TALLINNA ÜLIKOOL

INFORMAATIKA INSTITUUT

Mikk Lilles

POPULAARSETE VEEBI SISUHALDUS-SÜSTEEMIDE TURVALISUS

Seminaritöö

Juhendajad: Andrus Rinde

Tanel Toova

Tallinn 2014

Autorideklaratsioon:

Deklareerin, et käesolev seminaritöö 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)

Sisukord

Sissejuhatus

Tänapäeval on peaaegu kõik veebilehestikud dünaamilised, kasutavad andmebaase ja sisuhaldussüsteeme (CMS ehk *Content Management System*). Sisuhaldustarkvara valitakse peamiselt pakutavate võimaluste järgi, tihtipeale saab määravaks hind. Tarkvarade kirjeldustes rõhutatakse enamasti erilisi lisavõimalusi ning kasutamise mugavust ja lihtsust. Samas ei käsitleta peaaegu kunagi nende süsteemide turvalisust, mis võiks olla valiku tegemisel üheks kõige olulisemaks aspektiks.

Käesoleva seminaritöö teema on valitud autori huvist veebilahenduste ja sisuhaldustarkvarade vastu. Valikut toetab ka autori amet, mis on selle valdkonnaga seotud.

Seminaritöö teemat valides tutvus autor varem kirjutatud töödega ning avastas, et sisuhaldustarkvarade turvalisuse temaatikat ei ole varem lõputöödes käsitletud. Arvestades seda, et aina enam veebilehti ehitatakse üles eri sisuhaldustarkvaradele, on teema vägagi aktuaalne ning vajalik nii arendajatele, veebitoimetajatele kui ka teistele teema vastu huvi tundjatele.

Töö eesmärgiks on anda ülevaade tüüpilistest turvariskidest, erinevatest rünnakutüüpidest veebilehestike vastu ning sellest, kuivõrd need on aktuaalsed kõige populaarsemate vabavaraliste CMS-ide puhul. Üheks eesmärgiks on anda autori poolt hinnang, milline populaarsetest sisuhaldustarkvaradest võiks olla kõige turvalisem, ja mis võimalusi on rünnakute vältimiseks.

Eesmärkide saavutamiseks annab autor kirjanduse põhjal ülevaate tüüpilistest turvariskidest, valib välja kõige populaarsemad CMS-id ning selgitab kirjanduse ja rünnakute statistika alusel, millised on peamised nendega seotud riskid. Lõpuks annab autor kirjandusele toetudes hinnangu, milline on tema arvates kõige turvalisem valimisse võetud sisuhaldussüsteem.

1. Töös kasutatavad mõisted ja lühendid

Sisuhaldustarkvara (tarkvara, mis haldab dokumente veebisaitide tarvis) **–** CMS *(content management system*)

Kujundusteema (sisuhaldustarkvaradele mõeldud kujunduspakk) **–** *theme*

Nõrkus (ohtude realiseerimist võimaldav infovarade nõrk koht) – *vulnerability*

Programmiviga – *bug*

Süstimine (valideerimata info sisestamine kolmandate isikute poolt kasutaja serveris asuvatele failidele) – *injection*

Murdskriptimine (dünaamiliselt genereeritavate veebilehtede kasutamine võõrastesse arvutitesse tungimiseks) **–** *cross-site scripting* ehk XSS

OWASP (avaliku veebi liideste turvalisuse projekt) – *The Open Web Application Security Project*

CVE (keskkond, kus jagatakse informatsiooni ja statistikat levinud turvaaukude kohta) – *Common Vulnerabilities and Exposures*

Teenusetõkestamise rünne (arvuti või arvutivõrgu ülekoormamine samaaegselt suure hulga päringute saatmise teel) **–** *Denial of Service* ehk *DoS attack*

Omavoliline koodi käivitamine (ründaja võimalus käivitada igasugust koodi sihtarvutis või sihtprotsessis) – *execute code*

Mööda pääsema (ründaja pääseb mööda kasutaja poolt määratud turvatõketest, näiteks autoriseeringu päringust) – *bypass something*

Informatsiooni kättesaamine (privaatsete andmete omistamine kolmandate isikute poolt) – *gain information*

