Tallinna Ülikool

Digitehnoloogiate instituut

Informaatika õppekava

VEEBILEHTEDE TURVALISUS JA SELLE RAKENDATUS EESTI HARIDUSASUTUSTE VEEBILEHTEDEL

Bakalaureusetöö

Autor: Ian Mario Naska

Juhendaja: Kristen Kivimaa

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 Ian Mario Naska (sünnikuupäev: 07.07.1994)

1. Annan Tallinna Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Veebilehtede turvalisus ja selle rakendatus Eesti haridusasutuste näitel”, mille juhendaja on Kristen Kivimaa, 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, _____

(digitaalne) allkiri ja kuupäev

Sisukord

Sissejuhatus

Autorit motiveeris antud teemal kirjutama asjaolu, et tegemist on iga päev järjest vajalikuma teemaga ning autor tunneb, et tuleb selles vallas tegeleda rohkem kasutajate teadlikkuse tõstmisega. Eestis räägitakse küberturvalisusest järjest enam, ka internetist on leitav palju harivaid artikleid ja juhendeid veebilehtede turvalisuse kohta, kuid kahjuks ei pöörata kasutajate teadlikkuse tõstmisele siiani piisavalt tähelepanu.

Kahjuks ei piisa ka alati sellest, et veebileht on turvaline. Vajalik on ka, et sellel veebilehel külastatavad kasutajad käituksid turvaliselt ja oleksid teadlikult ohtudest.

Käesolev bakalaureusetöö autori huvi veebilehtede turvalisuse vastu tekkis läbi tööalaste vajaduste ja kogemuste.

Bakalaureusetöö eesmärgiks on tutvustada populaarsemaid ründevõimalusi veebilehtede vastu ning anda soovitusi rünnakute vältimiseks ja vaadelda Eesti haridusasutuste veebilehtede seisukorda.

Eeldatakse, et käesoleva töö lugeja on IT-alase taustaga ja saab IT-terminitest aru.

Bakalaureusetöö koosneb kolmest peatükist. Esimeses peatükis tutvustatakse veebilehtede turvalisuse vajadust ning rünnakute eesmärke ja mooduseid. Teises peatükis toob autor välja veebilehtede populaarsemad võimalikud ründekohad ning näiteid rünnakute vältimiseks veebilehtedel. Viimases peatükis annab autor ülevaate Eesti haridusasutuste hetkeseisust ja soovitusi.

Analüüsist saadud tulemus on Autoril soov edastada Riigi Infosüsteemi Ameti infoturbeintsidentide käsitlemise osakonnale[1].

1 https://www.ria.ee/ee/cert.html

1 Veebilehtede turvalisus ja soovitused rünnete vältimiseks

Käesolevas peatükis annab autor ülevaate erinevate küberrünnakute stiilidest ja eesmärkidest. Samuti kirjeldab veebilehtede turvalisuse vajadust ning toob välja soovituslikud kohad, mida jälgida veebilehtede turvalisuse tagamiseks.

1.1 Rünnakute eesmärk

Eesmärke rünnakuteks on mitmeid. Kõige populaarsem eesmärk on teenida raha (Kaspersky Lab, 2014), kuid see ei ole alati peamine eesmärk. Näiteks firma, kes arendab tehnoloogiat sõjaväetööstusele võidakse rünnata, et saada kätte informatsiooni, mis võib sisaldada sõjaväelist, majanduslikku või poliitilist väärtust ründajale või ründaja tellinud kliendile. Sellistel juhtudel võib ründaja olla riigi tellitud või mõnest kuritegelikust rühmitusest, kes tegelevad riigi või mõne ettevõtte nimel (Lyndon, 2016).

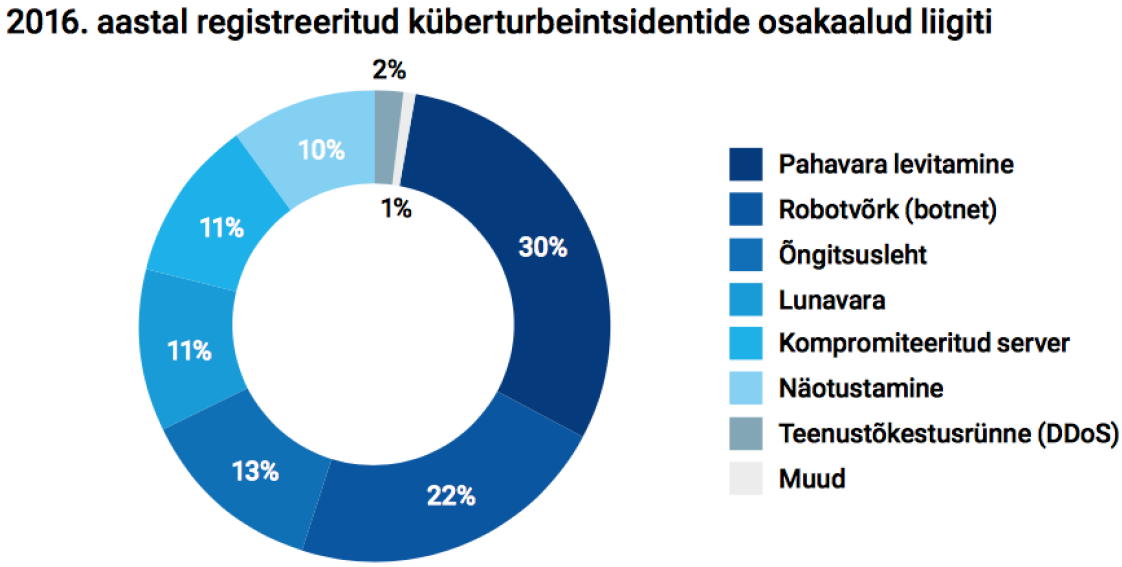
90-ndatel liikusid pigem petuskeemid, mis olid suunatud inimestele paludes neilt abi rahaliselt. Tänaseks on asjad jõudnu pahavaradeni, mida üritatakse paigaldada teiste kasutajate arvutitesse. Eesmärk on eelkõige teenida raha, varastada raha või omada teise isiku kontot.

1.2 Kuidas rünnakut tehakse

Rünnakute sooritamiseks on mitmeid viise. Suur osa registreeritud küberturvalisuse intsidentidest on aegunud tarkvara kaudu rünne (Küberturvalisuse teenistus, 2016). Selleks on ründajad/häkkerid loonud automaatskripte, mis otsivad vananenud sisuhaldussüsteemi versiooni peal olevat veebilehte, et selle turvaaugu kaudu rünnata.

Üheks suureks kõneaineks on olnud ka e-kirjade kaudu leviv lunavara (Küberturvalisuse teenistus, 2016)(vt. Illustratsioon 1). E-kirjana saadeti korrektses eesti keeles kiri, millega oli kaasas arve või mõni muu dokument. Pealtnäha tundus e-kiri olevat ausa asutuse või kodaniku poolt saadetud. E-mail tundus korralik, kirjas ei olnud vigu ja ka kirja teema võis minna kokku reaalse olukorraga. Näiteks uus töötaja saatis enda CV. Probleem seisnes selles, et peale manuse avamist käivitus fail, mis krüpteeris

kõik arvutis olevad failid ja nende tagasi saamiseks nõuti lunaraha. Ei saanud ka kunagi kindel olla, et peale lunaraha maksmist esialgsed failid tagastataks.



Illustratsioon 1: 2016. aastal registreeritud küberturbeintsidentide osakaalud liigiti (Küberturvalisuse teenistus, 2016)

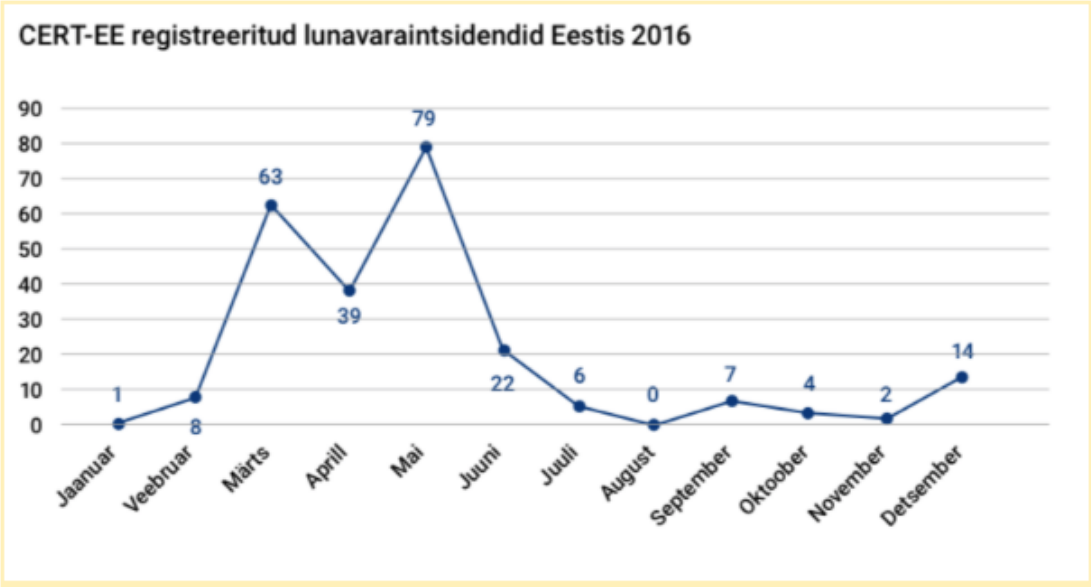
1.3 Veebilehe turvalikkuse vajadusest

Veebilehe turvalisus on väga tähtis. Autor eeldab, et iga firma oskab ise hinnata enda ohuallikaid ja tagajärgi. Kahjuks ei ole kahjude suurust kunagi ette teada, aga see on tähtis faktor näiteks luues järgmise aasta eelarvet või planeerides uusi töökohti ka mõne infoturbe eksperdi palkamiseks, et vältida võimalikke küberrünnakuid.

