language*) [55] ja iHTML

(*inline HTML*) [56], kuid nende vahendite kasutust Eesti veebis ei õnnestunud käesoleva uuringu valimi põhjal tuvastada;

- ✔ Lotus – IBM-i Lotus tarkvaraperekonnaga (Notes/Domino) seotud veebirakendused, mille loomiseks saab kasutada Domino serveris toetatud spetsiaalset programmeerimiskeelt LotusScript [41];
- ✔ Python ja Ruby – vabavaralised skriptikeeled, mida veebiloomise eesmärgil kasutatakse nii CGI skriptide realiseerimisel kui ka HTML-faili sisestatuna (nagu PHP korralgi);
- ✔ veebiserveri API rakendused – kõige kiiremini töötavad veebirakendused, mis luuakse veebiserverispetsiifilise API alusel ning integreeritakse vastava veebiserveriga. Näiteks Microsoft IIS veebiserveri korral luuakse programmeerimiskeeles C++ või Delphi vastavat API-t (ISAPI) kasutades kompileeritud DLL-fail, mis installeeritakse veebiserverisse ning mis võib täita kõiki veebirakendustele omaseid funktsioone [37]. Käesolevas uuringus leiti vaid ISAPI rakendusi;
- ✔ UserTalk – veebiserveriga UserLand Frontier integreeritud sisuhaldustarkvara poolt kasutatav spetsiaalne skriptikeel, millega on realiseeritud standardseid veebirakendusi, mida vastava sisuhaldussüsteemi poolt pakutakse;
- ✔ PL/SQL – Oracle'i andmebaasisüsteemiga seotud programmeerimiskeel, mida kasutatakse ka andmebaasipõhiste veebirakenduste loomisel;
- ✔ FoxWeb – Visual FoxPro põhjal veebirakenduste loomiseks kohandatud programmeerimiskeel, mille lähtetekst sisestatakse HTML-faili nagu PHP lähtetekstki;
- ✔ NSP (Novell Script Pages) – Novelli poolt loodud (täpselt sama, mis Microsofti ASP/VBScript Windows keskkonnas) dünaamiliste veebilehekülgede loomise tehnoloogia Netware keskkonnas [43].

Kuna veebiserveriteenust pakkuvad serverid on reeglina Unix-laadse operatsioonisüsteemiga (Tabel 8), siis serveripoolsete tehnoloogiate kasutuspopulaarsus kõikide saitide arvestuses väljendab paljuski sama olukorda, mis Unix operatsioonisüsteemidel põhinevate saitide arvestus. Tabeli 3 põhjal on PHP ülivõimsalt (93%) kõige hinnatum vahend Unixil töötavatest dünaamilistest saitidest, järgnevad CGI (6,1%) ja Perl (5,5%). Teistsugune olukord on aga saitidel, mis paiknevad teadaolevalt Windows operatsioonisüsteemidel: 69% dünaamilistest saitidest kasutab ASP-d, 16% PHP-d ja 6,3% CFML-i.

Tabelis 3 toodud andmed põhinevad ainult nendel saitidel, millele vastav operatsioonisüsteemi tüüp oli HTTP päistest määratav. Kahjuks oli operatsioonisüsteemide määramatuse tase koguni 13% nii IP-aadresside kui ka saitide arvestuses (täpsemalt tabelis 8).

Tabel 3. Populaarsemad serveripoolsed tehnoloogiad operatsioonisüsteemi tüüpide lõikes.

Vahend	dom, dün, Unix, 6638	dom, dün, Windows, 489
PHP	6151 (93%)	80 (16%)
CGI	403 (6,1%)	26 (5,3%)
Perl	362 (5,5%)	20 (4,1%)
SSI	157 (2,4%)	3 (0,61%)
JSP	37 (0,56%)	10 (2,0%)
Python	20 (0,3%)	0
ASP	8 (0,12%)	336 (69%)
CFML	3 (0,05%)	31 (6,3%)

2.4. Serveripoolsete tehnoloogiate erinevate uuringute võrdlused

Lisaks alampunktis 2.3. esitatud serveripoolsete tehnoloogiate kasutatavuse uurimusele, mis võttis arvesse nii veebiserveri päiseinfot kui ka saidisiseseid URLe ning mis peaks hindama tegelikku olukorda kõige paremini, viis käesoleva töö autor läbi ka pealiskaudsemaid uuringuid: ainult HTTP päiseinfo analüüsimine ning veebiotsingu saitide poolt indekseeritud veebilehtede arvuga arvestamine.

Erinevustest kolme meetodi – HTTP päis + HTML-sisu, ainult HTTP päis, veebiotsingu saidid – tulemustes annab ülevaate tabel 4. Serveripoolsete tehnoloogiate jaotus veebiotsingu saitide põhise info alusel veebilehekülgede arvu järgi ühtib põhjalikemate meetoditega saadud tulemustega: PHP on selgelt kõige populaarsem, järgnevad enam-vähem võrdselt ASP ja CGI. Ainult HTTP päiseinfot arvestav tulemus näitab seda, mis vahendeid saitide loomisel põhimõtteliselt on võimalik kasutada või mille kasutamine on kõige tõenäolisem. Selle alusel on JSP-d võimalik kasutada vähemalt 8,7%-l kõikidest saitidest, kuid nagu saitide sisust selgub kasutatakse JSP-d vaid 0,5%-l saitidest. Analoogiliselt Perli korral, mille populaarsuseks HTTP päiseinfo alusel hinnatakse 23%, kuid tegelik kasutus jääb vaid 2,7%-le.

