Tallinna Ülikool
Digitehnoloogiate instituut
Informaatika

Linuxis kasutatavate veebiserverite turvalisuse analüüs

Bakalaureusetöö

Autor: Hendrik Spiegelberg
Juhendaja: Edmund Laugasson

Autor: ..... „ ..... „2017
Juhendaja: ..... „ ..... „2017
Instituudi direktor: ..... „ ..... „2017

Tallinn 2017

Autorideklaratsioon

Deklareerin, et käesolev bakalaureusetöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

.....	.....
(kuupäev)	(autor)

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina _____ (sünnikuupäev: _____)
(*autori nimi*)

1. annan Tallinna Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
   _____
   _____
   _____
   (*lõputöö pealkiri*)

   mille juhendaja on _____,
   (*juhendaja nimi*)

   säilitamiseks ja üldsusele kättesaadavaks tegemiseks Tallinna Ülikooli Akadeemilise Raamatukogu repositooriumis.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tallinnas/Haapsalus/Rakveres/Helsingis, _____
(digitaalne) allkiri ja kuupäev

Sisukord

Sissejuhatus

Üle 80% veebisaitidest kasutavad oma lehtede kuvamiseks avatud lähtekoodiga veebiservereid (“Usage of web servers for websites,” 2017). Veebiserverite jooksutamiseks on olemas kaks põhilist operatsioonisüsteemi liigitust – Unix ja Windows. Unix all olevatest operatsioonisüsteemidest on umbes 55% erinevad Linuxi distributsioonid. (“Usage of operating systems for websites,” 2017).

Käesoleva uurimistöö peamistes eesmärkideks on:

1) Teada saada erinevaid Linuxi veebiserverite poolt kasutatavaid turvameetmeid, neid analüüsida ja võrrelda.

2) Uurida, mida on veebiserverite turvamiseks seni tehtud ja mis probleeme on esinenud.

3) Välja selgitada, millist veebitarkvara valitakse süstemmiadministraatorite poolt ja miks.

Uurimuse probleem seisneb selles, et enimkasutatud veebiserverid ei pruugi olla kõige turvalisemad ja sellepärast oleks vaja teada saada ja hiljem edastada info uuringutes osalejatele, kui turvalised on hetkel kasutuses olevad Linuxi veebiserverid. Teema on aktuaalne, sest tänapäeval on veebiserverite turvalisus väga oluline ja enamasti on veebiserverid üles seatud just Linux operatsioonisüsteemis.

Töö teoreetilises osas, esimeses peatükis, on antud ülevaade erinevatest avatud lähtekoodiga Linuxil saadavatest veebiserveritest.

Teises peatükis on käsitletud nende turvalisust. Selle all tuuakse välja peamised veebirünnakute meetodid. Seal on ka juttu sellest, kuidas tagada veebiserverite turvalisus ja ka operatsioonisüsteemide turvalisus, kuhu veebiserver paigaldatud on. Samuti on toodud välja paar näidet sellest, kuidas planeeritakse veebiserverite turvalisust tõsta.

Kolmandas peatükis on viidud virtuaalmasina peal olevate serverite peal erinevad DoS (*Denial of Service)* rünnakud. Katsetatud on, kui kiiresti laeb leht ära erinevaid rünnakuid kasutades. Kasutusele on ka võetud rakendus nende rünnakute peatamiseks.

Neljandas peatükis on lahti seletatud *Google Formsi* küsimustik, mis on veebiserverite turvalisuse teemal ja suunatud süsteemiadministraatoritele ja töö viiendas peatükis on selle vastuseid analüüsitud ning nende põhjal on tehtud uurimuse kokkuvõte.

Tänan oma juhendajat Edmund Laugassonit hea koostöö ja asjalike nõuannate eest.

1 Linuxi erinevad avatud lähtekoodiga veebiserverid

Üle 80% veebisaitidest kasutavad oma lehtede kuvamiseks avatud lähtekoodiga veebiservereid (“Usage of web servers for websites,” 2017). Veebiserverite jooksutamiseks on olemas kaks põhilist operatsioonisüsteemi liigitust – Unix (kõikidest kasutajatest umbes 66%) ja Windows (kõikidest kasutajatest umbes 33%). Unix all olevatest operatsioonisüsteemidest on umbes 55% erinevad Linuxi distributsioonid. (“Usage of operating systems for websites,” 2017) Alljärgnevalt on välja toodud kõige populaarsemate avatud lähtekoodiga Linux veebiserverite kirjeldused.

1.1 Apache HTTP Server

Kõige populaarsem veebiserver *Apache HTTP Server*, mis on tuntud ka kui httpd, aga ka lihtsalt Apache, tuli esmalt välja 1995 aastal. Apache on olnud kõige populaarsem veebiserver alates 1996. aastast. Kõikidest maailma veebisaitidest üle 50% kasutavad Apachet. Apache veebiserverit kasutatakse enim Linuxi masinates aga seda saab kasutada ka OS X ja Windows masinates. Apache kasutab modulaarset arhitektuuri. See tähendab, et serverile saab lisada mooduleid, millega saab serveri omadusi laiendada (Muilwijk, 2016). Kuna Apache on kõige populaarsem veebiserver, siis üheks eeliseks on väga hea dokumentatsioon. Apache serverit kasutatakse, sest see on paindlik, võimas ja laialdaselt toetatud. (Ellingwood, 2015)

1.2 NGINX

NGINX on populaarsuselt teine avatud lähtekoodiga veebiserver, mida kasutavad ligikaudu 33% kõikidest maailma veebisaitidest (“Usage of web servers for websites,” 2017). Nginx üheks eeliseks on selle kiirus. Nginx kasutab asünkroonseid sokleid ja ei tekita päringute peale palju eraldi protsesse. Üks protsess ühel protsessori tuumal saab hakkama tuhandete ühendustega, sellepärast on protsessori ja mälu kasutus Nginx’il palju madalam. Nginx on ka lihtsam kasutada, konfiguratsiooni failid on lihtsamini loetavad ja muudetavad kui näiteks Apaches ja piisab ainult paarist reast, et üles seada terviklik virtuaal *hosti* lahendus. Sarnaselt Apachele on ka Nginx modulaarne (Nedelcu, 2010). Nginx kasutatakse selle efektiivse ressursi kasutuse ja hea koormuse taluvuse pärast. (Ellingwood, 2015)

1.3 Apache Tomcat

Apache Tomcat on populaarsuselt kolmas avatud lähtekoodiga veebiserver. Apache Tomcati kasutavad 0.6% kõikidest maailma veebisaitidest (“Usage of web servers for websites,” 2017). Apache Tomcat erinevad tavalisest HTTP serverist sellepoolest, et see kasutab veebilehe näitamiseks *JavaServer Pages* (JSP) tehnoloogiat. JSP kasutab veebilehe sisu või kujunduse muutmiseks servlettte. Servletid on Java klassid, mida kasutatakse dünaamilise veebisisu näitamiseks (Boadas, 2016). Apache Tomcati kasutatakse selle lihtsuse, paindlikuse, stabiilsuse ja lisa turvalisuse võimaluse poolest. (Davis, 2014)