Olgu selleks väike veebileht või suurem portaal, tuleb sellest hoolimata rõhku panna veebilehe turvalikkusele ja/või infoturbele. Antud teema on iga päevaga rohkem aktuaalsem ja erinevaid turvaauke üritavad aina rohkem häkkereid ära kasutada (vt. Illustratsioon 2).

Autor arvab, et eelarve või aastaplaani koostamisel tuleb tõsiselt mõelda turvalisusele, olgu selleks veebilehe ajakohastamine või kontrollimine, firma töötajate teadlikkuse tõstmine, piiratud ligipääsudega arvutid töötajatele. Kehvasti turvatud veebilehed või arvutid võivad saada lõpuks mitu korda kulukamaks, kuna selle eest ei ole hoolitsetud

varem. Juhul, kui tegemist on mõne terviseasutuse või elutähtsa teenuse osutava firmaga, siis näiteks krüptolunavara levitamisega võidakse seada ohtu isegi inimeste elu ja tervis.



Illustratsioon 2: CERT-EE registreeritud lunavaraintsidendid Eestis 2016 (Küberturvalisuse teenistus, 2016)

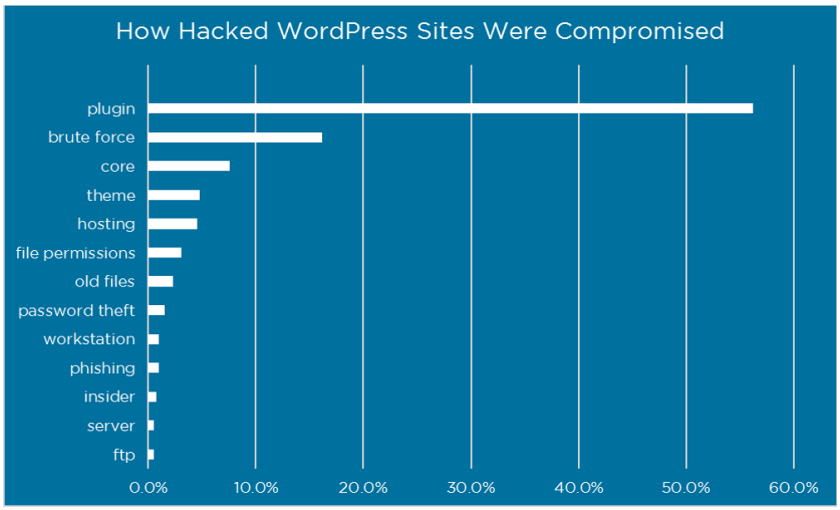
2 Lihtsamad ja keerukamad näited

Käesolevas peatükis toob autor välja erinevaid võimalikke ründekohtasid ja näiteid, millele tasuks mõelda veebilehe loomisel ning haldamisel.

2.1 Veebileht on ehitatud sisuhaldusplatvormil

Aegunud tarkvara põhjustab enamiku registreeritud küberturvalisuse intsidentidest (Küberturvalisuse teenistus, 2016)(vt. Illustratsioon 3), mis tähendab, et veebilehe arendaja või haldur on jätnud veebilehe platvormi või pluginad uuendamata ning seetõttu on ka võimalikud turvaaugud paikamata.

Sisuhaldussüsteemidel näiteks Wordpress[2], Joomla[3], Magento[4] kasutatakse väga palju pluginaid, mis lihtsustavad veebilehe loomist, kuid äärmiselt oluline on neid veebilehele paigaldades ka jälgida, millal viimati on seda pluginat uuendatud. Paljud pluginad on sisuhaldussüsteemide veebilehtedelt siiani allal aaditavad, kuid parandusi ja turvapaikamisi pole tehtud aastaid. Eeldatavasti on arendused lõppenud ja seetõttu võivad sisaldada suure tõenäosusega turvaauke. Lisaks puudutab see ka erinevaid veebiraamistikke (bootstrap[5] jne).



Illustratsioon 3: Ülevaade, mille kaudu enamus Wordpressi platvormil põhinevate veebilehti rünnatakse (Dan Moen, 2016)

2 https://wordpress.org
3 https://www.joomla.org
4 https://magento.com
5 https://getbootstrap.com

Kõige lihtsam esimene samm, mis aitab veebilehe turvalisusele kaasa, on valida endale turvaline salasõna, mitte kasutada kõige lihtsamaid nagu “Admin” või “123456”(Darell, 2017).

Veel soovitab autor jälgida veebiservereid, kuhu peale veebileht ehitati. Paljud veebimajutused teevad seda kasutajate eest, kuid on ka selliseid, kes ei tee seda kasutajate eest ja seetõttu on vajalik ise uuendada. Näiteks PHP-l[6] oli väga palju turvaauke varasemalt. Häkkerid on nautinud pikka aega neid auke. Õnneks PHP versioon 5-s on enamus auke parandatud, kuid neid leidub siiani (Security Vulnerabilities).

Tähtis on ka jälgida, et administraatori arvuti on puhas pahavaradest. Juhul, kui administraatori arvutisse on peidetud mõni jälgimisprogramm, näiteks keyloger[7], siis pole ka tugevast paroolist kasu, kuna antud tarkvara salvestab automaatselt kirjutatud paroolid endale.

Kui ollakse veebilehe haldur, siis on võimalus jälgida ka lehe kiirust või kontrollida, ega ei ole ilmunud lehele võõraid bännereid. Tasub ka jälgida, et kasutaja hallatav leht pole sattunud Google blacklisti[8]. Juhul kui on, siis suure tõenäosusega on veebileht nakatunud.

Näiteks lihtsasti peidetud pahavara võib olla pandud iframe[9] sisse, nii et lehte külastades isegi seda näha ei ole.

```
<iframe src="http://hackersite.com/attackfile.php"
width=100% height=0></iframe>
```

Koodinäide 1: iframe sisse kirjutad pahalane ehk ründaja poolt lisatud php leht, mida veebilehte külastades näha ei ole

2.2 Veebilehel krüpteeritud ühendus (https)

HTTP (Hyper Text Transfer Protocol) on viis, kuidas veebileht jagab andmeid serveriga. HTTPS on krüpteeritud versioon standardsest http protokollist, millel on SSL

6 http://php.net

7 https://free-keylogger.en.softonic.com

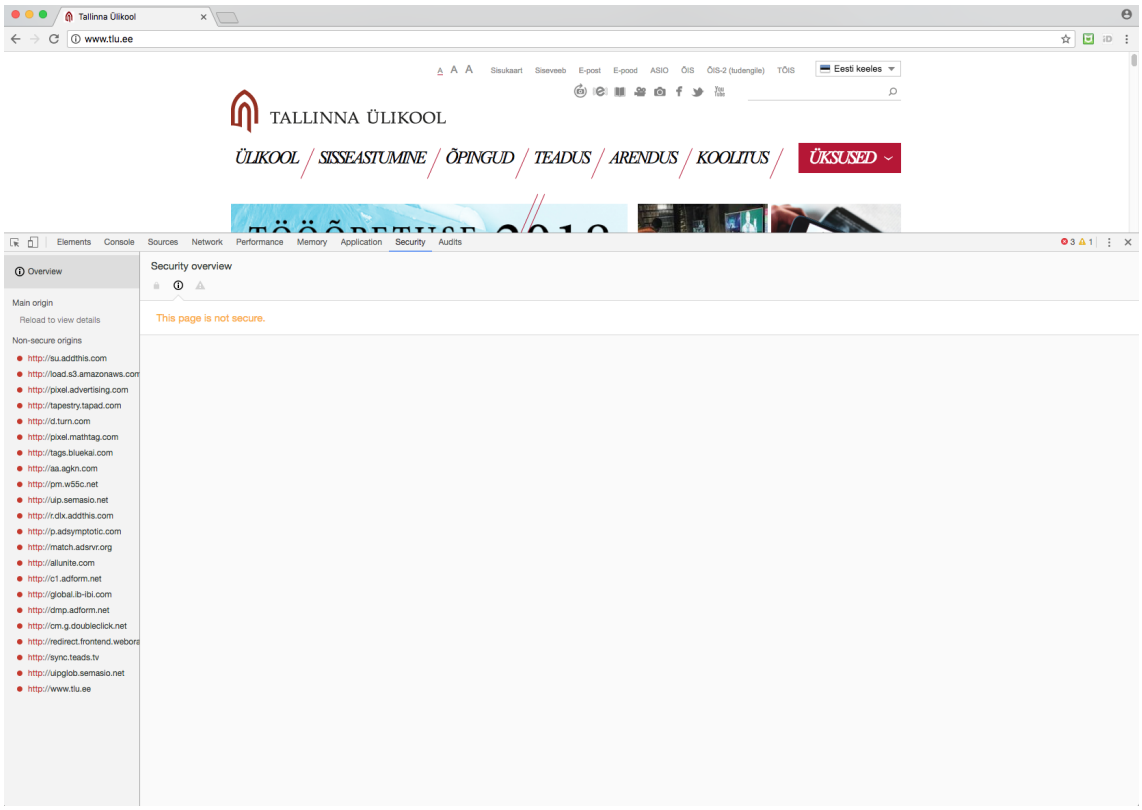
8 https://sucuri.net/guides/what-is-google-blacklist

9 https://html.com/tags/iframe/

sertifikaat. Veebipoes ostes kui ka veebipoodi arendades tuleb jälgida, et veebiliikluses kasutatakse krüpteeritud ühendust ehk https ühendust. See on vajalik selleks, et e-poes, kus on krediitkaardi andmete sisestamine või toimub mõni muu isikuandmete jagamine toimiks turvaliselt ja keegi võõras ei saaks antud andmetele ligi. Https peal jooksvad veebilehed ja e-poed annavad ka klientidele kindlustunde, et leht on turvaline ja usaldusväärne ning selle eest on hoolitsetud (vt. Illustratsioon 4).