Lubamatu kasutajaõiguste eskaleerimine (ründajal on kasutajaga samad õigused serveris) – *gain privilege*

Lubamatu faili lisamine (see võimaldab ründajal lisada faile serverisse, mis käib tavaliselt läbi kasutaja veebiserveri skripti) – *file inclusion*

Kataloogitee läbimine (ründaja käsib veebirakendusel avada faili, mis peaks olema juurdepääsmatu ning mille ligipääsuks läbitakse kataloogipuu) – *directory traversal*

HTTP vastuse poolitamine (veebirakendus või keskkond ei suuda korralikult kontrollida sisendväärtusi. Ründaja saab määrata suvalisi päiseid, kontrollida keha osa või moodustada mitu eraldi HTTP vastust.) – *HTTP responce splitting*

Ristpäringuvõltsing (ründaja saab ära kasutada veebisaidile usaldatud õigusi veebilehitseja poolt) – *Cross-site request forgery* ehk CSRF

2. Ründed veebilehestike vastu

Selles peatükis antakse ülevaade tüüpilistest rünnakuviisidest veebilehestike vastu ja tuuakse Eesti veebilehtede põhjal näiteid, kuidas neid rünnakuviise kasutatud on.

2.1. Levinud rünnakumeetodid

Igasuguse tarkvaraga võivad kaasneda suured turvariskid. Selles seminaritöös uuritakse, mis on populaarsete sisuhaldussüsteemide kõige enam levinud turvariskid, missuguseid rünnakuid on nendele vabavaralistele sisuhaldustarkvaradele tehtud ja mis võimalusi on rünnakute vältimiseks. Järgnevalt annab autor ülevaate tüüpilistest turvariskidest, mis võivad eri CMS-del esineda.

Veebilehtede vastu suunatud rünnakumeetodeid on väga erinevaid ja tulemus sõltub alati eesmärgist. Rünnaku eesmärgiks võib olla lihtsalt oma pingete maandamine kellegi veebilehte rikkudes või tuntuse kogumine sellega tegelevate inimeste ringis. Võib juhtuda, et sellised ründed ei ole väga tõsiste tagajärgedega ja piirdub ainult peidetud informatsiooni mingis mahus lugemisega. Väga tihti on aga tagajärjed tõsised ning ettearvamatud, mille üks näide on ründaja poolt sisestatud teksti välja kuvamine kasutaja veebisaidil. Väga ohtlike tagajärgedega võivad olla ka sellised rünnakud, mille eesmärgiks on isikuandmete kogumine – näiteks kontaktandmete müümine ettevõtetele, kes tegelevad massreklaamide saatmisega või telefonimüügiga. Kõige rängemad tagajärjed võivad olla aga sellisel juhul, kui näiteks pääsetakse ligi inimeste internetipanga paroolidele või väga isiklikele andmetele. Näiteks portaal www.eesti.ee peab olema eriti hästi turvatud, sest see sisaldab kõigi kodanike andmeid haigekassa, töösuhte, hariduse ja muu kohta. Samuti on väga ohtlikud sellised rünnakud, mille läbi halvatakse terve laialt kasutatava süsteemi töö, näiteks häiritakse internetipanga või kaarditehingute toimimist.

Veebirakenduste turvalisuse teemal püüab tõsta inimeste teadlikkust OWASPi programm „Top Ten" (OWASP 2013a). Sihtasutus OWASP (The Open Web Application Security Project – avaliku veebi liideste turvalisuse projekt) alustas tööd 2001. aastal. See on mittetulunduslik rahvusvaheline organisatsioon, mis tegutseb annetuste toel. Nad on määratlenud kõige kriitilisemad turvaprobleemid. (OWASP 2013a)

Sihtasutuse OWASP hinnangul on CMS-ide kõige kriitilisemad turvaprobleemid:

- **Süstimine** (*Injection*)

  Kõige tavalisem süstimise vorm on SQL-süstimine. See on rünnak andmebaasikihi tasandil, mis seisneb SQL-päringusse omavolilise informatsiooni sisestamisel (Lehesalu 2007). Ründajad kasutavad seda väga tihti ja see on üks kõige ohtlikumaid ründemeetodeid, mida saab veebilehtede vastu sooritada.