HTTP päiste põhise tulemuse (51%) ühtimine PHP tegeliku kasutuspopulaarsusega kõikidel saitidel (50%) on tõenäoliselt juhuslik ning päiseinfopõhine tulemus peaks olema oluliselt suurem, millele selgelt viitavad Perli ja JSP tulemused. PHP väikse populaarsuse on siin tõenäoliselt põhjustanud asjaolu, et erinevalt Perli (mod_perl) ja JSP (mod_jk) Apache moodulitest, saab PHP-d konfigureerides väga lihtsalt keelata PHP reklaamimise HTTP

päiseväljadel `Server` ja `X-Powered-By`.

Tabel 4. Serveripoolsete tehnoloogiate populaarsus erinevate meetoditega mõõdetuna üle kõigi saitide (aprill 2004, Eesti).

Vahend	dom, päis+sisu	dom, päis	Neti	Google (.ee)
PHP	50%	51%	40%	32%
CGI	3,6%	-	2,7%	10%
Perl	2,7%	23%	-	-
ASP	2,4%	4,7%	2,7%	11%
SSI	1,4%	-	0,8%	1,5%
JSP	0,5%	8,7%	0,6%	1,3%
CFML	0,3%	0,1%	0,2%	0,4%
Python	0,2%	1,5%	0,6%	0,2%

Eesti kohta teadaolev domeenipõhine ja ainult HTTP päiseid arvestav uuring serveripoolsete skriptide populaarsuse mõõtmiseks on pärit A. Sibolalt [1], kelle tulemustega on tabelis 5 toodud võrreldavad andmed käesolevast uuringust. Viimase nelja aasta jooksul on võrreldes teistega oluliselt suurenenud selliste saitide hulk, mis potentsiaalselt kasutavad PHP-d, Perli, Pythonit või eriti just JSP-d, ning langenud selliste saitide hulk, mis võiksid kasutada ASP-d.

Tabel 5. Saidipõhine serveripoolsete skriptide võrdlus HTTP päise meetodil saadud tulemustega.

Vahend	dom, päised, 04.2004	dom, päised, Sibola 04.2000
PHP	51%	44%
Perl	23%	20%
JSP	8,7%	
ASP	4,7%	9,7%
Python	1,5%	
pole teada	36%	35%

Kui võrrelda Netcrafti kogu Interneti põhjal saadud HTTP päiseinfo ja domeeninimepõhise tulemusega PHP vs ASP kohta [14], mis näitas 2002. aastal enam-vähem võrdset 24%-list turujaotust mõlemale (PHP veidi eespool), siis Eestis on PHP võrreldes ASP-ga ikka mitmeid kordi populaarsem.

Kui Eestis leiab kommertstehnoloogiatest kõige enam kasutust Microsofti ASP, siis maailma

kontekstis on selleks Macromedia ColdFusion. Netcrafti IP-aadressipõhise ja saitide esilehekülgede URLide alusel saadud tulemus [16] näitas 2004. aasta märtsi seisuga CFML-i kasutust 80 000 IP-aadressil ning ASP ja JSP kasutust 56 000 IP-aadressil (mõlemal juhul). Käesoleva uuringu tulemuste kohaselt kasutati Eestis ASP-d 13%-l dünaamilisi saite omavatest serveritest (176 IP-aadressi), JSP-d 4,7%-l (64 IP-aadressi) ja CFML-i 0,9% (12 IP-aadressi).

2.5. Veebiserverid

Käesoleva töö autori poolt läbi viidud uurimuse tulemus veebiserveri tarkvara kasutatavuse kohta on esitatud tabelis 6, kus on loetletud kõik leitud veebiserverite tüübid. Kasutatava tarkvara kindlaksmääramisel on kasutatud veebiserverite poolt tagastatud päiseinfot, mille alusel jäi tundmatuks vaid neljal saidil (14317-st) kasutatav veebiserveri tarkvara. Vaieldamatult kõige populaarsemad on Apache veebiserverid, mida kasutatakse 79%-l füüsilistest serveritest, järgnevad 17%-ga Microsofti, 2%-ga IBM-i ja 1,3%-ga Netscape'i veebiserverid. Eriserveritel kasutatakse Apache 6% võrra vähem (73%), mille arvelt leiavad rohkem kasutust Microsofti (21%), IBM-i (2,7%) ja Netscape'i (1,9%) veebiserverid. Suuremate veebiserveriteenuse pakkujate number üks valik veebiserveri tarkvara hulgast on kindlalt 98%-ga Apache, mis omakorda annab saidipõhistes tulemustes samuti Apache'le suure ülekaalu.

Tabel 6. Veebiserverite kasutuspopulaarsus Eestis.

Veebiserver	ip, 1842	ip, dom, erisaidid, 1170	ip, top100-hosting, 102	dom, top1000-saidid, 1001	dom, 14317
Apache	1452 (79%)	849 (73%)	100 (98%)	927 (93%)	13542 (95%)
Microsoft	315 (17%)	248 (21%)	3 (3%)	60 (6%)	672 (4,7%)
IBM	37 (2%)	31 (2,7%)	0	6 (0,6%)	46 (0,32%)
Netscape	23 (1,3%)	22 (1,9%)	0	6 (0,6%)	23 (0,16%)
Zope	12 (0,65%)	8 (0,68%)	0	3 (0,3%)	18 (0,13%)
Oracle	4 (0,22%)	3 (0,26%)	0	2 (0,2%)	4 (0,03%)
pole teada	4 (0,22%)	4 (0,34%)	0	1 (0,1%)	4 (0,03%)
Resin	3 (0,16%)	2 (0,17%)	0	1 (0,1%)	3 (0,02%)
NetWare-Enterprise	2 (0,11%)	2 (0,17%)	0	0	2 (0,01%)
Jetty	1 (0,05%)	1 (0,09%)	0	0	1 (0,01%)
Roxen	1 (0,05%)	1 (0,09%)	0	1 (0,1%)	1 (0,01%)
Tigershark	1 (0,05%)	1 (0,09%)	0	0	1 (0,01%)
UserLand-Frontier	1 (0,05%)	0	0	1 (0,1%)	9 (0,06%)
SilverStream	1 (0,05%)	1 (0,09%)	0	0	1 (0,01%)
Falcon	1 (0,05%)	1 (0,09%)	0	0	1 (0,01%)
Abyss	1 (0,05%)	1 (0,09%)	0	0	1 (0,01%)
NCSA	1 (0,05%)	1 (0,09%)	0	1 (0,1%)	1 (0,01%)
Zeus	1 (0,05%)	1 (0,09%)	0	0	1 (0,01%)
JavaWebServer	1 (0,05%)	1 (0,09%)	0	0	1 (0,01%)
24Link	1 (0,05%)	1 (0,09%)	0	0	1 (0,01%)