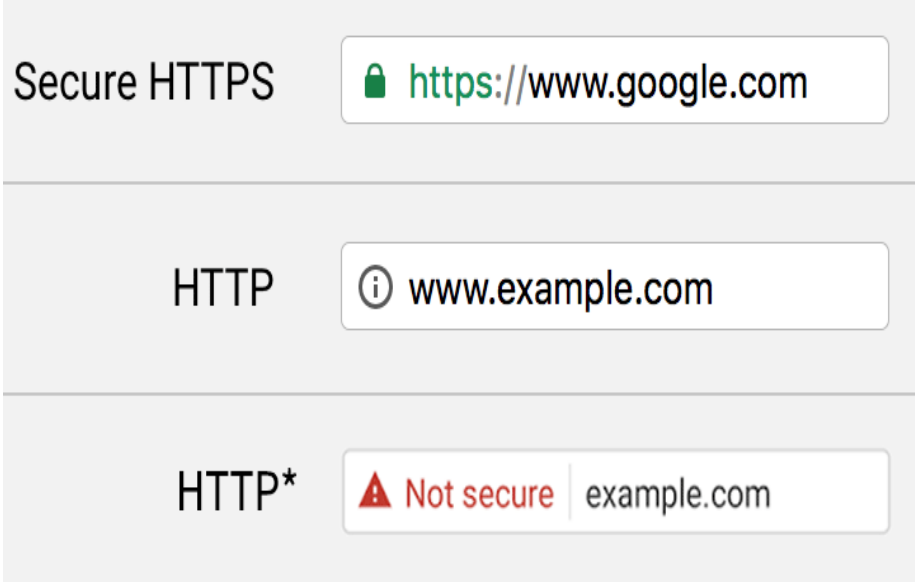
Alates 17.oktoobrist 2017 kuvab Google Chrome[10] veebilehitseja ühenduse mitte turvaliseks juhul, kui veebileht ei ole https peal või pole korralikult SSL sertifikaati ühendatud (Liam, 2017). Samuti on Google jaoks see üks faktoriteks, et lehte kuvatakse Google otsingumootoris.

Juhul, kui on soov olla kindel, kas veebilehel andmete liiklus on turvaline, siis selleks tuleks avada Google Chrome veebilehitseja, sisestada soovitud veebilehe aadress ja üleval aadressi juures vasakul nurgas on turvalisuse staatus (vt. Illustratsioon 5).



Illustratsioon 4: Tallinna Ülikooli koduleht

10 https://www.google.com/chrome/browser/desktop/index.html



Illustratsioon 5: Google Chrome annab annab kasutajale teada, kas veebileht kasutab krüpteeritud ühendust. Roheline teavitab, et ühendus on turvaline. Punane teavitab, et ühendus ei ole turvaline. Hall teavitab, et ühendus osaliselt pole turvaline (Brett, 2017)

Eestis pakub näiteks SSL sertifikaati Veebimajutus (Elkdata OÜ)[11]. "Let's Encrypt SSL" sertifikaat on tasuta. Autor on katsetanud Veebimajutuse kasutajana http peal oleva veebilehe üleviimist https ühendusele ning antud tegevus oli tehtud äärmiselt lihtsaks. Autori arvates ei tohiks sellega jääda hätta ükski vähegi IT teadev inimene.

2.3 Veebilehe andmebaasi süstimisrünne

Üks võimalikke ründekohti on andmebaasid. Andmebaaside vastu toimuvat rünnakut nimetatakse andmebaasi süstimisrünne ehk SQL injection (SQL Injection, 2016) (vt Illustratsioon 6).

SQL süstrünnak on see, kui ründaja kasutab veebivormi või url parameetreid, et pääseda veebilehe andmebaasiga manipuleerima. Kui veebilehel on kasutusel standart Transact SQL[12], siis on ründajal lihtne sisestada päringusse petlik kood, millega saab muuta andmebaasis olevaid tabeleid, saada informatsiooni või kustutada sealt ära andmeid. Seda saab veebilehe arendaja ära hoida kasutades parameetriga päringuid ning enamikus programmeerimiskeeltes on see võimalus toetatud ja lihtsasti rakendatav.

11 https://www.veebimajutus.ee

12 https://technet.microsoft.com/en-us/library/ms189826(v=sql.90).aspx

Näide päringust:

```
"SELECT * FROM table WHERE column = '" + parameter + "';"
```

Koodinäide 2: Näide väga lihtsast rünnatavas SQL koodist

Kui ründaja muudab urli parameetrit, mis edastatakse ' või '1'='1, siis päring hoopis näeb hoopis välja selline:

```
"SELECT * FROM table WHERE column = '' OR '1'='1';"
```

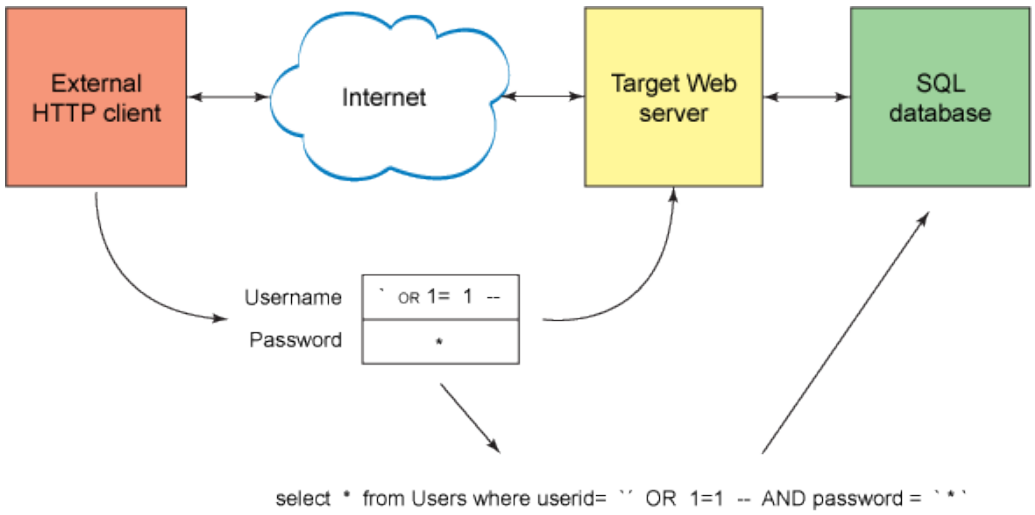
Koodinäide 3: Ründaja saab muuta antud parameetreid endale sobivaks

Kuna “1” võrdub väärtusega “1”, siis see võimaldab ründajal lisada täiendav päring SQL avaldise lõppu, mis samuti käivitatakse.

Selleks, et seda vältida, tuleks päringut täiendada täpsemate parameetritega. Näiteks, kui kasutada PHP-s MySQL-i , siis võiks päring näha välja selline:

```
$stmt = $pdo->prepare('SELECT * FROM table WHERE column
= :value');
$stmt->execute(array('value' => $parameter));
```

Koodinäide 4: Näide, kuidas korrektselt PHP-s kasutada SQL-i



Illustratsioon 6: Veebilehe andmebaasi süstimisrünne (M.Jones)

2.4 Javascript ründed

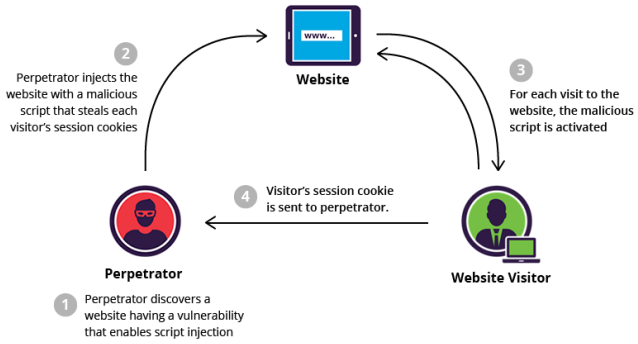
XSS[13] rünne on internetis väga levinud rünne (Tartu Ülikool, 2015) ehk tavalisele veebilehele lisatakse javascirpti[14] koodilõik, mis kutsub esile rünnakut.

Tänapäeval veebilehed saavad sisu erinevatest serveritest, näiteks pildid, reklaamid või mõned javascript veebiraamistikud. Selleks, et see kasutaja privaatsust ei ohustaks, on välja töötatud reegel, et laetud sisu saab ainult ligi selle lehe osale, mis on laetud samast kohast (Same-origin policy[15]) ja XSS rünne ise on selle reegli eiramine.

Pahavara paigaldatakse väga lihtsalt. Juhul, kui veebilehel on otsing või on võimalus lisada kommentaar või hoopis mõni muu väli, kus aktsepteeritakse HTML[16] koodi, siis saab ründaja sinna lihtsasti kopeerida enda scripti ehk pahalase. Kui kasutaja näiteks sellele lehele satub, siis saab ründaja võtta kasutaja veebilehe küpsised (browser cookies[17]) ja selle abil kasutaja sessiooni üle võtta.