1.4 Node.js

Node.js on populaarsuselt neljas avatud lähtekoodiga veebiserver. Node.js kasutavad 0.3% kõikidest maailma veebisaitidest (“Usage of web servers for websites,” 2017). Node.js erineb teistest veebiserveritest sellepoolest, et et see kasutab sündmusjuhitavat arhitektuuri, mis tähendab, et see ootab sündmuste toimumist ja reageerib neile, mitte ei tee läbi kindlaksmääratud samme. Samuti on Node.js asünkroonse I/O võimeline. Läbi selliste disainilahenduste on võimalik Node.js serveriga luua reaalaja rakendusi (Muilwijk, 2016). Node.js kasutatakse selle kiiruse, reaalaja rakenduste, programmeerimislihtsuse ja tegusa kommuuni pärast. (Pal, 2016)

1.5 Lighttpd

Lighttpd on populaarsuselt viies avatud lähtekoodiga veebiserver. Lighttpd kasutavad 0.1% kõikidest maailma veebisaitidest (“Usage of web servers for websites,” 2017). Lighttpd on väike ja tõhus veebiserver, mis erineb teistest veebiserveritest väikese mälu kasutuse, efektiivse protsessori töö haldamise ja funktsionaalsuse poolest. Lighttpd on hea lahendus serveritele, millel on probleeme koormusega. (“Lighttpd,” 2017)

2 Linuxi veebiserverite turvalisus

Tänapäeval on veeb muutunud väga populaarseks ja paljud ettevõtted sõltuvad veebist oma igapäevastel tegemistel. Samuti kasutavad miljardid inimesed igapäevaselt veebi. Sellise populaarsuse kasvu tõttu on kasvanud ka ründajate huvi veebi vastu. Eksisteerib palju erinevaid võimalusi, kuidas on võimalik veebi rünnata (peatükis 2.1 on populaarsemad neist välja toodud). Samuti saab ära kasutada ka näiteks veebilehel katkist autoriseerimissüsteemi ja sessiooni haldust. Mida rohkem uusi tehnoloogiad veebis kasutusele võetakse, seda enam rünnakuvariante tekib. (van Goethem, Chen, Nikiforakis, Desmet, & Joosen, 2014)

2.1 Peamised veebirünnakute meetodid

Peamised veebirünnakute meetodid on: *Cross-Site Scripting* (XSS), *Cross-Site Request Forgery* (CSRF), *Distributed Denial/Denial of Service* (DDOS/DOS) ja q*SQL Injection.*

2.1.1 *Cross-site scripting* (XSS)

Cross-site scripting (tuntud ka kui XSS või CSS) on rünnak, mis juhtub kui dünaamiliselt genereeritud veebilehtedel kuvatakse sisendvälja, mis ei ole õigesti valideeritud. Seeläbi saab ründaja lehele lisada *JavaScripti* koodi ja seda käivitada veebilehe külastaja arvutis. Selline rünnak võib ohustada igat lehekülge, kus on kasutajal võimalik sisestada andmeid. Eduka rünnaku korral võib ründaja kätte saada salastatud informatsiooni, manipuleerida või varastada küpsiseid, luua päringuid, mis tunduvad nagu need tuleksid päris kasutajalt või käivitada pahatahtlikku koodi kasutaja süsteemis. (Spett, 2005)

2.1.2 *Cross-site request forgery* (CSRF)

Cross-site request forgery on ründemeetod, mille kaudu ründaja saab petta kasutajat mingit tegevust tegema, kasutades selleks kasutaja õigusi ja andmeid. CSRF rünnak esitab *HTTP* päringuid kasutaja veebibrauseri kaudu haavatavale veebirakendusele. Kui kasutaja on veebirakendusse sisselogitud, siis brauser saadab automaatselt autentimisinfo, mis sisaldub küpsises, koos päringuga. Päring täidetakse

veebirakenduse poolt, sest veebirakendus ei saa aru, et päring on võltsitud. (Käfer, 2008)

2.1.3 *Distributed denial/Denial of service* (DDoS/DoS)

Nii *Distributed denial of service* (DDoS), kui ka *Denial of service* (DoS) rünnaku mõtteks on koormata üle veebiserveri ribalaius ja muud ressursid. Seeläbi muutub veebiserver teistele kasutajatele kättesaamatuks. DoS rünnaku puhul kasutatakse ühte arvutit ja ühte internetiühendust selleks, et koormata veebiserverit *TCP/UDP* pakettidega. DDoS rünnak on sarnane DoS rünnakule aga selle tulemus on väga erinev. Ühe arvuti ja internetiühenduse asemel kasutatakse mitmeid arvuteid ja ühendusi. Sellise rünnaku puhul kasutatavad arvutid on tavaliselt mööda maailma laiali ja on osa niinimetatud *botnetist*. Kuna veebiserverit koormavad ühe ühenduse asemel üle sajad või isegi tuhanded päringud, siis on sellise rünnakuga palju raskem toime tulla kui tavalise DoS rünnakuga. (Munson, 2009)

2.1.4 *SQL injcetion*

Kasutades *SQL injection* rünnakut, saab ründaja muuta dünaamiliselt genereeritud päringu eeldatavat tulemust veebirakenduses. Sellega võib ründaja saada autoriseerimata ligipääsu veebirakenduse andmebaasile ja teha andmebaasis sisalduva informatsiooniga erinevaid toiminguid. Tehtavad toimingud varieeruvad alustades lihtsalt informatsiooni pärimisega või koodi käivitamisega kuni terve süsteemi ohtu seadmiseni välja. (Umar, Bakar Md Sultan, Zulzalil, Admodisastro, & Taufik Abdullah, 2016)

2.2 Veebiserveri turvalisuse tagamine

Veebiserverid on juba oma disaini poolest turvalisena loodud. Sellest ei ole aga kasu, kui näiteks operatsioonisüsteem kuhu veebiserver paigaldatakse, on valesti seadistatud. Täpselt sama tähtis on ka turvaline veebiserveri konfiguratsioon.

2.2.1 Operatsioonisüsteem

Operatsioonisüsteemi paigaldamisel ei ole vaikimisi kasutusel olev seadistus väga turvaline. Jälgida tuleb, et paigaldamisel ei tuleks kaasa võrguteenuseid, mida ei ole vaja. Mida rohkem selliseid teenuseid töötab, seda rohkem porte on avatud, mis

tähendab, et ründajal on rohkem võimalusi rünnaku läbiviimiseks. *Nmap* on hea rakendus, millega kontrollida, millised pordid avatud on. Seda saab teha käsuga *nmap localhost.*

```
Starting Nmap 7.12 ( https://nmap.org ) at 2017-05-02 13:06 EEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00010s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
3306/tcp open  mysql
```

Joonis 1 Nmap poolt kuvatav informatsioon

Kõik kasutajakontod, mis tulevad operatsioonisüsteemi paigaldamisega kaasa ja ei ole kasutusel tuleks inaktiveerida. Igal süsteemiadministraatoril peaks olema oma isiklik kasutaja ja ei tohiks kasutada vaikimisi loodavat kasutajat, kes on superkasutaja õigustes ega *root* kasutajat.

Hea oleks kui veebiserverile logiks administraator sisse kohalikult aga see pole tänapäeval kõige praktilisem variant. Selleks et veebiserverile kaugemalt ligipääseda tuleks teha kindlaks, et selline ühendus on turvaline, kasutades selleks ühenduse loomisel tunnelit ja krüpteerimist. Parooliga sisse logimise asemel tuleks kasutada sertifikaate. Kaugpääs serverile peaks olema piiratud ainult mõnele kindlale kasutajale. Serverile ligipääsemiseks ei tohiks kasutada ka avalikke arvuteid.