Serveritel, mille operatsioonisüsteemina tuvastati Windows, kasutati Apache veebiservereid 14%-l.

Võrreldes neli aastat tagasi olnud seisuga (Tabel 7), on vähenenud nende saitide hulk, mida serveerivad Microsofti veebiserverid, ning suurenenud oluliselt Apache tarkvara poolt teenindatavate saitide hulk. Füüsiliste serverite arvestuses ei pruugi muutused olla samasugused, sest kui arvestada, et neli aastat tagasi võis domeeninimepõhine tulemus olla enam-vähem sama, mis IP-aadressipõhine tulemus, siis võrreldes praegu saadud IP-aadressipõhise tulemusega oleksid muutused hoopis vastupidises suunas – rohkem Microsofti tarkvara installatsioone võrreldes Apache'ga.

Tabel 7. Virtuaalserveritepõhine populaarsemate veebiserverite võrdlus.

Veebiserver	dom, 04.2004	dom, Sibola 04.2000
Apache	95%	87%
Microsoft IIS	4,7%	10%
IBM Lotus-Domino	0,3%	0,7%
Netscape	0,16%	1,0%
pole teada	0,03%	1,0%
muud	0,34%	0,93%

Võrreldes muu maailmaga, on Eestis Apache veebiserverid oluliselt populaarsemad ning Microsofti omad vähem. Kogu Internetti hõlmav domeeninimepõhine tulemus veebiserveri tarkvara turujaotuse kohta firma E-Soft Inc andmetel [10] näitab 2004. aasta aprilli seisuga koguni 23%-list osakaalu Microsoftile ja 70%-list Apache'le.

2.6. Operatsioonisüsteemid

Käesoleva uurimuse tulemus veebiserveritega kasutatavate operatsioonisüsteemide kohta on esitatud tabelis 8. Operatsioonisüsteemi tüübi kindlaksmääramisel on kasutatud veebiserverite poolt väljastatavat HTTP päiseinfot. Turuosa jaguneb kahe erineva operatsioonisüsteemi tüübi Windows ja Unix (peamiselt Linux) vahel. Unix-laadset operatsioonisüsteemi kasutatakse vähemalt 67%-l serveritel ning Windowsi 20%-l serveritel. Suurel osal (13%) serveritest ei õnnestunud operatsioonisüsteemi määrata, kuna vastavat infot ei saa paljude veebiserverite (IBM, Netscape jt) väljundist lugeda ning lisaks saab Apache veebiserveri konfigureerimisel väga lihtsalt ära keelata operatsioonisüsteemi info näitamise. Arvestades, et 72% serveritest, kus operatsioonisüsteem jäi teadmata, kasutati veebiserverina Apache't ning 95% Apache veebiserveritest asub Unix-laadsel operatsioonisüsteemil, siis võib oletada, et vähemalt 70% teadmata operatsioonisüsteemiga serveritest kasutab Unix-laadset operatsioonisüsteemi. Seega võib kõikide serverite alusel hinnata Unixi populaarsuseks vähemalt 76% ning Windowsile maksimaalselt 24%.

Nagu tabelist 8 võib veel näha, siis kasutatakse eriserveritel operatsioonisüsteemina Windowsi mõnevõrra rohkem (vähemalt 25%) ning veebiserveriteenust pakkuvatel serveritel oluliselt vähem (umbes 3%) võrreldes Unixiga. Saitide arvestuses on näha, et 5% kõikidest saitidest on seotud Windows operatsioonisüsteemiga ning vähemalt 81% Unixiga. Olukord on sama ka suurimate saitide arvestuses.

Tabel 8. Koos veebiserveritega kasutatavate operatsioonisüsteemi tüüpide jaotus.

Op-süs tüüp	ip, 1842	ip, dün, 1359	dom, ip, erisaidid, 1170	ip, top100-hosting, 102	dom, top1000-saidid, 1001	dom, 14317
Unix	1241 (67%)	924 (68%)	719 (61%)	90 (88%)	804 (80%)	11609 (81%)
Windows	372 (20%)	256 (19%)	295 (25%)	3 (3%)	66 (6,6%)	749 (5,2%)
Netware	1 (0,05%)	1 (0,07%)	1 (0,09%)	0	0	1 (0,01%)
MacOS	1 (0,05%)	1 (0,07%)	0	0	0	1 (0,01%)
pole teada	234 (12,7%)	184 (13,5%)	158 (13,5%)	10 (9,8%)	134 (13,4%)	1962 (13,7%)

Eesti serveritel kasutatakse operatsioonisüsteemina Windowsi oluliselt vähem kui mujal maailmas. Netcrafti andmetel [17] kasutatakse serveritel (IP-aadressipõhiselt) Windowsi kokkuvõttes isegi rohkem kui Unix-laadseid operatsioonisüsteeme.