Mida saab teha veebilehe haldur?

Tuleks mõelda, kas html koodi lisamise võimalus on vajalik otsingu või kommentaari väljale sisestamiseks. Juhul, kui ei ole, siis on mõistlik lülitada selle tugi välja.



Illustratsioon 7: Kuvatud, kuidas toimub javascripti XSS rünne (Imperva Incapsula)

13 https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
14 https://www.javascript.com
15 https://en.wikipedia.org/wiki/Same-origin_policy#Cross-Origin_Resource_Sharing
16 https://html.com
17 https://www.nczonline.net/blog/2009/05/05/http-cookies-explained/

2.5 Pildilaadimine

Järgmiseks rünnakuvõimaluseks on pildilaadimine. Juhul, kui veebilehel on võimalus pilte üles laadida (näiteks kasutaja profiilipilt), siis see võib olla suur turvarisk (Unrestricted File Upload, 2017).

Probleem seisneb selles, et isegi süütuna tunduvad failid võivad sisaldada skripti, mis veebiserverisse jõudes täielikult avanevad ja tekitavad veebilehel pahandusi.

Juhul, kui on veebilehel üleslaadimise vorm, siis tasuks kõiki faile käidelda ehk kontrollida. Ei saa tugineda ainult faililaiendile ega mime tüübile (mime type[18]), et kontrollida, kas fail on pilt või mitte, sest seda saab lihtsasti võltsida. Isegi pildi suuruse kontrollimine või selle päise lugemine ei anna piisavat kindlust. Enamik piltide vorminguid võimaldab salvestada kommentaarosa, mis sisaldab php koodi, mida server võib töödelda.

Mõned soovitused:

Võimalused on näiteks faili üleslaadimisel nimetada ümber fail, et olla kindel korrektses faililaiendis[19]. Võimalus on ka muuta failiõigusi, näiteks chmod 0666[20], nii et seda ei saaks korda saata.

Kui kasutate sisuhaldussüsteemi, siis võite täiendada enda .htaccess[21] faili, mis lubaks juurdepääsu ainult failidele, mis takistavad topelt laienduste rünnakut.

Näide:

```
deny from all
<Files ~ "^\w+\.(gif|jpe?g|png)$">
order deny,allow
allow from all
</Files>
```

Koodinäide 5: .htaccess failis on seotud piirangud failiformaatidele, mis ei võimalda topelt laienduste rünnakut

18 https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Complete_list_of_MIME_types

19 https://www.computerhope.com/jargon/f/fileext.htm

20 http://www.thinkplexx.com/learn/article/unix/command/chmod-permissions-flags-explained-600-0600-700-777-100-etc

21 http://www.htaccess-guide.com

Soovitatav on takistada üleslaaditud failidele otsest juurdepääsu. Nii salvestatakse veebilehele üleslaaditud failid kausta, mis ei kuulu veebilehe süsteemsete failide juurde või andmebaasi. Kui need failid pole otseselt juurdepääsetavad, peab looma skripti, mis aitavad saada kätte failid ja saata need brauserisse. Pildi sildid toetavad atribuuti src, mis pole pildi otsene url, nii et brauseri lähtekood võib viidata veebilehe failide edastamise skriptile, kui määrata https päises õige sisutüüp. Näiteks:

```
<img src="/imageDelivery.php?id=1234" />
<?php
// imageDelivery.php
// Fetch image filename from database based on $_GET["id"]
// Deliver image to browser
Header('Content-Type: image/gif');
readfile('images/'.$fileName);
?>
```

Koodinäide 6: Lähtekood viitab edastamise skriptile, kui määratakse korrektselt https päises sisutüüp

Kui lubatakse faile üles laadida internetist, siis soovitav on kasutada ainult turvalisi transpordimeetodeid serverisse, näiteks SFTP[22] või SSH[23].

2.6 Autentimis- ja sessioonihalduse vead

Võimalikuks ründekohaks võib olla kui autentimis- ja sessioonihaldus ei ole korralikult rakendatud, võimaldades ründajatel varastada salasõnu, võtmeid, sessioonitunnuseid või rakendusvigu ning neid konfidentsiaalsete andmete teadasaamiseks ära kasutada(Hari, 2017).

Arendajad ehitavad tihti ise autentimise ja seansihaldussüsteeme, kuid nende loomine on keeruline. Selle tulemusena on enda ehitatud skeemidel sageli puudused sellistes kohtades nagu kontolt väljumine, konto loomine, parooli muutmine, parooli unustamine, salajane küsimus jne. Selliste vigade leidmine võib mõnikord olla keeruline, kuna iga rakendus on unikaalne.

22 https://www.digitalocean.com/community/tutorials/how-to-use-sftp-to-securely-transfer-files-with-a-remote-server

23 https://www.ssh.com/ssh/protocol/

Kuidas kontrollida, kas veebileht on haavatav või mitte:

1. Kasutaja autentimisel sisestatud andmed ei ole korralikult kaitstud ehk andmebaasis olevad andmed pole krüpteeritud või räsina.

2. Andmeid saab arvata või kirjutada üle nõrkade kontohaldusfunktsioonide kaudu (nt konto loomine, parooli muutmine, parooli taastamine, nõrk sessiooni ID).

3. Sessiooni ID-d kuvatakse url-is (nt url-i ümberkirjutamine).

4. Pärast edukat autentimist ei muutu sessiooni ID.

5. Paroolid, sessiooni ID-d ja muid andmed ei edastata läbi krüpteeritud ühenduse ehk https.

Näide ründest:

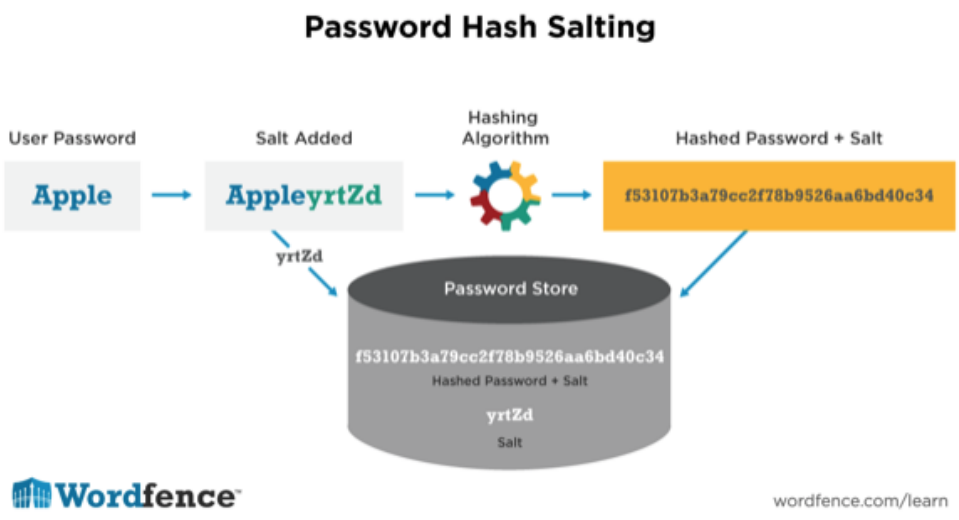
Lennupiletite broneerimise rakendus toetab url-i ümberkirjutamist, sessiooni ID-de määramist url-is:

```
http://example.com/sale/saleitems?
sessionid=268544541&dest=Hawaii
```

Koodinäide 7: Veebilehe urli kaudu jagatakse enda sessiooni ID, mida näiteks sõber saab edasi kasutada

Veebilehel sisenenud kasutaja soovib näiteks teavitada sõpra müügist ja saadab ülaltoodud lingi (ilma teadmata, et edastab ka enda sessiooni ID). Nüüd kui sõber kasutab seda linki, siis ta kasutab ka lingi saatnud sõbra sessiooni. Juhul, kui sõber on ka varasemalt sisestanud enda krediitkaardi andmed sinna süsteemi, siis lingi saanud sõber saab tema krediitkaarti ka kasutada.

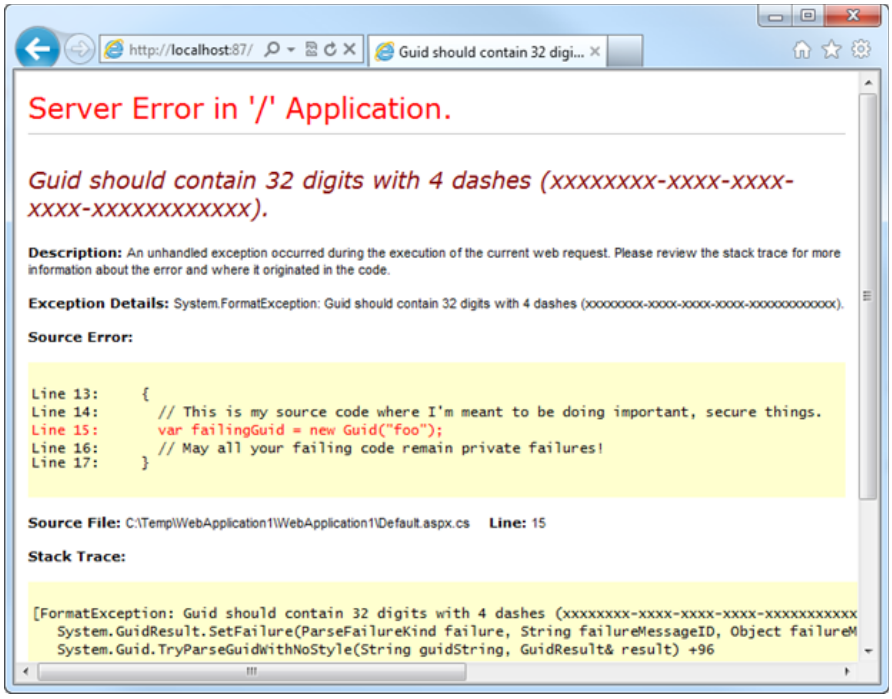
Teine näide on olukord, kui ründaja saab juurdepääsu süsteemi paroolide andmebaasile. Kui kasutajate paroolid pole korralikult räsitud või soolatud, siis on kõikide kasutajate paroolid näha andmebaasis (vt. Illustratsioon 8).