Kui muu tarkvara seda lubab ja konflikte ei teki, siis tuleks operatsioonisüsteemile ja tarkvarale saadaval olevad turvalisuse uuendused ära teha esimesel võimalusel, sest paljud rünnakud on suunatud just veebiserveritele, millel ei ole uuendusi tehtud. (“Keeping Web and Database Servers Secure,” 2017)

2.2.2 Veebiserver

Peale veebiserveri tuleks ära muuta selle konfiguratsioon, sest paljud vaikimisi seaded, ei ole turvalised ja võivad tagada veebiserverile liiga palju ligipääsu.

Ära tuleks keelata *Server Signature* ja muu taolise süsteemi informatsiooni kuvamine, sest selline informatsioon aitab ründajal leida täpselt selle versiooni turvaauke. Seda informatsiooni saab vaadata, kasutades Linuxi käsureal käsku „*curl -I http://serveri.aadress/“*. Veebiserveri informatsiooni kuvamist saab välja lülitada

veebiserveri konfiguratsioonifaili kaudu. Apache puhul tuleb muuta */etc/apache2/apache2.conf* faili. Selle faili lõppu tuleb lisada juurde 2 rida:

1) *ServerTokens ProductOnly*. Kui seda väärtust pole eraldi määratud, siis on see vaikimisi *ServerTokens Full*, mis näitab ära nii veebiserveri versiooni, operatsioonisüsteemi tüübi ja kasutuselolevad moodulid. Teised võimalikud *ServerTokens* valikud on: *Major*, *Minor*, *Minimal* ja *OS*.

```
HTTP/1.1 200 OK
Date: Tue, 02 May 2017 10:09:53 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Sun, 23 Apr 2017 18:17:36 GMT
ETag: "beb-54dd9814b7000"
Accept-Ranges: bytes
Content-Length: 3051
Vary: Accept-Encoding
Content-Type: text/html
```

Joonis 2 Vaikimisi kuvatav informatsioon

```
HTTP/1.1 200 OK
Date: Tue, 02 May 2017 10:07:59 GMT
Server: Apache
Last-Modified: Sun, 23 Apr 2017 18:17:36 GMT
ETag: "beb-54dd9814b7000"
Accept-Ranges: bytes
Content-Length: 3051
Vary: Accept-Encoding
Content-Type: text/html
```

Joonis 3 Informatsioon peale ServerTokens ProductOnly muudatust

2) *TraceEnable Off*. Vaikimisi on see sees ja *TRACE* päring (*curl -v -X TRACE http://serveri.aadress/*) annab vastueks *200 OK*. Välja lülitades ei saa teha enam *TRACE* päringuid, annavad veateate: *405 Method Not Allowed.*

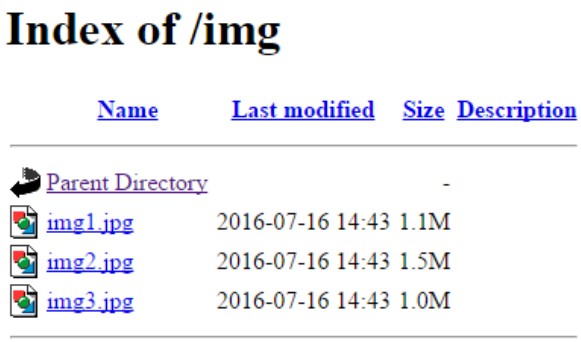
```
HTTP/1.1 200 OK
Date: Tue, 02 May 2017 10:11:00 GMT
Server: Apache/2.4.18 (Ubuntu)
Transfer-Encoding: chunked
Content-Type: message/http
```

Joonis 4 Vaikimisi kuvatav TraceEnable informatsioon

```
HTTP/1.1 405 Method Not Allowed
Date: Tue, 02 May 2017 10:12:33 GMT
Server: Apache/2.4.18 (Ubuntu)
Allow:
Content-Length: 298
Content-Type: text/html; charset=iso-8859-1
```

Joonis 5 Informatsioon peale TraceEnable välja lülitamist

Directory listing laseb *index.html* faili puudumisel vaadata serveris olevaid faile, mis võib anda ligipääsu failidele, mida ei tohiks muidu kuvada. Nginx puhul on see vaikimisi välja lülitatud.



Joonis 6 Serveri /img kaustas olevad pildid

Apache puhul saab seda inaktiveerida läbi konfiguratsioonifaili. Konfiguratsioonifailis tuleb otsida üles koht, kus on ära kirjeldatud serveri poolt serveeritavad kaustad ja sealt soovitud kaustal eemaldada *Indexes* väärtus.

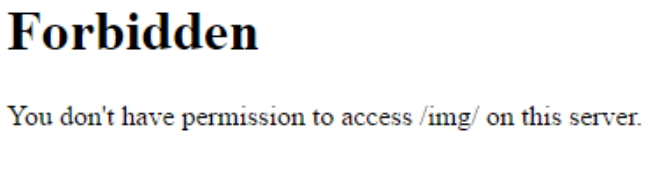
```
<Directory /var/www/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>
```

Joonis 7 Kaustade seaded koos Indexes väärtusega

```
<Directory /var/www/>
        Options FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>
```

Joonis 8 Kaustade seaded pärast Indexes väärtuse eemaldamist

Peale selle inaktiveerimist kuvatakse alamkataloogi liikudes teade, et selle kausta kuvamiseks ei ole piisavalt õigusi.



Joonis 9 Peale DirectoryListing inaktiveerimist kuvatav teade

Kui kasutada veebiserverit, mis kasutab erinevaid mooduleid, siis tuleks mittevajalikud moodulid inaktiveerida. Apache puhul saab kasutusel olevaid mooduleid näha käsuga: *apachectl -M*. Mooduleid saab inaktiveerida käsuga *sudo a2dismod moodulinimi* ja aktiveerida käsuga *süda a2enmod moodulinimi*. Peale muudatuste tegemist tuleb alati teha serverile restart käsuga *sudo service apache2 restart*. Samuti tuleks võimalusel kasutada veebiserveri kõige uuemat versiooni. (Kumar, 2015)

2.3 Plaanid veebiserverite turvalisuse tõstmiseks

Veebiserverite turvalisus on tänapäeval väga tähtis. Turvalisuse dokumenteerimisega tegelev organisatsioon OWASP aitab oma informatsiooniga edendada veebi turvalisust. Luuakse uusi lahendusi selleks, et veebiturvalisust tõsta. Üheks selliseks lahenduseks on Apache poolt loodav veebiturvalisuse raamistik Milagro.

2.3.1 OWASP

Open Web Application Security Project (OWASP) on ülemaailmne mittetulundusorganisatsioon, mis tegeleb veebitarkvara turvalisuse edendamisega. Nende eesmärgiks on teha turvalisus kättesaadavaks inimestele ja organisatsioonidele. OWASP pakub erinevaid tasuta tööriistu, dokumente ja foorumeid kõikidele, kes on huvitatud turvalisuse edendamisest. Näiteks saab seal lehel informatsiooni serveri konfigureerimise kohta. Samuti on seal väga täpselt välja toodud võimalike rünnakute kirjeldused. (“OWASP,” 2017)

2.3.2 Apache Milagro

Apache Milagro on uus veebiturvalisuse raamistik, mis on hetkel arendusfaasis. Milagrot arendatakse *MIRACL* ja *Nippon Telegram and Telegraph (NTT)* koostööna. *Milagrol* puudub avaliku võtme infrastruktuur (*public key infrastructure PKI)* ja seetõttu ei ole kasutusel ühtegi parooli. Sellepärast ei ole vaja autentida ennast läbi miljonite veebiserverite, sellise vajaduse puudumisel väheneb võimalusi pettusi läbi viia. See süsteem laseb valida omale “usalduse” pakkujaid, iga selline pakkuja hoiab enda käes osa sinu põhivõtmest. Kolm asja, mida *Milagro* platvorm teeb, mida pole kasutuses üheski teises platvormis:

1. Puudub risk, et keegi ründab ja pääseb ligi paroolide andmebaasile, sest paroole ei ole vaja kasutada.

2. Muudab kasutajatele autentimise lihtsamaks. Kasutajad vajavad ainult nelja kohalist PIN koodi, mis on 128-bitise turvalisusega.