Võrreldes nelja aasta taguse olukorraga (Tabel 9), on serverite haldajad hakanud oluliselt rohkem varjama kasutatavat operatsioonisüsteemi, mis eeskätt tuleneb Apache veebiserveri konfigureerimisest nii, et operatsioonisüsteemi info väljastamine keelatakse. Saite, millele vastavat operatsioonisüsteemi tüüpi ei õnnestunud määrata, oli kokku koguni 13,7% kõikidest saitidest, millest 96% (13,2% kõikidest saitidest) moodustasid Apache'l baseeruvad saidid. Arvestades, et 95% Apache veebiserveritest asub Unixil, siis vaatamata suurele määramatuse osakaalule saab siiski kindlalt järeldada, et viimase nelja aasta jooksul on vähenenud selliste saitide hulk, mis asuvad Windows operatsioonisüsteemil. Kindlasti on siin oma mõju veebiserveriteenuse pakkujatel, kes kasutavad valdavalt Unixit.

Tabel 9. Virtuaalserveritepõhine operatsioonisüsteemi tüüpide jaotus.

Op-süs tüüp	dom, 04.2004	dom, Sibola 04.2000
UNIX	81,1%	81,6%
Windows	5,2%	10,9%
Netware	0,01%	0,29%
pole teada	13,7%	7,2%

2.7. Andmebaasiserverid

Kuna eelmistes jaotistes esitatud uuringutulemused näivad tõepärastena ning sisaldavad piisavalt vähe teadmatust, siis kasutatakse tarkvara standardse installatsiooni omadusi ära ka

andmebaasiserverite kasutuse määramisel. Et hinnata andmebaasipõhise veebi loomiseks kasutatavate andmebaasiserverite populaarsust, on käesolevas uuringus appi võetud turvalisuse kontrollimise ja võrguanalüüsimise programm Nmap, mida kasutati iga valimis olnud IP-aadressi korral ning uuriti tuntumate andmebaasiserverite vaikimisi määratud kuulamisportide avatust.

Vaatamata ebatäpsele uuringumeetodile on tulemus (Tabel 10), et dünaamilisi saite serveerivatest serveritest vähemalt 37%-l võidakse kasutada andmebaasipõhiseid saite, täiesti usutav ja reaalsena tunduv, arvestades, et dünaamiliseks on siinjuures loetud ka neid saite, mille veebilehel näiteks näidatakse serveri kellaaega või mille veebileheküljed paneb staatilisest päisest, sisust ja jalusest kokku SSI skript.

Serveri administraatorite vaikeinstallatsioonide lembusele viitab käesolevas tulemuses ka andmebaasiserverite omavaheline loogiline järjestus ning turuosa jaotus. Teades, et PHP on ülivõimsalt kõige populaarsem serveripoolne skriptivahend, siis peabki andmebaasipõhise veebi juures kasutatavatest andmebaasidest olema konkurentsitult kõige populaarsem MySQL. Samuti peab MySQL-i järgselt tulema PostgreSQL, mis on populaarsuselt teine andmebaasisüsteem, millest räägitakse PHP-ga seoses. Andmebaasipõhiste veebirakenduste loomises PHP järel kindlasti teisel kohal oleva Microsofti ASP tehnoloogia tõttu on seletatav ka MS SQL Serveri kolmas koht andmebaasisüsteemide pingereas.

Tabel 10. Tuntumate andmebaasiserverite jaotus.

Andmebaas	**ip, dün, 1359**	**ip, 1842**
MySQL	424 (31%)	530 (29%)
PostgreSQL	69 (5,1%)	81 (4,4%)
MS SQL Server	25 (1,8%)	30 (1,6%)
Sybase	18 (1,3%)	21 (1,1%)
Firebird	16 (1,2%)	21 (1,1%)
IBM DB2	6 (0,44%)	7 (0,38%)
Oracle	5 (0,37%)	5 (0,27%)
Adabas	1 (0,07%)	2 (0,11%)
mSQL	1 (0,07%)	1 (0,05%)
pole teada + pole kasutusel	860 (63%)	1220 (66%)

2.8. Serveripoolse tarkvara varjamine

Siin alampunktis esitatakse tulemused, mis selgitavad, kui paljud serveri administraatorid on vaeva näinud kasutatava serveripoolse tarkvara varjamisega, kui palju harrastatakse tegutseda ebastandardselt ning kui suures ulatuses võib käesoleva uuringu tulemuste korrektsuses kahelda.

Ainult neljal veebiserveril/saidil oli muudetud väljastatavaid HTTP päiseid nii, et polnud võimalik tuvastada vastavat veebiserveri tarkvara. Seega võib arvata, et kui HTTP väljundis üldse on varjamise eesmärgil muudetud serveritarkvara spetsiifilist infot, siis on sinna paigutatud pigem võõra veebiserveri tarkvara kohta käiv info, näiteks Microsofti veebiserveri tutvustamine Apache veebiserverina. Seetõttu käesoleva uuringu tulemustes tundub kahtlane, et 11 saiti, mis asusid Apache veebiserveril ja millest 8 paiknes Linuxil, kasutasid serveripool väidetavalt Microsofti ASP tehnoloogiat, kuid samas on selline lahendus reaalselt täiesti võimalik [57].

Et veebiserveri poolt väljastatava HTTP päiseinfo vastu tuntakse serverihaldajate poolt huvi ning üritatakse igasugu liigset infot mitte väljastada, viitab asjaolu, et 12%-l serveritest, kus töötas Apache veebiserver, ei õnnestunud määrata vastavat operatsioonisüsteemi, kuna seda infot HTTP päises ei leidunud, mis Apache vaikimisi konfiguratsiooni korral aga peaks olemas olema.