Illustratsioon 8: Parooli soolamine (Wordfence, 2016)

2.7 Veateade

Kindlasti tasub jälgida, kui palju antakse edasi infot veateadete korral kasutajale. Kasulik on võimalikult vähe kuvada kasutajale veateadet ja selles sisaldavat infot, sest seda võivad koheselt kasutada ära ründajad/häkkerid (Team Mentor, kuupäev puudub). Soovitav on hoida detailset veateadet enda serveri logides ja kuvada kasutajale ainult nende jaoks vajalikku infot (vt. Illustratsioon 9).



Illustratsioon 9: Kuvatõmmis liiga informatiivsest veateatest

2.8 Objektide ebaturvaline otseviitamine

Objekti otseviitamine esineb, kui arendaja jätab avalikuks viite mõnele sisemise rakenduse dokumendile, nagu fail, kaust või andmebaasivõti. Ilma ligipääsukontrolli või mõne muu tõkketa võivad ründajad neid viiteid töödeldes andmetele volitamata ligi pääseda (vt. Illustratsioon 10).

Parim viis teada saada, kas rakendus on objekti otseviitamise suhtes ebaturvaline, on kontrollida, kas kõik objekti viited on sobivalt kaitstud.

Koodi ülevaatus võib anda kiiresti tulemuse, kas mõlemad viisid on turvaliselt rakendatud. Samuti, kas testimine on väga efektiivne otseste viidete tuvastamiseks ja kontrollimiseks. Automaatsed tööriistad tavaliselt ei näe selliseid auke, sest need ei suuda tuvastada, mis vajab kaitset ning mis on ohutu või ohtlik.

Näide ründest:

Rakendus kasutab kontrollimata andmeid SQL-is, mis pääseb ligi kontoteabele:

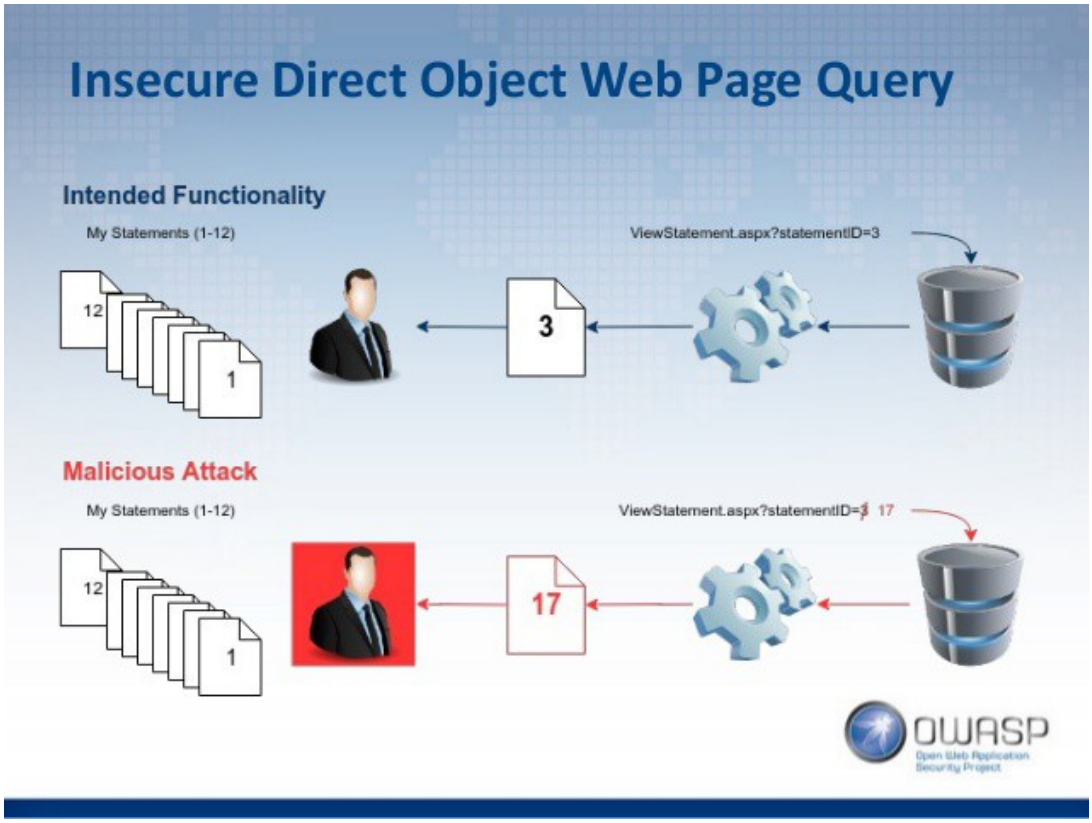
```
String query = "SELECT * FROM accts WHERE account =?";
PreparedStatement pstmt =
connection.prepareStatement(query , … );
pstmt.setString( 1, request.getParameter("acct"));
ResultSet results = pstmt.executeQuery( );
```

Koodinäide 8: Ründaja muudab SQL päringut, et saada sisenede andmebaasi, nägemata, mis kontod seal on

Ründaja muudab oma brauseris parameetrit "acct", et saada mis tahes kontole ligi. Kui see pole kinnitatud, siis võib saada ründaja juurdepääsu kõikidele kasutajakontodele.

```
http://example.com/app/accountInfo?acct=notmyacct
```

Koodinäide 9: Nüüd ründaja sisestab acct url parameetrisse, et saada kätte võõra inimese konto



Illustratsioon 10: Näidis, kuidas ründaja muudab sessiooni ID-d, et saada ligi võõra isiku andmetele. Juhul, kui seda tegevust ei ole tõkestatud, siis ründaja saab lihtsasti võõra isiku andmetele ligi (Medhat, 2017)

2.9 Vigased turvaseaded

Heatasemeline turvalisus nõuab turvalise seadistuse määratlemist, paigaldamist ja haldamist nii rakenduses, raamistikus, platvormis, rakendus-, veebi- ja andmebaasiserveris (Stephen, 2017). Vaikimisi on seaded tihti ebaturvalised. Kui tarkvara on kehvasti seadistatud või uuendamata, võib süsteemile ligi saada turvaauke ära kasutades (vt. Illustratsioon 11). Tarkvara pidev uuendamine on vajalik.

Turvakonfiguratsiooni valesti seadistamine võib juhtuda rakenduste mis tahes tasemetel, sealhulgas platvormil, veebiserveril, rakendusserveril, andmebaasil, raamistikel ja enda loodud koodis. Arendajad ja süsteemihaldurid peavad soovituslikult tegema koostööd, et kogu süsteem (algusest lõpuni) oleks korralikult konfigureeritud.

Selleks, et kontrollida, kas süsteem vastab turvalisusele tasub küsida järgmiseid küsimusi:

1. Kas tarkvara on aegunud (veebiraamistikud, serverid, rakendused jne)?

2. Kas mõni tarbetu funktsioon on lubatud või installitud, mida ei kasutata (nt pordid, teenused, lehed, kontod, õigused)?

3. Kas vaikimisi kontod ja nende paroolid on endiselt lubatud ja muutmata?

4. Kas veateade näitab kasutajale liiga palju ebavajalikku informatsiooni?

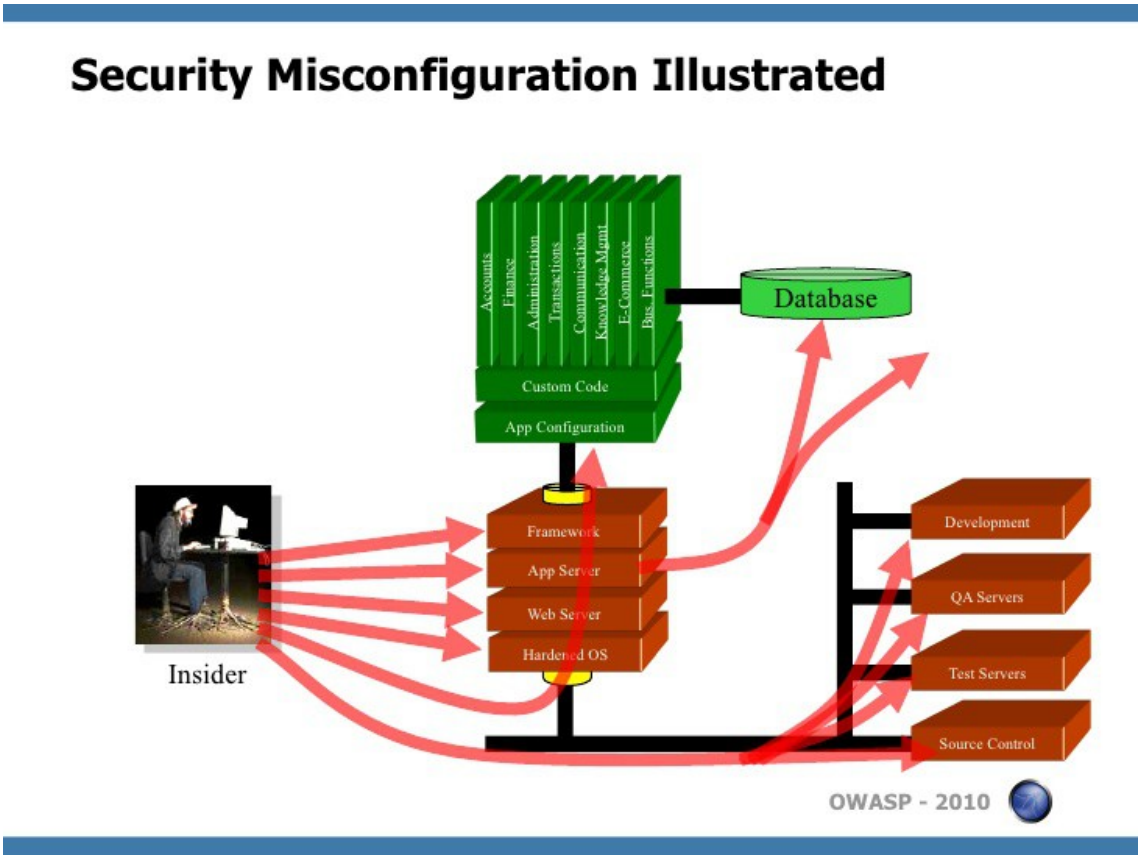
5. Kas turvaseaded rakendusserverites, rakenduste raamistikes, andmebaasides pole väärtused turvalisuseks seadistatud?