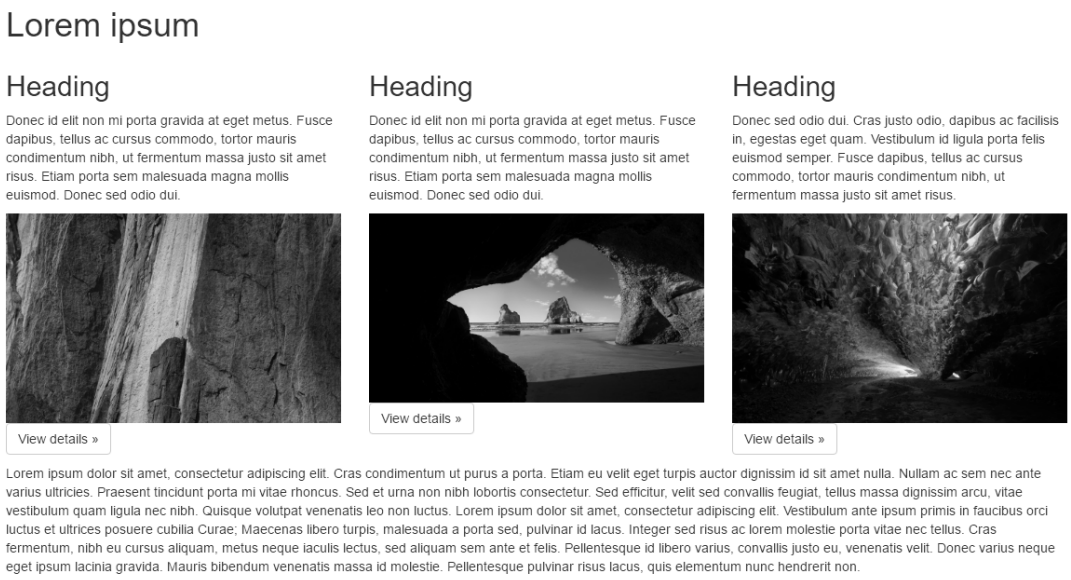
3. On võimalik kasutada erinevaid autentimismeetodeid, näiteks biomeetria, geolokatsioon, riistvara jne.

Milargo on küll varajases faasis aga on juba võetud kasutusele Briti maksuametis ja krediidiagentuuris Experian. (Murphy, 2016)

3 Rünnakute läbiviimine virtuaalmasinates

Viin läbi erinevaid DoS (*Denial of Service*) rünnakuid virtuaalmasinas olevate veebiserverite peal. Rünnakute läbiviimiseks kasutan rakendusi LOIC (*Low Orbit Ion Cannon)* ja *OWASP HTTP Post Tool (OWASP Switchblade 4.0)*. Proovin rünnakud läbi Apache ja Nginx veebiserverite peal, mis on paigaldatud Ubuntu Server 16.10 operatsioonisüsteemile. Virtuaalmasinale on antud 1024 megabaiti operatiivmälu ja üks protsessorituum.

Mõlemad serverid kasutavad sama HTML faili, kus on kasutusel ka *Twitter Bootstrap.* Leht koosneb pealkirjast, kolmest tulbast, kus on üks paragrahv teksti ja pilt ning lõpus veel üks tekstiparagrahv. Lehel kasutusel olevad pildid on kõik mahu poolest umbes 1 megabait.



Joonis 10 Kuvatõmmis veebilehest

Katsetan mõlema serveriga nelja erinevat stsenaariumi. Esimeseks on lehe laadimine tavaliselt, ilma ründamata. Teiseks kasutan LOIC rakendust ja määran rünnakumeetodiks HTTP ja *threadide* arvuks 10. Kolmandaks kasutan *OWASP Switchblade* rakendust ja määran ühenduste arvuks 100. Nelja testi puhul kasutan samuti *OWASP Switchblade* rakendust aga ühenduste arvuks määran 150. Igat stsenaariumi proovin läbi kümme korda, selleks, et keskmine veebilehe laadimiseaeg määrata. Katsetuste käigus lehte uuesti laadides kasutan *Google Chrome* lehe värskendamisevalikut *Empty Cache and Hard Reload*, selleks, et kogu leht laetaks

täielikult uuesti. Lehe laadimiseaja mõõtmiseks kasutan *Google Chrome* laiendust „*Page load time“.* Peale rünnakute läbiviimist näitan, kuidas sellist tüüpi rünnakuid peatada, kasutades selleks rakendust Fail2ban.

3.1 Apache

Kasutusel on vaikimisi paigaldatav Apache versioon, mis on 2.4.18. Apache peal lehte normaalselt laadides jääb lehe laadimiseaeg 1,25 sekundi kanti. Laadides lehte siis, kui on kasutusel LOIC kaudu tehtav DoS rünnak, siis veebilehe laadimiseaeg on umbes 3,49 sekundit ja veebileht on kasutatav. Kasutades *OWASP Switchblade* rünnakut, kus ühenduste arvuks on 100, siis võtab lehe laadimine aega umbes 3,62 sekundit ja on kasutatav. Kui kasutada ründamiseks sama rakendust aga ühenduste arvuks määrata 150, siis ei lae leht enam üldse ära.

3.2 Nginx

Testimise käigus on kasutusel vaikimisi paigaldatav Nginx, mille versioon on 1.10.1. Lehte normaalselt, ilma ründamata, laadides on veebilehe laadimiseaeg umbes 1,5 sekundit. Laadides lehte LOIC rünnaku ajal on veebilehe laadimiseaeg enamasti umbes 4 sekundit, aga väga tihti annab server veateate – *500 Internal Server Error,* mis lehe kasutatavust oluliselt mõjutab. Lehe laadimine *OWASP Switchblade* rünnaku ajal, kui ühenduste arvuks on määratud 100, võtab aega umbes 3,1 sekundit ja leht on kasutatav. Määrates ühenduste arvuks 150, tõuseb lehekülje laadimiseaeg umbes 3,79 sekundini.

3.3 Fail2ban

Fail2ban on rakendus, mida saab panna jälgima erinevate rakenduste logifailide tegevust ja mis tuvastab seal kahtlase tegevuse. Peale kahtlase tegevuse tuvastamist saab rakendus uuendada operatsioonisüsteemi tulemüüri reegleid ja keelata teatud ajaks ära IP aadress, millelt kahtlased päringud tulevad. Seda saab DoS rünnaku puhul panna jälgima veebiserveri *access.log* faili.