Kuuel serveril oli kasutatav serveripoolne tehnoloogia varjatud faililaiendiga, mis üldtuntult midagi ei tähenda (dhtml, jge, dvl, krv, ekl, mec, brc). Põhiline faililaiend oli html, millega serveripool seoti mingisugune skriptiinterpretaator. Seda võimalust kasutas koguni 3,6% dünaamilistest saitidest. Ilma faililaiendita dünaamilisi URLe kasutas 3,7% dünaamilistest saitidest. Dünaamilisi saite omavatest serveritest 2%-l leidus selliseid saite, millel kasutatav serveripoolne programmeerimisvahend jäi käesoleva uuringu mõttes tundmatuks.

Tulemustest selgus, et vähemalt 10% serverite haldajaist on serveri kaitseks midagi ette võtnud või osanud valida turvalist tarkvara või võrgu vaheseadmeid. Just nii paljudel juhtudel andis paljude häkkerite tööriist Nmap tulemuseks, et server on maas, kuigi tegelikult veebiserver seal täielikult funktsioneeris.

Tõsiseid turvafanaatikuid on Eesti veebihaldurite seas 15 inimest, sest nii paljudelt IP-aadressidelt käidi käesoleva uuringuga seotud veebilehel, mille aadressi (roboti päringus

joonisel 5 jaotises 1.5.2.) teadsid ainult need inimesed, kes on uurinud oma veebiserveri logisid. Märkimisväärne on veebihaldurite reageerimiskiirus: kõik külastused nimetatud veebilehele tehti ajavahemikul 02.04.2004 kuni 05.04.2004 (veebirobot tegutses intensiivselt 01.04.2004 – 04.04.2004), kusjuures külastuste logimine nimetatud veebilehel toimis kuu aega enne (veebiroboti testimisperiood) kui ka kuu aega peale veebiroboti tegutsemist.

2.9. Võimalikud ebatäpsuste põhjused uuringu tulemustes

Vaatamata leheküljel 20 (saidi dünaamika komponentide loetelu esimeses punktis) kirjeldatud veebivormidega arvestamise kriteeriumitele, võivad veebivormid siiski põhjustada mõningate staatiliste saitide määramise dünaamilisteks, kui veebivormi kasutatakse antud saitidel vaid kliendipoolses rakenduses või saidi kujunduse osana.

Uuringu tulemustesse toob ebatäpsust asjaolu, et JavaScripti kasutades võib linkidega seotud URL tähendada tegelikkuses hoopis võõra saidi URLi või mitteeksisteerivat saidisisest URLi. Seega võidakse staatiline sait määrata dünaamiliseks. Kui kõikidele linkidele vastavad URLid pannakse JavaScripti poolt programselt kokku või kui linkidele/suunamistele saab järgneda ainult JavaScripti toetav veebivaataja, siis märgitakse tegelikkuses võibolla dünaamiline sait staatiliseks. Analoogselt, kui kõik URLid on peidetud Flash-objekti, siis märgitakse ka vastav sait staatiliseks, kuna lihtne veebirobot neid URLe ei näe.

Kui saidi sisu tuleb mingilt muult veebiaadressilt (näiteks freimidena) või kui kõik dünaamilised veebiviidad kuuluvad teisele saidile, siis nende mittearvestamise tõttu muutub antud sait käesoleva uuringu mõttes staatiliseks.

Mõningate veebiserverite korral tekitasid uurimuse tulemustesse palju ebatäpsusi nende poolt väljastatavad „ilusad“ (kasutajasõbralikud) veateated. Selle asemel, et mitteeksisteeriva veebilehe või koguni saidi korral väljastada HTTP-protokolli abil veateade (olekukoodiga *404 Not Found* vms), väljastati hoopis edu tähistav teade (*200 OK*), mis tähendab, et lehekülg on olemas ja täielikult töökorras ning lehekülje sisuna esitati mingi vaikimisi seatud lehekülg, mis veaolukordadel välja kutsutakse.

Dünaamiliste saitide määramisel tekitas raskusi serveripoolne dünaamiliste URLide teisendamine staatilisteks URLideks (*url rewriting*), mille eesmärk on koledaid (parameetritega) URLe esitada ilusamatena ning veebirobotite jaoks meelsamini järgitavatena

[58]. Näiteks võib dünaamilise URLi

http://www.host.ee/index.php?section=15243&oid=19978

esitada staatilisena näivana mitmel erineval viisil:

http://www.host.ee/index.php/section=15243/oid=19978/

http://www.host.ee/index/section=15243/oid=19978/

http://www.host.ee/section=15243/oid=19978/

http://www.host.ee/section/15243/oid/19978/

http://www.host.ee/15243/19978/

http://www.host.ee/15243/19978

Reeglid (jaotises 1.5.4.), mille alusel otsustati saitidel konkreetsete serveripoolsete tehnoloogiate kasutus, olid üldised ning üksikute konkreetsete saitide korral võivad anda vale tulemuse.

Raskesti tuvastatavaid vigu põhjustavad tulemustes ühe serveripoolse tehnoloogia varjamine teisele tehnoloogiale omaste tunnustega. Näiteks asp asemel php faililaiendi kasutamine või küpsise PHPSESSID asemel CFID kasutamine.

Kuna käesolev uuring põhines tarkvara standardsel installatsioonil põhinevatel väärtustel, siis tekitab ebatäpsusi kõik, mis pole standardne või on lihtsale veebirobotile üle jõu käiv.