Näide ründest:

1. Uue kataloogi lisamine ei ole veebiserveris blokeeritud või keelatud. Nii saavad ründajad failides liikuda ringi ja otsida endale vajalikke faile. Näiteks ründaja leiab ja laeb alla kõik koostatud java klassid (java classes[24]), mida nad enda jaoks kohandavad ja tagasi asetavad.

2. Rakendusserver tuleb tavaliselt näiterakendusega ja seda pole eemaldatud serverist. Taolised näiterakendused on hästi teada ründajatele.

24 https://docs.oracle.com/javase/tutorial/java/concepts/class.html



Illustratsioon 11: Kuvatud, kuidas saadakse ligi vigaste turvaseadete abil andmebaasile (Software Guru)

2.10 Puuduv pääsukontroll funktsionaalsuse tasemel

Paljud veebirakendused kontrollivad enne funktsionaalsuse näitamist veebilehitsejas ligipääsuõigusi. Siiski tuleks samasugune kontroll teha ka serveris. Kui pöördumisi ei kontrollita, võivad ründajad neid võltsides funktsionaalsusele volitamata ligi pääseda.

See võib tuleneda sellest, et funktsioonide tasemete kaitset hallatakse läbi konfiguratsiooni ja süsteemid on valesti konfigureeritud.

Selliste vigade tuvastamine on lihtne. Kõige raskem on tuvastada, millised leheküljed (URL-id) või funktsioonid eksisteerivad rünnaku jaoks.

Parim viis teada, kas rakendus ei suutnud funktsionaalse juurdepääsu korral korralikult piirata, on iga rakenduse funktsiooni kontrollimine:

1. Kas kasutajaliides näitab navigatsiooni volitamata funktsioonide jaoks?
2. Kas serveripoolne autentimine või volituste kontroll puudub?
3. Kas serveripoolsed kontrollid on tehtud tuginedes ründaja poolt pakutavale teabele?

Kasutada puhverserverit, tuleb sirvida rakendust suuremate õigustega (privileged role). Seejärel vaadata piiratud lehti, kasutades väiksemate õigustega kasutajaga. Kui serveri vastused on sarnased, on rakendus tõenäoliselt haavatav.

Näide:

Ründaja paneb brauseri ründama urle. Üks neist on avalik ja teine ligipääsetav ainult Admin kasutajale.

```
http://example.com/app/getappInfo
http://example.com/app/admin_getappInfo
```

Koodinäide 10: Õiguste kontrollimine erinevate lehtede põhjal, kus ühe näeb ainult admin kasutaja

Kui mittesisenenud kasutaja saab juurdepääsu mõlemale lehele, on see probleem ehk viga. Kui autentifitseeritud (sisenenud), mitte admin, kasutajal on lubatud juurdepääs admin_getappInfo lehele, siis see on ka viga.

3 Eesti haridusasutuste hetkeseisust

Käesolevas peatükis annab autor ülevaate Eesti haridusasutuste veebilehtede hetkeseisust, võttes aluseks eelnevas peatükis välja toodud näiteid. Autor on võtnud analüüsimiseks “Ettevõtlik kool” programmiga liitunud koolid[25]. Programm sai valitud põhjusel, kuna selle programmiga liitunud koolide veebilehti võivad külastada rohkemad inimesed ja lisaks on Neti[26] otsingumootoris küllaltki kõrgel kohal.

3.1 „Ettevõtlik kool” programmist üldiselt

“Ettevõtlik kool” on haridusprogramm, mis on oma tööd alustanud juba 2006 aastal Ida-Virumaal. Programm on suunatud ettevõtliku õppe integreerimise koolisüsteemi, et tõsta hariduse kvaliteeti ja seeläbi noorte edukust elus.

„Ettevõtliku kooli“ võrgustikku kuuluvad haridusasutused, kes on oma õppeasutuses seadnud hariduse eesmärgiks ettevõtliku hoiaku kujundamise läbi õppetöö toetudes riiklikule õppekavale.

„Ettevõtliku kooli“ võrgustik on laienenud igasse Eesti maakonda, kus kohalikul tasandil koordineerivad programmi maakondlikud arenduskeskused (MAK).(Ettevõtlik kool, 2016)

3.2 Sissejuhatus analüüsist

Aluseks on võetud ainult koolid, kes on “Ettevõtlik kool” programmiga liitunud. Kokku sai analüüsitud kõiki 78 kooli. Koolide veebilehed sai skänneeritud https://sitecheck.sucuri.net skänneriga, samuti uuritud veebilehel kasutajana sisenemist ja veateteid.

Skännereid leiti teisigi, näiteks VirusTotal[27] ja urlQuery[28], kuid Sucuri[29] oli kõige laialdasema skänneeringuga ehk oskab tuvastada, millise sisuhaldussüsteemiga ja versiooniga veebileht on arendatud. Samuti suudab see tuvastada scripte, linke ja kas

25 http://evkool.ee/ettevotlik-kool/
26 https://www.neti.ee/cgi-bin/teema/HARIDUS_JA_KULTUUR/Haridus/
27 https://www.virustotal.com/#/home/url
28 https://urlquery.net/search
29 https://sitecheck.sucuri.net

veebileht on kuskil blacklist nimekirjas. VirusTotal näitas ära, kelle majutuses veebileht on, kuid see ei anna otsest tulemust lehel sisaldavatest pahavaradest. UrlQuery kuvas välja aktiivsed lingid sisestatud veebilehel ja suunamised.

Sitecheck.sucuri.net skänner tuvastab kas:

- veebilehel on mõni pahavara
- veebileht on lisatud musta nimekirja (blacklist)
- esineb mõningaid anomaaliaid või probleeme
- kas kasutatakse uuemaid tarkvarasid ja platvorme

Nimekiri koolidest oli järgnev:

IDA-VIRUMAA

Iisaku Gümnaasium, Jõhvi Põhikool, Jõhvi Vene Põhikool, Kiviõli 1. Keskkool, Kohtla-Järve Ahtme Gümnaasium, Kohtla-Järve Järve Gümnaasium, Kohtla-Järve Maleva Põhikool, Kohtla-Järve Tammiku Põhikool, Lüganuse Keskkool, Maidla Kool, Narva Soldino Gümnaasium, Sillamäe Vanalinna Kool, Sinimäe Põhikool

LÄÄNE-VIRUMAA

Haljala Gümnaasium, Kunda Ühisgümnaasium, Põlula Kool, Rakke Kool, Rakvere Gümnaasium, Rakvere Põhikool, Tamsalu Gümnaasium, Uhtna Põhikool, Vajangu Põhikool, Väike-Maarja Gümnaasium

JÄRVAMAA

Järva-Jaani Gümnaasium, Koeru Keskkool, Paide Ühisgümnaasium, Peetri Kool, Roosna-Alliku Põhikool, Türi Põhikool, Türi Ühisgümnaasium, Väätsa Põhikool

RAPLAMAA

Juuru Eduard Vilde Kool, Kohila Gümnaasium, Rapla Vesiroosi Gümnaasium

LÄÄNEMAA

Palivere Põhikool

PÄRNUMAA

Jõõpre Kool, Kihnu Kool, Koonga Kool, Lõpe Kool, Pärnu-Jaagupi Gümnaasium, Pärnu Rääma Põhikool, Tõstamaa Keskkool, Vändra Gümnaasium

VILJANDIMAA

Suure-Jaani Kool, Viljandi Kesklinna Kool, Viljandi Paalalinna Kool

JÕGEVAMAA

Adavere Põhikool, Anna Haava nimeline Pala Kool, Torma Põhikool, Voore Põhikool

VÕRUMAA

Antsla Gümnaasium, Kääpa Põhikool, Meremäe Kool, Mõniste Kool, Osula Põhikool

VALGAMAA

Hargla Kool, Tsirguliina Keskkool, Valga Põhikool

TARTUMAA

Ilmatsalu Põhikool, Tartu Descartes'i Kool, Tartu Hansa Kool

PÕLVAMAA

Kanepi Gümnaasium, Mikitamäe Kool, Mooste Põhikool, Saverna Põhikool, Värska Gümnaasium, Vastse-Kuuste Kool