Rakendust saab paigaldada Ubuntu operatsioonisüsteemi käsuga *sudo apt-get install fail2ban.* Peale rakenduse paigaldamist tuleb luua uus reegel, mis jälgiks kasutusel oleva veebiserveri logifaili. Need reeglid asuvad failis */etc/fail2ban/jail.conf* ja uue

reegli saab lisada sinna kõige lõppu. Logpath on Nginx puhul */var/log/nginx/access.log*. Järgneva reegliga on ära määratud, kui 60 (*findtime)* sekundi jooksul tuleb 200 *(maxretry) GET* päringut, siis IP aadress lisatakse 600 *(bantime)* sekundiks tulemüüri.

```
[http-get-dos]
enabled = true
port = http, https
filter = http-get-dos
logpath = /var/log/apache2/access.log
maxretry = 200
findtime = 60
bantime = 600
action = iptables[name=HTTP, port=http, protocol=tcp]
```

Joonis 11 DoS vastane reegel jail.conf failis

Peale selle reegli lisamist tuleb luua ka filter, mida see reegel kasutab. Filter kontrollib regulaaravaldisega, millised tegevused logifailis lubatakse läbi ja millised mitte. DoS rünnaku puhul saab sinna lisada regulaaravaldise, mis kontrollib *GET* päringuid. Filtri faili loomiseks ja muutmiseks saab kasutada käsku *sudo nano /etc/fail2ban/filter.d/http-get-dos.conf.*

```
[Definition]
failregex = ^<HOST> -.*GET.*/
ignoreregex =
```

Joonis 12 http-get-dos.conf faili sisu

Peale failide muutmist tuleb Fail2ban taaskäivitada käsuga *sudo service fail2ban restart.* Kui peale Fail2ban seadistamist käivitada näiteks LOIC rünnak, siis saab Fail2ban tegemisi jälgida selle logifaili *(/var/log/fail2ban.log)* abil, kasutades käsklust *tail -f /var/log/fail2ban.log.*

```
2017-05-02 13:18:06,896 fail2ban.filter         [2502]: INFO    [http-get-dos] Found 192.168.0.6
2017-05-02 13:18:06,964 fail2ban.actions        [2502]: NOTICE  [http-get-dos] Ban 192.168.0.6
```

Joonis 13 fail2ban.log faili sisu

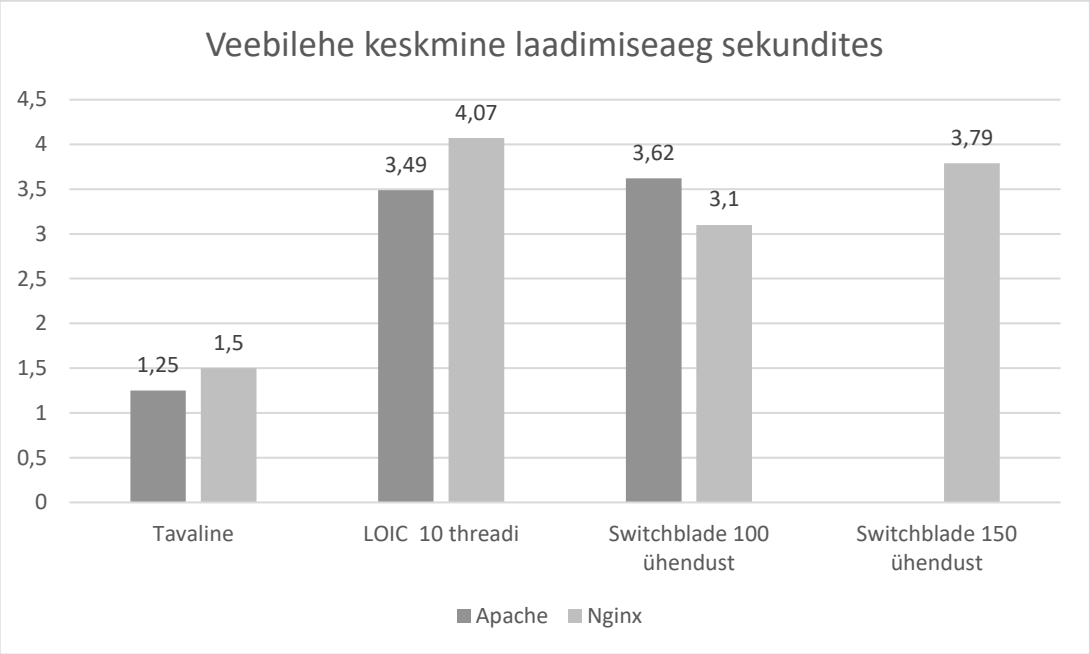
Peale IP aadressi blokeerimist saab vaadata blokeeritud IP aadresse tulemüüri kaudu, kasutades selleks käsku *sudo iptables -L.*

```
Chain f2b-HTTP (1 references)
target     prot opt source               destination
REJECT     all  --  192.168.0.6          anywhere             reject-with icmp-port-unreachable
RETURN     all  --  anywhere             anywhere
```

Joonis 14 Tulemüüri poolt kuvatav informatsioon

3.4 Kokkuvõte

Apache osutus natukene kiiremaks tavatingimustes, aga eriti just LOIC rünnaku ajal, sest leht võttis küll laadimiseks kauem aega aga laadis alati lehe lõpuks ära. Nginx oli kiirem *OWASP Switchblade* rünnaku ajal, eriti just 150 ühendusega rünnaku puhul, sest selle peale lakkas Apache töötamast. Kasutades Fail2ban rakendust on nii LOIC kui ka *OWASP Switchblade* rünnak ebaefektiivne, sest kohe kui päringud ületavad lubatud koguse, siis ei luba tulemüür enam neil uusi ühendusi tekitada.



Joonis 15 Veebilehe keskmine laadimisaeg sekundites

4 Küsitluse metoodika ja valimi kirjeldus

Küsitluse üheks eesmärgiks oli teada saada, kuidas erinevad süsteemiadministraatorid veebiserveri turvalisust mõistavad ja kuidas nemad sellega kokku on puutunud. Teiseks eesmärgiks oli teada saada, millist veebiserverite tarkvara ja operatsioonisüsteeme süsteemiadministraatorid kasutavad. Püstitatud eesmärkide saavutamiseks koostati *Google Formsi* küsimustik (LISA 1).

4.1 Küsitluse metoodika

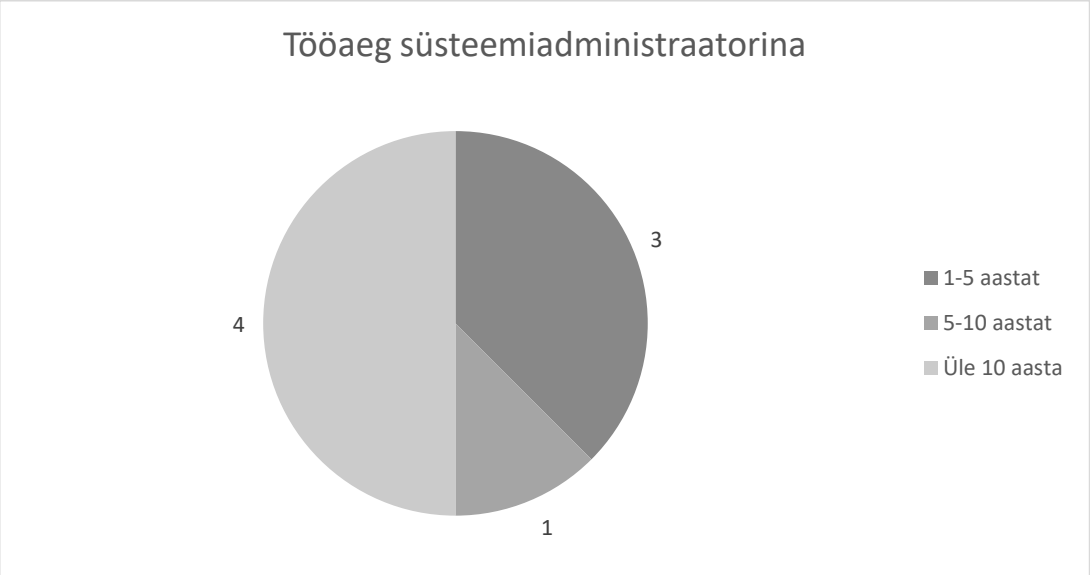
Küsitluses olevad küsimused jagusid kaheks. Esimese poole küsimused on selleks, et teada saada üldist informatsiooni: sektor, kus vastaja töötab ning veebiserverid ja operatsioonisüsteemid, mida ta kasutab. Teises pooles olevad küsimused on seotud veebiserverite turvalisusega. Küsitud on, kas kasutusel olev veebiserveri tarkvara on kõige uuem versioon ja kui ei ole, siis mis on selle põhjuseks. Samuti on ka küsitud kas veebiserverit on rünnatud ja millist tüüpi rünnakuga on olnud tegu. Lõpetuseks on palutud hinnata oma veebiserveri turvalisust viie punkti süsteemis ja kirjutada kolm kõige tähtsamat asja, mida veebiserveri turvalisuse tagamiseks tegema peaks.