Kokkuvõte

Käesolevas töös on teostatud uurimus Eesti veebi serveripoolsest keskkonnast: veebiserveri tarkvarast, operatsioonisüsteemidest ning dünaamilise veebi loomiseks kasutatavatest programmeerimistehnoloogiatest. Nimetatud komponentide kasutuspopulaarsuse määramisel on kasutatud HTTP-protokolli, mille alusel on automaatselt kogutud informatsiooni Eesti veebiserveritelt ja saitidelt. Võrreldes olemasolevate analoogiliste uurimustega, on käesolevas töös kajastatud uuritavat valdkonda põhjalikumalt, vaadeldes tulemusi nii domeeninimepõhiselt, IP-aadressipõhiselt kui ka eritunnustega valimite põhjal. Erilise tähelepanu all on serveripoolsed programmid, mille kohta pole töö autoril siiani õnnestunud leida põhjalikku uurimust. Serveripoolsete tehnoloogiate populaarsuse hindamisel on antud töös lähtutud HTML-sisus leiduvatest veebiaadressidest, mis võivad paikneda veebisaidi kuitahes sügaval leheküljel. Kombineerides HTML-sisust saadud tulemusi HTTP päiseridadelt saadud infoga, oli võimalik serveripoolne tehnoloogia määrata 99%-l dünaamilistest saitidest.

Ehkki veebiserveri poolt väljastatavaid HTTP päiseridu ning serveripoolsetele tehnoloogiatele vastavaid katalooge ja faililaiendeid saavad serverihaldajad muuta oma suva järgi, ei näe töö autor massiliselt mõjuvat põhjust, miks peaks esitatama valeinfot kasutatava veebiserveri tarkvara ja programmeerimisvahendite kohta. Eeldatavasti on selliste serverite hulk väike ning ei mõjuta töös esitatud tulemusi. Käesoleva uuringu andmetel oli veebiserveri tarkvara tundmatuseni peidetud vaid neljal saidil 14317-st ja neljal serveril 1842-st, html-laiendit parameetritega URLides kasutas 3,6% dünaamilistest saitidest.

Standardse (vaikimisi) tarkvara installatsiooni eeldusel põhineva käesoleva uuringu tulemustest selgus, et valimiks olnud saitidest 57% olid dünaamilised, millest 87% kasutab PHP-d, 6,2% CGI programme, 5,3% Perli, 4,3% ASP-d, 2,4% SSI-d ja 1% JSP-d. Ülejäänud 9 tuvastatud programmeerimisvahendit jäid populaarsuselt alla 0,5%. Tuhande suurima saidi arvestuses erilisi muutusi polnud, küll aga erisaitide arvestuses, kus PHP-d kasutas 65% dünaamilistest erisaitidest, ASP-d 16%, CGI programme 12%, Perli 5,9% ja JSP-d 5,5%. Suurimatel hostinguserveritel kasutas PHP-d 93% dünaamilistest saitidest, CGI programme 5,1%, Perli 4,5%, SSI-d 2,4% ja ASP-d 1,3%.

Apache veebiserverid olid kasutuses 79%-l füüsilistest serveritest, Microsofti veebiserverid aga 17%-l. Domeeninimepõhise arvestuse kohaselt serveeriti 95% kõikidest saitidest Apache

veebiserveritel, Microsofti osakaal siin oli vaid 4,7%. Eriserverite arvestuses oli Microsofti osakaal 21%, kuid suurematel hostinguserveritel vaid 3%.

Veebiservereid omavatest füüsilistest serveritest kasutas Unix-laadset operatsioonisüsteemi hinnanguliselt vähemalt 76% ning Windowsi maksimaalselt 24%. Windowsi populaarsus võrreldes Unixiga on mõnevõrra suurem erisaitidel ning oluliselt väiksem hostinguserveritel.

Serveripoolse keskkonna osana vaadeldi ka andmebaasiservereid. Dünaamilisi saite omavatest serveritest paistis Nmap programmi abil teostatud portide analüüsi alusel olevat MySQL installeeritud 31%-l, PostgreSQL 5,1%-l ja MS SQL Server 1,8%-l.

Võrreldes ülemaailmsete analoogiliste uuringutega, kasutatakse Eestis vabatarkvara (Linux, Apache, PHP) oluliselt rohkem.

Antud töös valminud vahendeid on võimalik kasutada regulaarselt (näiteks iga aasta) analoogiliste uuringute läbiviimiseks. See annaks võrdlusmaterjali, mis võimaldaks näha muutusi Eesti avaliku veebi serveripoolses keskkonnas.

Tänu A. Sibola uuringule, oli osaliselt võimalik võrrelda käesolevaid tulemusi nelja aasta taguse olukorraga, mis domeeninimepõhiselt näitab igas valdkonnas vabavara populaarsuse suurenemist ning kommertsvara vähenemist Eesti veebimaastikul.

The survey of server-side environment of web sites in Estonia

Veiko Sang

Abstract

When the web transitioned from a publishing to an interactive e-commerce medium, server-side developments have blossomed and many web technologies have been implemented and adapted. This survey tries to measure the market share of different server-side technologies based on web sites and web servers related to Estonia. Data is gathered via HTTP-based communication where the information taken into account resides in local URLs (from HTML page source) and HTTP header fields, which very often expose the usage of certain web server software, operating system and server-side technology. The results presented by this survey unfortunately include potentially incorrect or misleading data received from servers that are secured through obscurity.

The results presented in this work are based on many different aspects counting all domains, all IP addresses, dynamic web sites, the biggest web sites, dedicated servers and web hosting servers.

According to this survey PHP is the most widely used scripting language with 87% of dynamic web sites, followed by CGI programs 6.2%, Perl 5.3%, ASP 4.3%, SSI 2.4% and JSP 1%. The percentages are pretty much the same when counting the most biggest web sites. Considerably different situation is among dedicated servers where PHP is used by 65% of dynamic web sites, followed by ASP 16%, CGI 12%, Perl 5.9% and JSP 5.5%.

Apache web servers are used by 79% of all servers, where Microsoft has 17%. When counting by all domains Apache has a market share of 95% and Microsoft 4.7%. Unix-like operating systems are used at least in 76% of all web related servers and Windows is used at most 24%.

Comparing to Netcraft's reports this survey reveals that open source and free software (Linux, Apache, PHP) is utilized in percentage terms more often in Estonia than it is in the world as a whole.