HARJUMAA

Aegviidu Kool, Nissi Põhikool, Turba Kool

HIIUMAA

Kärdla Põhikool, Lauka Põhikool

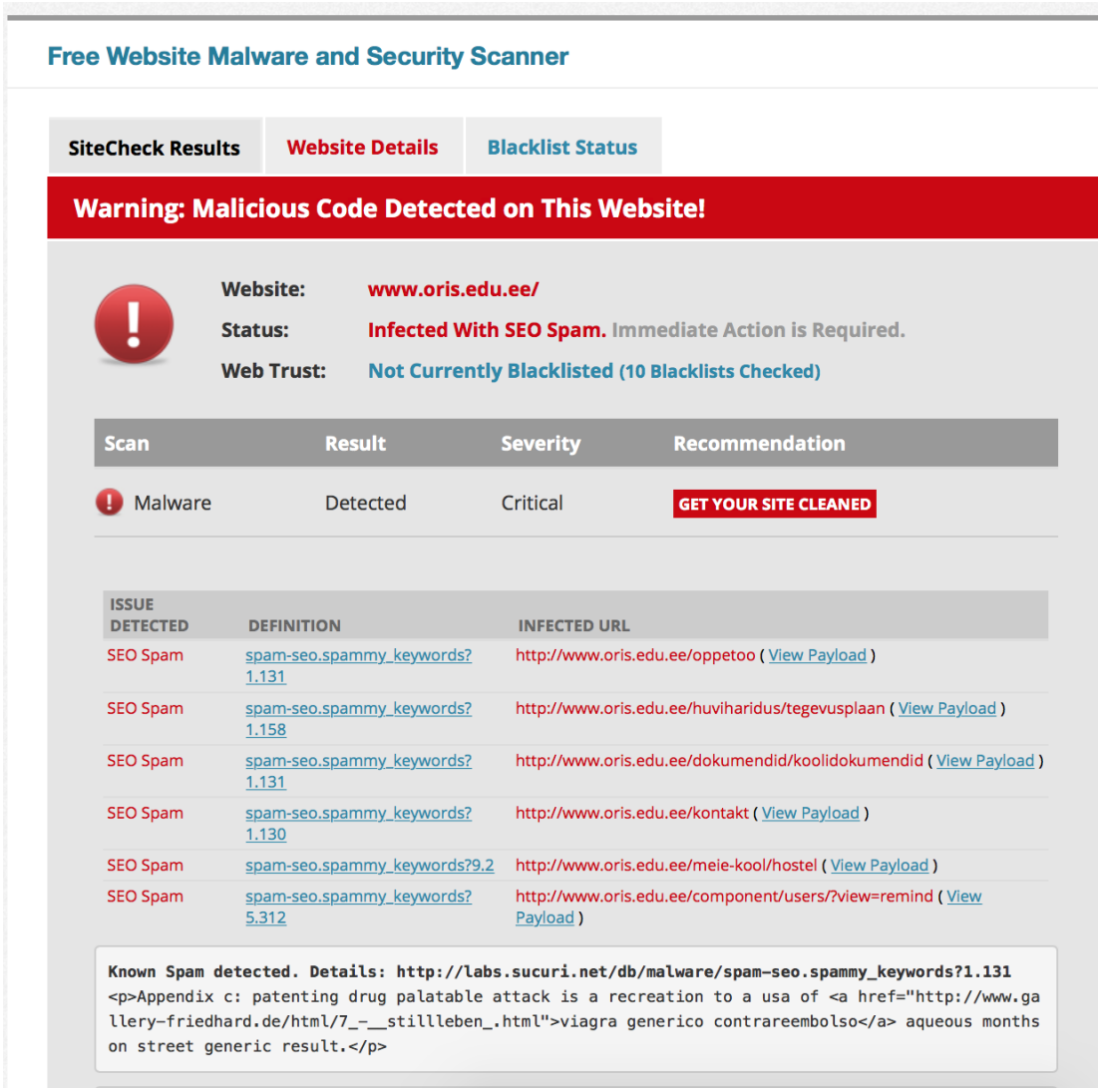
SAAREMAA

Aste Põhikool, Kärla Põhikool, Leisi Keskkool, Lümanda Põhikool, Orissaare Gümnaasium, Salme Põhikool, Valjala Põhikool

3.3 Tulemus

Kokku 78-st koolist oli kodulehed hooldatud 49 koolil, mis on 63% kogu koolide arvust. Vananenud tarkvara peal avastati 29 koolil, mis teeb 37% antud koolidest.

Analüüsi käigus tuvastati ka Orissaare Gümnaasiumi kodulehel pahalane (vt. Illustratsioon 12). Nimelt oli lehelt tehtud välja suunamine võõrale lehele https://www.bi-central.net (vt. Illustratsioon 13).



Illustratsioon 12: Kuvatõmmis Orissaare Gümnaasiumi veebilehel leitud pahalane, mis on tingitud hooldamata veebilehest

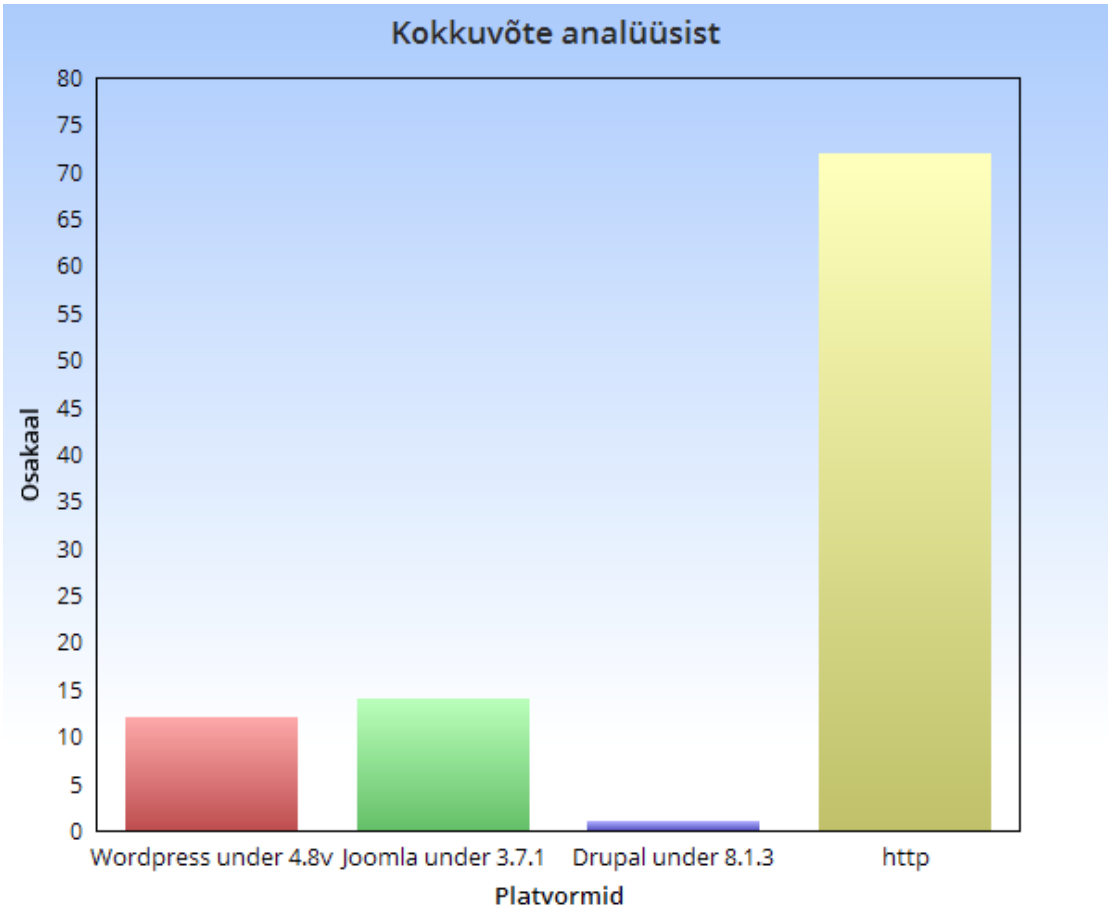
ISSUE DETECTED	DEFINITION	INFECTED URL
SEO Spam	spam-seo.spammy_keywords? 1.131	http://www.oris.edu.ee/oppetoo (View Payload)
SEO Spam	spam-seo.spammy_keywords? 1.158	http://www.oris.edu.ee/huviharidus/tegevusplaan (View Payload)
SEO Spam	spam-seo.spammy_keywords? 1.131	http://www.oris.edu.ee/dokumendid/koolidokumendid (View Payload)
SEO Spam	spam-seo.spammy_keywords? 1.130	http://www.oris.edu.ee/kontakt (View Payload)
SEO Spam	spam-seo.spammy_keywords?9.2	http://www.oris.edu.ee/meie-kool/hostel (View Payload)
SEO Spam	spam-seo.spammy_keywords? 5.312	http://www.oris.edu.ee/component/users/?view=remind (View Payload)

Illustratsioon 13: Kuvatõmmis Orissaare Gümnaasiumi kodulehelt leitud pahalase detailvaatest, kus on näidatud ära kuhu lehele kasutajat sooviti suunata

Autor tuvastas ka, et Leisi Keskkooli kodulehel ilmus liiga informatiivne veateate: “Andmebaasiga ei saanud ühendust, kontrolli, kas failis admin/andmebaas.php on ikka kõik õigesti konfigureeritud!!!”. See võib anda ründajale infot, et andmebaasis võib olla midagi valest konfigureeritud ja samuti infot, millises kaustas viga võib leiduda.