- **Vigane autentimine ja sessiooni haldamine** (*Broken Authentication and Session Management*)

  Autentimise ja sessiooni haldamise alla käivad kõik autentimise ja aktiivsete sessioonidega seotud toimingud. Autentimine on väga oluline protsess ja kui selle toimingud – näiteks parooli vahetus, parooli unustamise korral uuesti saatmine, parooli meeldejätmine, konto uuendus ja teised sarnased tegevused – on vigased, siis saab seda väga tõsiste tagajärgedega ära kasutada (OWASP 2013b).

- **Murdskriptimine** (*Cross-Site Scripting – XSS*)

  Selle tehnika alus seisneb ühe serverarvuti mõjutamises, mille tulemusena server edastab oma kasutajale ründaja etteantud sõnumi. Sõnum võib olla nii kuuldelis-vaateline kui ka peidetud instruktsioon, mille käigus kasutaja arvuti teeb midagi ettearvamatut. (Wikipedia 2013)

- **Turvamata otseviide objektile** (*Insecure Direct Object References)*

  See on rünnakumeetod, mis tähendab seda, et viide mingile objektile on valideerimata ja ründaja saab sisestada endale meelepärase info. Näiteks väga levinud on URL-i muutmine, kus kasutaja andmete viide suunab hoopis teisele lehele. (OWASP 2013d)

- **Vigane turvakonfiguratsioon** (*Security Misconfiguration*)

  Selle nõrkuse väga hea näide on see, kui serveris installitakse automaatselt rakenduse administreerimise liides ja kasutaja ei kustuta seda installimise lõppedes ära. Ründaja avastab, et serveris on alles üldadministreerimise failid, millele saab ligi vaikimisi määratud paroolidega ning nüüd tekib ründajal võimalus serverit kontrollida (OWASP 2013f).

- **Privaatandmete kättesaadavus** (*Sensitive Data Exposure*)

  Tavaliselt salvestatakse IT-süsteemides kasutajate andmed, näiteks salasõnad, krediitkaardi andmed, aadress jm, andmebaasidesse. Kui see süsteem ei ole piisavalt kaitstud, siis on suur tõenäosus, et keegi saaks seda pahatahtlikult ära kasutada ja selle informatsiooni varastada.

- **Pääsuõiguste reguleerimine** (*Missing Function Level Access Control*)

  Tarkvarad ei kaitse alati oma funktsioone korralikult. Mõnikord määratakse funktsiooni kaitse konfiguratsioonis, mille seaded on aga valed. Arendajad peaksid lisama sobiva koodi, et kontrollida seadeid, aga unustavad tihti seda teha (OWASP 2013g).

- **Ristpäringuvõltsing** *(Cross-Site Request Forgery – CSRF*)

  See on rünne, kus lõppkasutaja sunnitakse käivitama soovimatuid programme, kus kasutaja on autenditud. Vähese abiga (saates lingi e-postile või vestlusesse) saab ründaja kasutaja ära petta, et ta avaks ja käivitaks programme, mida pahaline soovib. Kui lõppkasutajal on administraatori õigused, siis võib ohus olla terve veebitarkvara (OWASP 2013c).

- **Tuntud nõrkade tarkvarakomponentide ärakasutamine** (*Using Known Vulnerable Components*)

  Põhimõtteliselt on kõikidel rakendustel selline oht, sest enamik arendajad ei hoia oma komponente uuendatuna. Väga tihti arendajad ei teagi, mis komponente nad kasutavad ja kui need on omavahel sõltuvuses, siis see teeb asja hullemaks (OWASP 2013h).

- **Valideerimata ümber- ja edasisuunamine** (*Unvalidated Redirects and Forwards*)

  Väga tihti suunab tarkvara kasutaja teistele veebilehtedele või kasutavad siseviiteid sarnasel kujul. Mõnikord see veebileht kuhu kasutaja suunatakse on täpsustatud valideerimata parameetriga, mis lubab ründajal valida lehe, kuhu kasutaja suunatakse. Selliste ümbersuunamiste tulemusel võidakse proovida installida pahavara või üritatakse kasutajat panna sisestama salasõnu või muud privaatset informatsiooni.