Kirjandus

Kõik veebiallikad on kontrollitud 04.05.2004.

1. A. Sibola. *Veebiinfosüsteemid*. Magistritöö, 2000, 83 lk. http://kodu.neti.ee/~aulis/Opingud/AulisMag/
2. V. Sang. *Vahendid andmebaasipõhiste veebilehekülgede tegemiseks*. Bakalaureusetöö, 2002, 67 lk. http://math.ut.ee/~veikos/baktoo/
3. The Internet Society. *Hypertext Transfer Protocol – HTTP/1.1*. 1999 http://www.w3.org/Protocols/rfc2616/rfc2616.html
4. *Nmap Site*. http://www.insecure.org/nmap/
5. J. Lim. *Measuring PHP, JSP, ASP, CFM Popularity*. 04.03.2002 http://php.weblogs.com/discuss/msgReader$178
6. *All About Search Indexing Robots and Spiders*. 18.06.2002 http://www.searchtools.com/robots/
7. D. Sullivan. *Invisible Web Gets Deeper*. 02.08.2000 http://searchenginewatch.com/sereport/article.php/2162871
8. M. Rowe. *Identifying Oracle database installations during a network scan.* http://www.pentest.co.uk/documents/ora_db_on_network.htm
9. Netcraft. *What is the market share of the different servers?*. http://www.netcraft.com/survey/Reports/index.html
10. E-Soft Inc. *Internet Survey Report Archive.* http://www.securityspace.com/s_survey/archive.html?mondir=/200401
11. Netcraft Ltd. *JSP continues fast growth, on a surprisingly diverse set of operating systems*. 23.07.2003 http://news.netcraft.com/archives/2003/07/23/jsp_continues_fast_growth_on_a_surprisingly_diverse_set_of_operating_systems.html
12. Netcraft Ltd. *Java Servlet Engines*. 10.04.2003 http://news.netcraft.com/archives/2003/04/10/java_servlet_engines.html
13. C. Babcock. *Open-Source Scripting Language Becoming Dominant*. 06.11.2003 http://www.informationweek.com/story/showArticle.jhtml?articleID=16000533
14. Ramat Gan. *PHP Overtakes Microsoft's ASP as Web's #1 Server-side Scripting Language.* 05.06.2002 http://www.zend.com/news/zendpr.php?id=49
15. Netcraft Ltd. *PHP growing surprisingly strongly on Windows*. 30.08.2003 http://news.netcraft.com/archives/2003/08/30/php_growing_surprisingly_strongly_on_windows.html
16. Netcraft Ltd. *ASP.NET Overtakes JSP and Java Servlets*. 23.03.2004

http://news.netcraft.com/archives/2004/03/23/aspnet_overtakes_jsp_and_java_servlets.html

17. D. Wheeler.*Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers!*. 31.12.2003 http://www.dwheeler.com/oss_fs_why.html#market_share
18. Port80 Software, Inc. *Port80 Surveys the Top 1000 Corporations' Web Servers*. http://www.port80software.com/surveys/top1000webservers/
19. Elion. *Eesti WWW Serverid*. http://www.neti.ee/cgi-bin/serverid
20. Elion. *Eesti WWW Top 100*. http://www.neti.ee/info/top100.html
21. *Services files*. 02.2004 http://www.graffiti.com/services
22. Systinet Corp. *Sybase ASE 12.5*. http://www.systinet.com/doc/wasp_uddi/uddi/sybasease12.0.htm
23. DataDirect Technologies. *Connecting to Sybase*. http://www.datadirect-technologies.com/download/docs/dotnet/dotnetref/netquik.html#wp927214
24. M. Farooqi. *Introduction to the Firebird Database*. 06.02.2004 http://www.linuxjournal.com/article.php?sid=7010
25. *JDBC driver for Adabas D*. http://beta1.wi-inf.uni-essen.de/adabas/jdbc/jdbc.htm
26. Process Software Corporation. *Installing and Configuring INGRES/Net*. http://vms.process.com/tcpware/iccuconf.htm#E80E58
27. IBM. *Installing DB2 Servers*. http://www-306.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/document.d2w/report?fn=db2v7ixdb2ix14.htm#HDRUNIXSERVER
28. *Browser bug opens cookie files*. http://privacy.net/cookiebug/
29. *File Extensions Windows/OS2/Apple/UNIX*. http://www.icdatamaster.com
30. *The File Extension Source*. http://filext.com
31. Macromedia, Inc. *Developing ColdFusion MX Applications with CFML*. http://livedocs.macromedia.com/coldfusion/6/Developing_ColdFusion_MX_Applications_with_CFML/contents.htm
32. D. Farnsworth-Livingston. *An Overview in the Differences Between ASP and ASP.net*. http://www.tutorial-web.com/asp.net/index.aspx
33. D. Hurst. *Share session state between ASP and ASP.NET apps*. 24.02.2004 http://searchvb.techtarget.com/tip/1,289483,sid8_gci951935,00.html
34. Oracle Corporation. *Servlet and JSP Technical Background*. http://otn.oracle.com/docs/tech/java/oc4j/Jsp1131/tecbkgnd.htm
35. *mod_jk*. http://jakarta.apache.org/tomcat/tomcat-4.1-doc/jk2/
36. R. Lee. *Dynamic Web Pages Using ODBC Through Microsoft's IIS*. http://www.interex.org/pubcontent/interact/sept97/08tam2/tam2.html