4.2 Küsitluse valim

Küsitlus saadeti välja aprillikuus ja selle valimiks oli 25 erinevates sektorites töötavat süsteemiadministraatorit, kes olid välja valitud sihipärasuse alusel. Sektorid jaotusid kolmeks: erasektor, avalik sektor ja kolmas sektor. Igast sektorist oli valimis inimesi võrdselt. Küsimustikule vastanuid oli kokku 8, kellest pooled olid erasektorist ja pooled avalikust sektorist. Kolmandast sektorist vastanuid ei olnud.

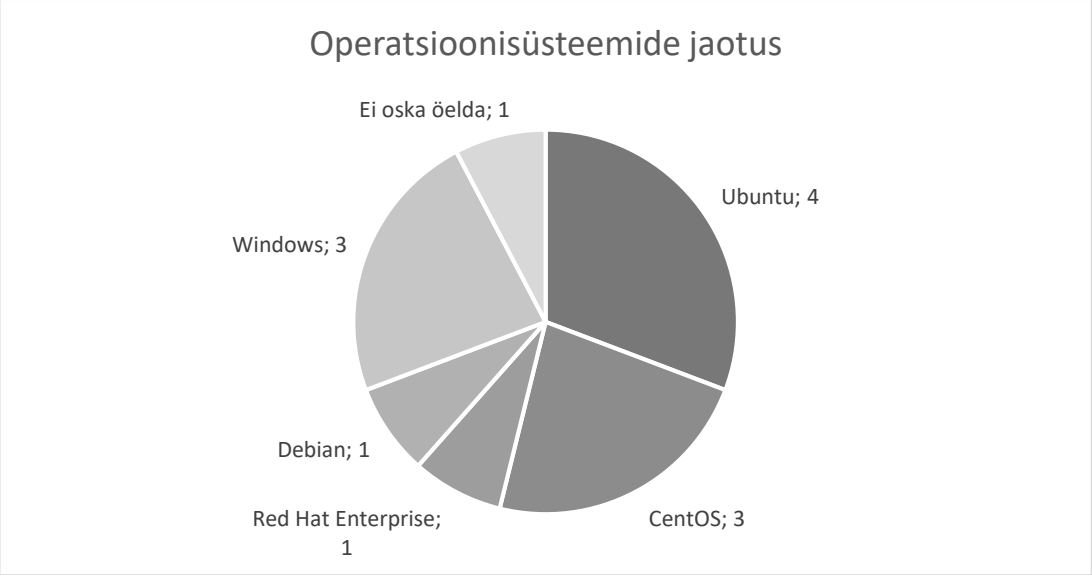
5 Küsitluse analüüs

Esmalt uuriti süsteemiadministraatoritelt, et kui kaua on nad süsteemiadministraatorina töötanud. Vastustest tuli välja, et kaheksast süsteemiadministraatorist neli on töötanud üle kümne aasta, kolm 1-5 aastat ja üks 5-10 aastat.



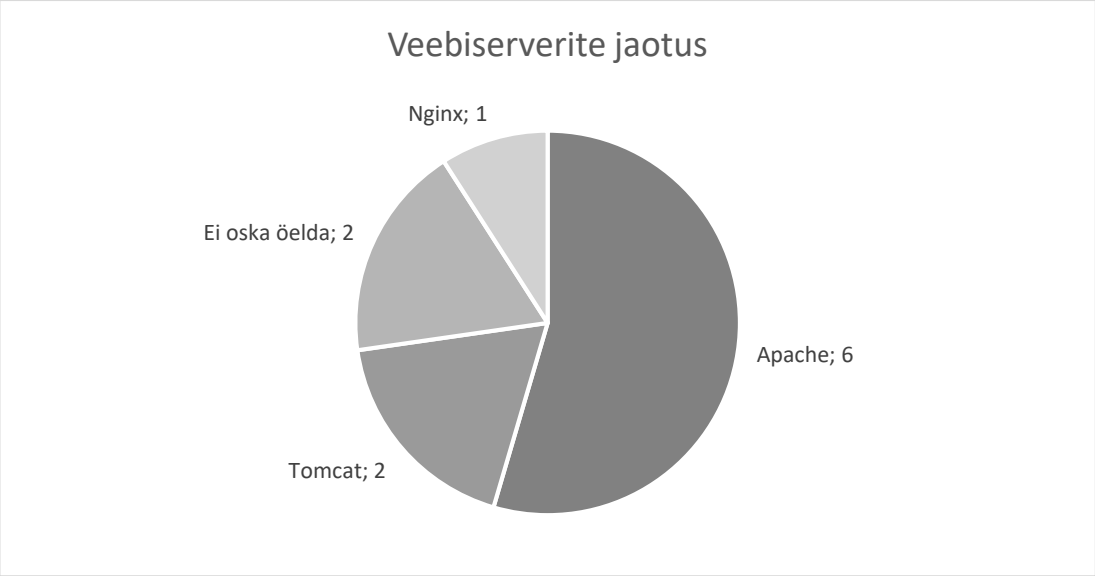
Joonis 16 Tööaeg süsteemiadministraatorina

Järgmiseks küsisin operatsioonisüsteemide kasutuse kohta. Vastustest selgus, et operatsioonisüsteemid jaotusid järgmiselt. Kõige populaarsemaks osutus Linuxi operatsioonisüsteem Ubuntu, mida oli neli. Sellele järgnesid Linuxi operatsioonisüsteem CentOS ja Windows, mõlemat oli kolm. Linuxi operatsioonisüsteemida olid esindatud veel Debian ja Red Hat Enterprise, mõlemal üks. Üks vastanu ei osanud öelda, millist operatsioonisüsteemi ta kasutab.



Joonis 17 Operatsioonisüsteemide jaotus

Järgmine küsimus oli veebiserverite kasutuse kohta. Vastustest selgus, et kõige populaarsemaks osutus Apache veebiserver. Sellele järgnesid Apache Tomcat, mida oli 2 ja Nginx, mida oli 1. Kaks inimest ei osanud öelda, millise veebiserveriga on tegu ja üks täpsustas seda vastates, et nende kool on Tallinna Haridusameti võrgus.



Joonis 18 Veebiserverite jaotus

Selle peale, et kas veebiserveri versioon on kõige uuem, vastas jaatavalt ainult üks inimene ja kolm ei osanud öelda. Need, kes vastasid eitavalt tõid põhjenduseks kas selle, et neil pole piisavalt ressursse uuendamiseks, veebiserver on seotud mingi muu

tarkvaraga, mida ei saa uuendada ja ka seda, et kõige uuemad versioonid ei ole enamasti kõige stabiilsemad.