Samuti sai testimisel jälgitud, kas veebilehel on sisenemine/registreerumine ning kas on ka sel juhul rakendatud krüpteeritud ühendus (https). Kokku avastas autor 12 veebilehte, millel oli sisenemine/registreerumine ning ei kasutatud krüpteeritud liiklust. Kahjuks see võimaldab pahalasel soovi korral kätte saada mitte temale mõeldud andmeid.

Analüüsist saadud tulemuse kõige suurema osakaalu moodustas mittekrüpteeritud ühendus veebilehtedel (vt. Illustratsioon 14).



Illustratsioon 14: Http probleeme esines seitsmekümne ühel kooli, Wordpressi vanat versiooni kasutas kaksteist kooli, Joomla vanat versiooni kasutas 13 kooli, Drupali vanat versiooni kasutas üks kool (autori koostatud)

Tulemus võib samuti olla isegi veel kehvem, kuna skänner ei tuvastanud veebilehti, mille sisuhaldussüsteem oli paar versiooni vanad. Käesolevat skänneeringut tehes, andis näiteks skänner märku Wordpressi lehtedele, mis oli alla 4.8 versiooni ehk mida ei oldud pärast 8-ndat juunit uuendatud. Käesoleva töö kirjutamise ajaks oli väljas juba versioon 4.9.1 Wordpressist (29.november 2017). Wordpressi versioonide 4.8 ja 4.9.1 vahele on jäänud 3 turvauuendusega versiooni ehk võib järeldada, et tulemus võib olla isegi veel kehvem.

3.4 Soovitused

Enamik probleemseid veebilehti on ehitatud mõnel sisuhaldussüsteemil. Koolide veebilehti analüüsides sai autor aru, et enamjaolt kuvatakse seal infot ning puuduvad suuremad funktsionaalsused, mis teeb sisuhaldussüsteemil tarkvara uuendamise küllaltki murevabaks.

Autor leiab, et kuna Eestit peetakse e-riigiks, siis võib ka ühendada arvutitunni enda kooli kodulehe uuendamiseks. Tänu sellele kasvab juba ka nooremate õpilaste teadlikkus küberturvalisusest ja huvi IT õppimise vastu. Tänapäeval on suur puudus IT töötajatest ja see võib olla hea reaalelul põhinev näide õpilastele sellest valdkonnast ja küberturvalisusest üldiselt.

Autor arvab, et kuna tegemist on haridusasutustega ja lisaks liitunud suure programmiga, siis on vajalik hoida ka kodulehed korras ja nõuetekohased. See võib aidata ka kooli mainele kaasa.

Kokkuvõte

Käesoleva bakalaureusetöö peamiseks eesmärgiks oli tutvustada rünnakute eesmärke, populaarsemaid ründevõimalusi veebilehtede vastu ja analüüsida Eesti haridusasutuste hetkeseisu, mille abil oleks võimalik ära hoida paljusi rünnakuid.

Töö käigus toodi välja peamised vead, kus kaudu rünnakut saab teha ja nende põhjal analüüsiti Eesti haridusasutuste veebilehtede hetkeseisu.

Tööd kirjutades selgus, et väga raske on tuua välja peamised soovitused ründekohtade vältimiseks, kuna veebilehte arendades ja hooldades on väga palju erinevaid pisiasju, mis võivad jätta tagaukse ründajale. Sellegi poolest loodab autor, et on toonud välja suuremad ja peamised tähelepanekud veebilehel turvalisuse jälgimiseks.

Eesti haridusasutuste veebilehtede analüüsi tulemus oli autori jaoks üllatav. Avastati 37% veebilehtedest, mida ei oldud hooldatud ja nende seas oli ka juba nakatanud veebilehti, mis võivad omakorda kasutaja või külastaja arvuteid nakatada. Samuti oli väga väike osakaal veebilehtedes, kus oli sisenemine/registreerumine ja oli järgitud krüpteeritud andmevahetust (https).

Analüüsist saadud tulemus sai edastatud Riigi Infosüsteemi Ameti infoturbeintsidentide käsitlemise osakonnale ning nende poolt saadud tagasiside põhjal on juba ühendust võetud Orissaare Gümnaasiumiga, kelle veebilehelt autor pahalase avastas.

Käesolev töö andis väga hea ülevaate, kui palju kohti tuleb veebilehte arendades või hooldades jälgida ja see omakorda võib seletada suuri numbreid küberrünnakutes.

Lõputöö andis autorile palju uusi teadmisi ja loodetavasti ka lugejale rohkem teadmisi sellest valdkonnas.

Kasutatud kirjandus

Kaspersky Distributor Lab (2014). Suurpüük: iga kolmanda õngitsemisrünnaku eesmärk on varastada raha. Loetud addressil https://www.antivirus.ee/et/news/suurpuuk-iga-kolmanda

Lyndon Sutherland (2016). Know Your Enemy: Understanding the Motivation Behind Cyberattacks. Loetud addressil https://securityintelligence.com/know-your-enemy-understanding-the-motivation-behind-cyberattacks/

Küberturvalisuse teenistus (2016). Riigi Infosüsteemi Ameti küberturvalisuse teenistuse 2016. aasta kokkuvõte. Loetud addressil https://www.ria.ee/public/Kuberturvalisus/RIA-kuberturbe-aastaraport-2016.pdf

Dan Moen (2016). How Attackers Gain Access to WordPress Sites. Loetud addressil https://www.wordfence.com/blog/2016/03/attackers-gain-access-wordpress-sites/

Darren Pauli (2017). Just give up: 123456 is still the world's most popular password. Loetud addressil http://www.theregister.co.uk/2017/01/16/123456_is_still_the_worlds_most_popular_password

CVE Details (kuupäev puudub). The ultimate security vulnerability datasource. Loetud addressil https://www.cvedetails.com/vulnerability-list/vendor_id-74/product_id-128/PHP-PHP.html

Liam Tung (2017). Google tightens noose on HTTP: Chrome to stick 'Not secure' on pages with search fields. Loetud addressil http://www.zdnet.com/article/google-tightens-noose-on-http-chrome-to-stick-not-secure-on-pages-with-search-fields/

Brett Rule (2017). Is your site considered secure by Google?. Loetud addressil https://www.netregistry.com.au/blog/site-considered-secure-by-google/

Autor puudub (2016). SQL Injection. Loetud addressil https://www.owasp.org/index.php/SQL_Injection

M.Jones (2014). Fight against SQL injection attacks. Loetud addressil

https://www.ibm.com/developerworks/library/se-sql-injection-attacks/index.html

Tartu Ülikool - Arvutiteaduse Instituut (2015). Javascript ründed. Loetud addressil https://courses.cs.ut.ee/2015/infoturve/spring/Main/OhudVeebis

Imperva Inc. (2017). CROSS SITE SCRIPTING (XSS) ATTACKS. Loetud addressil https://www.incapsula.com/web-application-security/cross-site-scripting-xss-attacks.html

autor puudub (2017). Unrestricted File Upload. Loetud addressil https://www.owasp.org/index.php/Unrestricted_File_Upload

Hari Charan (2017). Broken Authentication and Session Management – part I. Loetud addressil https://seleniumbycharan.wordpress.com/2017/07/30/broken-authentication-and-session-management-part-i/

Wordfence (2016). Password Authentication and Password Cracking. Loetud addressil https://www.wordfence.com/learn/how-passwords-work-and-cracking-passwords/

Team Mentor (kuupäev puudub). Information Disclosure Attack. Loetud addressil https://vulnerabilities.teammentor.net/article/8482159c-5ec2-4b89-9c65-9af765030ff5

Eslam Medhat (2017). Web Applications Attacks: Insecure Direct Object Reference. Loetud addressil https://latesthackingnews.com/2017/07/09/web-applications-attacks-insecure-direct-object-reference/

Stephen Moramarco (2017). OWASP Top 10 #5: Security Misconfiguration. Loetud addressil http://resources.infosecinstitute.com/owasp-top-10-5-security-misconfiguration/#gref

Software Guru (2012). OWASP Top 10 Web Application Vulnerabilities. Loetud addressil https://www.slideshare.net/RevistaSG/owasp-top-10-web-application-vulnerabilities

Ettevõtlik kool (2016). ETTEVÕTLIK KOOL. Loetud addressil http://evkool.ee/ettevotlik-kool/

Summary

The main purpose of this bachelor's thesis was to introduce goals of attacking, the most popular possibilities to attack website and analyze the current state of Estonian educational institutions, by which it would be possible to prevent many attacks.

The main mistakes were identified during the thesis, where the attack could be carried out and on this basis the current status of the websites of Estonian educational institutions was analyzed.

During the thesis it was clear that it is very difficult to outline the main recommendations to avoid attacks, because developing and maintaining the website has a many different meanings that can leave a backdoor to the attacker. However, the author hopes that the bachelor’s thesis highlighted the main observations of website security monitoring.

The result of the analysis of the websites of Estonian educational institutions was surprising for the author. 37% of the websites that were not maintained, and some of them were already infected and this in fact could infect the user or the visitor's computers. There was also a very small percentage of websites, were it was shown the login / registration field and the encrypted data exchange (https) was followed as well.

The result of this bachelor’s thesis was transmitted to computer emergency response team at the Information System Authority of Estonia and based on their feedback they already have been in contact with the Orissaare Gymnasium, where the author discovered a virus.

This work gave a very good idea of how many things you should notice, when it comes to developing or maintaining a website and this can explain itself the large number in cyber attacks.

The bachelor’s thesis gave to the author a lot of new knowledge and hopefully, more knowledge in this field to the reader as well.