37. M. Mousavi. *What an ISAPI extension is?* 05.10.2001 http://www.codeproject.com/isapi/isapi_extensions.asp
38. *CGI Links and Resources*. http://www.webreference.com/programming/cgi.html
39. M. Slemko. *Micro$oft's attempt at FrontPage 98 server-side extensions for Apache.* http://www.insecure.org/sploits/frontpage.server_side_apache_extensions.html
40. Aegis Group. *FoxWeb Documentation*. http://www.foxweb.com/document/index.htm?page=/document/program.htm
41. J. Chamberlain. *An alternative to the OpenServer URL command.* 01.05.2001 http://www-10.lotus.com/ldd/today.nsf/0/ca8ba86a52afb7d685256a3f004b7143?OpenDocument
42. D. Winer. *Website and Scripting Tutorial*. 29.08.1999 http://frontier.userland.com/tutorial/
43. *Novell Script for NetWare.* http://developer.novell.com/ndk/doc/nscript/index.html?page=/ndk/doc/nscript/nsc1_enu/data/abzcigw.html
44. *mod_plsql*. http://www.skywayradio.com/tech/mod_plsql/
45. PHP Documentation Group. *PHP Manual*. http://www.php.net/manual/en/
46.*Embperl*. http://perl.apache.org/embperl/
47. *Using Server-Side Directives.* http://www.4guysfromrolla.com/webtech/082599-1.shtml
48. *Python*. http://www.python.org/
49. *Ruby*. http://www.ruby-lang.org/en/
50. E. Bragger. *HTTP Header Repository*. http://headers.bragger.net/
51. Netcraft. *Directory of Web Server Home Sites*. http://www.netcraft.com/Survey/servers.html
52. *Web Servers*. http://www.serverwatch.com/stypes/index.php/V2Vi
53. *Google*. http://www.google.com/search?hl=en&ie=UTF-8&oe=UTF-8&q=inurl%3Aphp+site%3Aee
54. *Neti*. http://www.neti.ee/cgi-bin/otsing?src=web&query=allinanchor%3A+php
55. *Lasso Tutorial*. http://www.lassoassociation.co.uk/las_uklassoc_1wa.LassoApp?-responselassoapp=index.lasso
56. Inline Internet Systems, Inc. *Making Dynamic Websites Affordable*. http://www.ihtml.com/about/
57. *mod_mono*. http://go-mono.com/asp-net.html#mod_mono
58. J. Gilmore. *Creating User-Friendly URLs*. 24.04.2003 http://www.zend.com/zend/trick/tricks-apr-2003-urls.php

Lisa. Laserplaat

Käesolevale tööle lisatud laserplaat (CD) sisaldab antud töö elektroonilist koopiat PDF-formaadis (uuring.pdf) ning töö lisade kataloogi. Lisade kataloogi on paigutatud uuringus kasutatud ja loodud andmefailid ning andmete töötlemiseks kasutatud PHP käsureaskriptid, mille töötamine on testitud järgmisel baastarkvaral: Windows XP, PHP 4.3.1 CLI SAPI, MySQL 3.23.51 ja Nmap 3.3.

Kataloogi Lisad sisu:

- ✔ netiWWWserverid.html – Neti veebiotsingu saidilt pärit Eesti veebiserverite ja saitide nimekiri IP-aadresside alusel grupeerituna (01.04.2004);
- ✔ netiWWWserverid_by_size.txt – Neti veebiotsingu saidilt pärit Eesti veebisaitide nimekiri, sorteerituna indekseeritud veebilehekülgede arvu järgi (01.04.2004);
- ✔ wwwServerid.php – PHP skript, mis valib failist netiWWWserverid.html saitide nimede põhjal antud uuringu jaoks arvesse minevad saidid (sõnega „www.“ algavad saidid);
- ✔ serveriteIP.txt – PHP skripti wwwServerid.php väljund, kus asuvad uuringu jaoks vaatluse alla tulevad Eesti veebisaidid grupeerituna IP-aadresside järgi;
- ✔ laeAndmed.php – uuringu veebirobot, mis kogub veebiserverite ja saitide kohta HTTP-protokolli alusel uuringu jaoks vajalikku infot, lähtudes failist serveriteIP.txt ning kasutades abiprogrammi Nmap. Veaolukorrad saitidel navigeerimisel salvestatakse faili debugLog.txt;
- ✔ andmed.xml – veebiroboti laeAndmed.php väljundfail, mis sisaldab uuringu jaoks vajalikku infot HTTP päistest, saitide sisust (URLid, robots.txt, indeksfailid) ja Nmap väljundist;
- ✔ fingerprints.txt – siin failis on kirjas otsused, mille alusel veebiserveritelt saadud info grupeeritakse uuritud tarkvara (veebiserverid, operatsioonisüsteemid, serveripoolsed programmeerimisvahendid) esindavate märksõnade alla;
- ✔ analyys.php – kirjeldatud otsuste (fingerprints.txt) põhjal andmeid (andmed.xml) analüüsiv skript, mis iga tundmatu olukorra puhul peatab töö ja esitab info, mille põhjal täiendada otsuste faili. Kui kõik andmed on ära kirjeldatud, siis võimaldab skript info salvestamise andmebaasi, kust on võimalikud edasised SQL päringud uuringutulemuste välja võtmiseks;
- ✔ andmebaas.sql – andmeanalüüsi (analyys.php) lõpliku tulemusena saadav MySQL andmebaas;
- ✔ ABp2ringud.txt – andmebaasi (andmebaas.sql) struktuuri ning uuringu tulemusi

esitavate tabelite koostamiseks kasutatud SQL andmebaasipäringud;

- ✔ saitideSuuruseLisamine.php – skript, mis lisab andmeanalüüsi (analyys.php) käigus lõplikult valminud saitide tabelile juurde nende suurused, kasutades faili netiWWWserverid_by_size.txt;
- ✔ google_neti_info.txt – andmed serveripoolsete programmide populaarsuse kohta indekseeritud veebilehekülgede arvu järgi, kasutades veebiotsingu saite Google ja Neti (05.04.2004).