Küsides, kas nende veebiserverit on rünnatud, vastasid viis inimest eitavalt. Kolm inimest vastasid, et nende veebiserverit on rünnatud. Kaks inimest, kes vastasid, et nende veebiserverit on rünnatud, tõid lisaks välja, et rünnatud on igat võimalikku moodi. Üks inimene täpsustas, et ründamiseks on ära kasutatud CMS-ide (sisuhaldussüsteemide) turvaauke.

Paludes hinnata oma veebiserveri turvalisust viiepunkti skaalal, jagunesid vastused järgnevalt. Kaks vastanut hindasid, et nende veebiserver on väga turvaline ja andsid omale viis punkti. Kolm vastanut hindasid oma veebiserveri turvalisust neljale punktile ja kolm vastanut kolmele punktile.

Viimase asjana küsisin, et mis oleks kolm kõige tähtsamat asja, mida veebiserveri turvalisuse tagamiseks tegema peaks. Kõik vastanud tõid välja, et üheks asjaks oleks kindlasti tarkvara õigeaegne uuendamine. Täpsemalt toodi välja, et tuleks lisaks veebiserveri tarkvarale tuleks uuendada ka kogu ülejäänud süsteem, näiteks operatsioonisüsteem, rakendused, vahekihid/teegid/interpretaatorid ja ka kõik muud serveris jooksvad teenused, näiteks FTP ja SSH. Kolm inimest tõid välja, et kasutajate õigused ja ligipääs serverile peaks olema hästi läbi mõeldud, näiteks peaks olema ligipääs veebiserverile ainult konkreetsetele isikutele, kasutajatel peaks olema täpselt nii palju õigusi, kui on vaja ja kirjutamisõigus peaks kasutajal olema vaid väga kontrollitud keskkonnas. Kaks vastanut tõid välja, et tähtis on ka läbimõeldud ja regulaarne varundamine koos kontrollitud taastamisprotseduuriga. Kaks vastanut tõid välja ka selle, et veebiserveris jooksvad rakendused peaksid olema hästi turvatud. See tähendab seda, et veebirakenduse kood peaks olema õige ja kvaliteetne. Lisaks toodi välja, et veebiserveril peaks olema turvaline ja pädev seadistus, põhjendades seda sellega, et administraator peab aru saama, mida ta teeb ja mitte lihtsalt järgima õpetusi internetist. Veel toodi välja, et turvaauke peaks ennetavalt lappima, veebiserveri majutus peaks hea olema ja ei tohiks jooksutada kaheldava väärtusega sisu.

Kokkuvõte

Käesolevas töös on antud ülevaade erinevatest Linuxil kasutatavatest veebiserveritest, veebiserverite turvalisusest, võimalikest rünnakumeetoditest ja plaanidest, millega planeeritakse veebiserverite turvalisust tõsta.

Esimeses peatükis on välja toodud ja kirjeldatud erinevaid avatud lähtekoodiga veebiservereid Linuxi operatsioonisüsteemile. Täpsemalt on ära kirjeldatud kui populaarsed need tänapäeval on, kuidas need täpsemalt toimivad ja miks neid kasutatakse.

Teises peatükis on juttu veebiserverite turvalisusest. Seal on välja toodud erinevaid kasutusel olevaid rünnakumeetodeid. Lisaks on seal selgitatud, kuidas peaks veebiservereid ja operatsioonisüsteeme seadistama, et need oleksid võimalikult turvalised. Teise osa lõpetuseks on toodud välja paar võimalust, kuidas tahetakse veebiserverite turvalisust tulevikus tõsta.

Kolmandas peatükis on virtuaalmasinas olevate veebiserverite peal DoS (*Denial of Service)* rünnakud, analüüsitud, kuidas veebilehte nende rünnakute all käitub, selgitatud ja kuidas kasutada rakendust Fail2ban nende rünnakute peatamiseks.

Neljandas peatükis on selgitatud läbi viidud *Google Formsi* küsimustiku metoodikad ja valim. Viiendas osas on selle küsimustikku tulemusi analüüsitud ja tehtud kokkuvõte. Tulemustest selgus, et uurimuses osalenud Eesti süsteemiadministraatorite kasutatavate veebiserverite valik ühtib enamasti veebist leitava statistikaga. Samuti saadi teada, milliseid rünnakuid on nende veebiserverite vastu kasutatud. Lisaks on tehtud kokkuvõte soovitustest, kuidas veebiserveri turvalisust tõsta.

Kasutatud kirjandus

Boadas, J. (2016). Difference Between Apache Web Server and Tomcat. Retrieved March 20, 2017, from https://examples.javacodegeeks.com/enterprise-java/tomcat/difference-apache-web-server-tomcat/

Davis, M. (2014). Five Reasons You Should Use Tomcat. Retrieved March 20, 2017, from https://www.futurehosting.com/blog/five-reasons-you-should-use-tomcat/

Ellingwood, J. (2015). Apache vs Nginx: Practical Considerations. Retrieved March 20, 2017, from https://www.digitalocean.com/community/tutorials/apache-vs-nginx-practical-considerations

Keeping Web and Database Servers Secure. (2017). Retrieved April 5, 2017, from https://www.acunetix.com/websitesecurity/webserver-security/

Kumar, C. (2015). 10 Best Practices To Secure and Harden Your Apache Web Server. Retrieved April 12, 2017, from https://geekflare.com/10-best-practices-to-secure-and-harden-your-apache-web-server/

Käfer, K. (2008). Cross Site Request Forgery. Retrieved from https://i.kkaefer.com/csrf-paper.pdf

Lighttpd. (2017). Retrieved March 21, 2017, from https://www.lighttpd.net/

Muilwijk, R. (2016, August). Top 5 open source web servers. Retrieved January 26, 2017, from https://opensource.com/business/16/8/top-5-open-source-web-servers

Munson, L. (2009). DoS vs DDoS – What Is The Difference? Retrieved March 22, 2017, from http://www.security-faqs.com/dos-vs-ddos-what-is-the-difference.html

Murphy, I. (2016). Apache Milagro: A New Security System for the Future of the Web. Retrieved March 28, 2017, from https://www.linux.com/news/apache-milagro-new-security-system-future-web

Nedelcu, C. (2010). *Nginx HTTP Server*. Packt Pub. Retrieved from

https://books.google.ee/books?id=wRIWM-6rqBIC&dq=NGINX&lr=&source=gbs_navlinks_s

OWASP. (2017). Retrieved April 12, 2017, from https://www.owasp.org/index.php/Main_Page

Pal, P. (2016). 12 Benefits of Using Node.js For Web Applications. Retrieved March 21, 2017, from https://think360studio.com/12-benefits-of-using-node-js-for-web-application/

Spett, K. (2005). Cross-site scripting. *SPI Labs*, (1), 1–20. https://doi.org/10.1016/B978-1-59749-543-1.00001-3

Umar, K., Bakar Md Sultan, A., Zulzalil, H., Admodisastro, N., & Taufik Abdullah, M. (2016). SQL Injection Attack Roadmap and Fusion. *Indian Journal of Science and Technology*, *9*(28). https://doi.org/10.17485/ijst/2016/v9i28/97810

Usage of operating systems for websites. (2017). Retrieved March 8, 2017, from https://w3techs.com/technologies/overview/operating_system/all

Usage of web servers for websites. (2017). Retrieved March 8, 2017, from https://w3techs.com/technologies/overview/web_server/all

van Goethem, T., Chen, P., Nikiforakis, N., Desmet, L., & Joosen, W. (2014). Large-Scale Security Analysis of the Web: Challenges and Findings (pp. 110–126). Springer, Cham. https://doi.org/10.1007/978-3-319-08593-7_8

Summary

Title: Security Analysis of Web Servers Used in Linux

The purpose of this Bachelors Thesis was to give an overview of the security of different types of web servers that are used in Linux.

The aims of this thesis were:

1) To find out, analyse and compare the different types of security measures that are used by Linux web servers.
2) Research what has been done to improve the security of web servers and what is being planned.
3) Find out what type of web software is used by system administrators and why.

The first chapter gives an overview on different open-source web servers for Linux, how populaar they are, how do they work and how populaar they are.

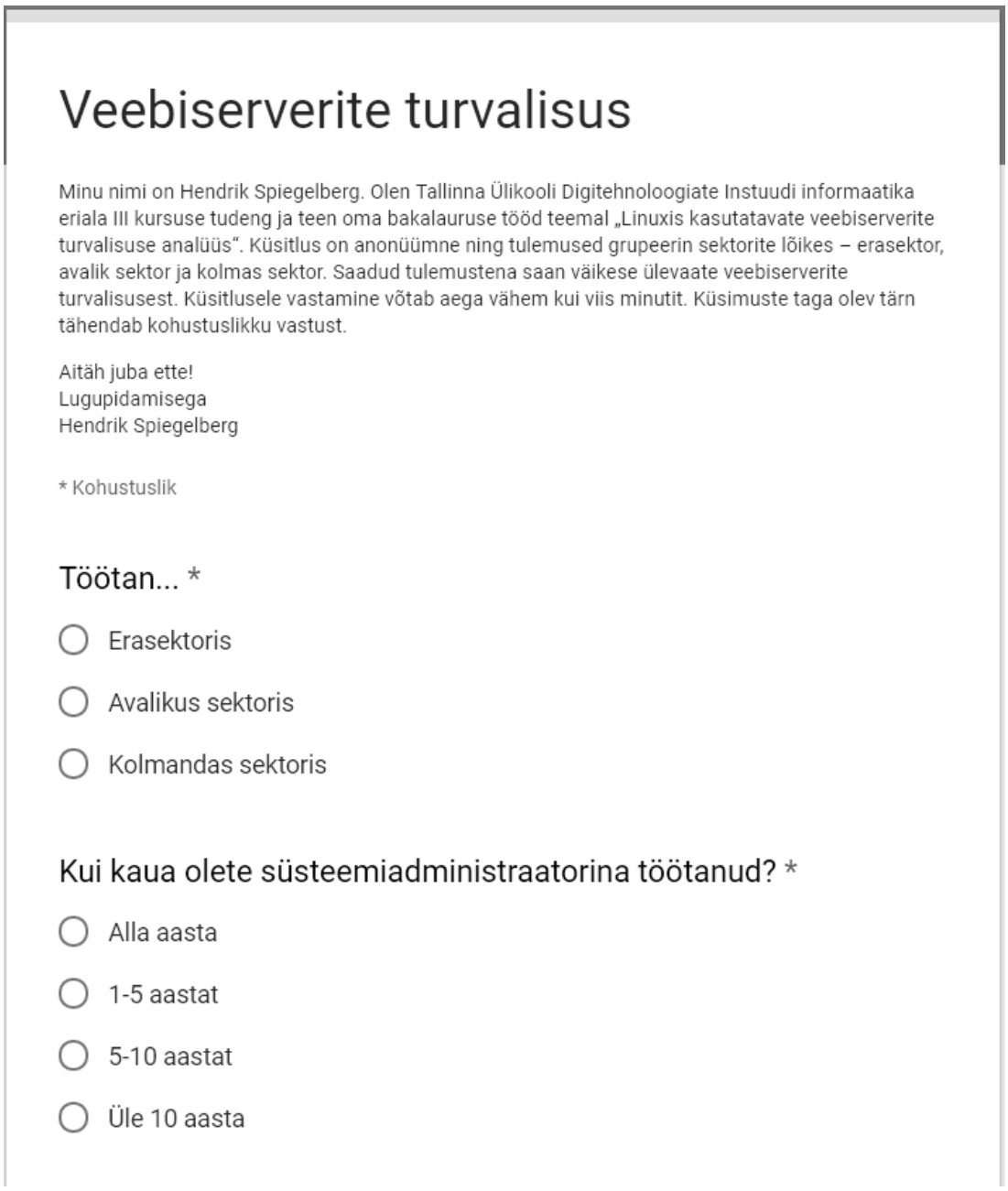
The second chapter is about the security of web servers. Explained there are the different types of attacks used against web servers, how to configure operating systems and web servers so that they are secure and the future of security for web servers.

In the third chapter, DoS (Denial of Service) attacks are carried out on virtual machines and the results are analysed. The installation and usage of a safeguard application is also explained.

In the fourth chapter a survey on the security of web servers was conducted among Estonian system administrators and in the fifth chapter the results of the survey are analysed.

Lisad

Lisa 1. Läbiviidud küsitluse ankeet



Veebiserverite turvalisus

Minu nimi on Hendrik Spiegelberg. Olen Tallinna Ülikooli Digitehnoloogiate Instuudi informaatika eriala III kursuse tudeng ja teen oma bakalauruse tööd teemal „Linuxis kasutatavate veebiserverite turvalisuse analüüs". Küsitlus on anonüümne ning tulemused grupeerin sektorite lõikes – erasektor, avalik sektor ja kolmas sektor. Saadud tulemustena saan väikese ülevaate veebiserverite turvalisusest. Küsitlusele vastamine võtab aega vähem kui viis minutit. Küsimuste taga olev tärn tähendab kohustuslikku vastust.

Aitäh juba ette!
Lugupidamisega
Hendrik Spiegelberg

* Kohustuslik

Töötan... *

- ◯ Erasektoris
- ◯ Avalikus sektoris
- ◯ Kolmandas sektoris

Kui kaua olete süsteemiadministraatorina töötanud? *

- ◯ Alla aasta
- ◯ 1-5 aastat
- ◯ 5-10 aastat
- ◯ Üle 10 aasta

Millist operatsioonisüsteemi kasutate? *

- [ ] Ubuntu
- [ ] Debian
- [ ] CentOS
- [ ] Ei oska öelda
- [ ] Muu: _____

Millist veebiserverit kasutate? *

- [ ] Apache
- [ ] Nginx
- [ ] Tomcat
- [ ] Node.JS
- [ ] Lighttpd
- [ ] Ei oska öelda
- [ ] Muu: _____

Kas Teie veebiserveri versioon on kõige uuem? *

- ○ Jah
- ○ Ei
- ○ Ei oska öelda

Kui ei ole, siis mis on selle põhjuseks?

Teie vastus

Kas Teie veebiserverit on rünnatud? *

- ◯ Jah
- ◯ Ei

Kui jah, siis millist tüüpi rünnakuga oli tegu?

Teie vastus

Kuidas hindaksite oma veebiserveri turvalisust? *

	1	2	3	4	5	
Väga ebaturvaline	◯	◯	◯	◯	◯	Väga turvaline

Millised oleksid Teie arvates kolm kõige tähtsamat asja, mida veebiserveri turvalisuse tagamiseks tegema peaks? *

Teie vastus

SAADA ÄRA