  (OWASP 2013a)

Kõik OWASP-i poolt välja toodud kümme rünnakutüüpi on väga sagedased ja tihti, kui kasutajad leiavad viisi selliste rünnakute kaitseks, arendavad häkkerid välja uued meetodid kuidas neid kaitsemehhanisme nõrgestada.

Samuti on väga levinud turvaohuks omavoliline koodi käivitamine, kuigi seda 2013. aasta OWASP-i aruandes ei olnud. See on väga ohtlik programmiviga, kuna see võimaldab ründajal üle võtta terve vigase protsessi. Sellest edasi saab pahaline üle võtta masina, mille peal see protsess töötab. Kuna see on üks kõige tulemuslikumatest rünnakumeetoditest pahatahtlikel eesmärkidel, siis peaks selle ennetamine olema väga tähtis ülesanne veebilehe administraatorite poolt.

1999. aastal loodi CVE (Common Vulnerabilities and Exposures), mille eesmärgiks oli jagada informatsiooni turvalisuse nõrkustest ja privaatsete andmete pahatahtliku kättesaadavuse kohta. CVE kogub ja kategoriseerib info erinevatest selle valdkonna andmebaasidest ja kuvab andmed süstematiseeritud kujul kasutajatele. Uus sissekanne algab potentsiaalse turvaugu või süsteemi nõrkuse leidmisega. Omistatud informatsioonile antakse CVE identifikaator ja pannakse veebilehele üles. (CVE 2013b) CVE kodulehelt saab väga täpset ja aktuaalset infot sisuhaldustarkvarade turvaaukude ja nõrkuste kohta.

2.2. Rünnakud Eesti veebilehtede vastu

Eesti veebilehtede vastu toimub rünnakuid nagu kõikjal mujal maailmas ning ründemeetodid on üldjuhul just need, mida eelmises alapeatükis kirjeldati. Viimasel ajal on rünnakud ka meedias palju kõlapinda leidnud, eriti riigiasutuste veebilehtede vastu. Põhjus, miks just riigiasutused on levinud sihtmärgiks, võib olla selles, et rünnakute sõnumid on tihti poliitilised ja teatakse, et sõnum levib meedias paremini.

Järgnevalt mõned näited viimase aja rünnakute kohta.

Üheks viimase aja kõige enam tähelepanu pälvinud näiteks on 2013. aasta novembris Eesti Kaitseministeeriumi lehe ründamine. Majanduslehe E24 toimetaja Kalev Aasmäe andmetel oli häkkerite rühmitus Anonymous Ukraine teatanud, et plaanib rünnata Eesti kaitseministeeriumi kodulehte ning lehel tekkis tõepoolest probleeme. Ründe sõnum oli poliitiline, nimelt NATO-vastane. (Aasmäe 2013b)

Vaid mõned päevad hiljem rünnati Eesti Kunstimuuseumi kodulehte. Ründaja jättis kodulehele teate: „Vabandust, administraator, aga teie kodulehe turvalisus on madal. Palun lappige oma süsteem ära”. (Aasmäe 2013a)

ERR-i ajakirjaniku Maarja Rooni (2013) sõnul oli nende kahe veebilehe ründamise viis erinev – „kaitseministeeriumi leht koormati päringutega üle, kunstimuuseumil otsiti aga süsteemi nõrku kohti, et muuta lehe sisu". Ta tõdeb, et ülekoormamise tõrjumine on üsna keeruline, kuid kunstimuuseumi rünnakut oleks olnud tõepoolest lehe haldajatel võimalik lihtsasti ennetada.

„Mis puudutab veebilehtede näotustamist ehk soovimatu sisu üles riputamist, nagu juhtus kunstimuuseumi puhul, siis ka selliste rünnete korraldamine pole ülearu keeruline, sest reeglina kasutavad ründajad ära veebilehe haavatavusi, mis tulenevad uuendamata tarkvarast, kehvast seadistusest või halduspraktikast," rääkis RIA infoturbeekspert Triin Nigul. (Roon 2013)

„Taoliste ründajate elu saavad veebihaldurid raskemaks teha, uuendades regulaarselt veebilehe sisuhaldustarkvara (näiteks Joomla, WordPress) ja kasutades veebilehe haldamisel mõistlikke turvameetmeid," ütles Triin Nigul. (Roon 2013)

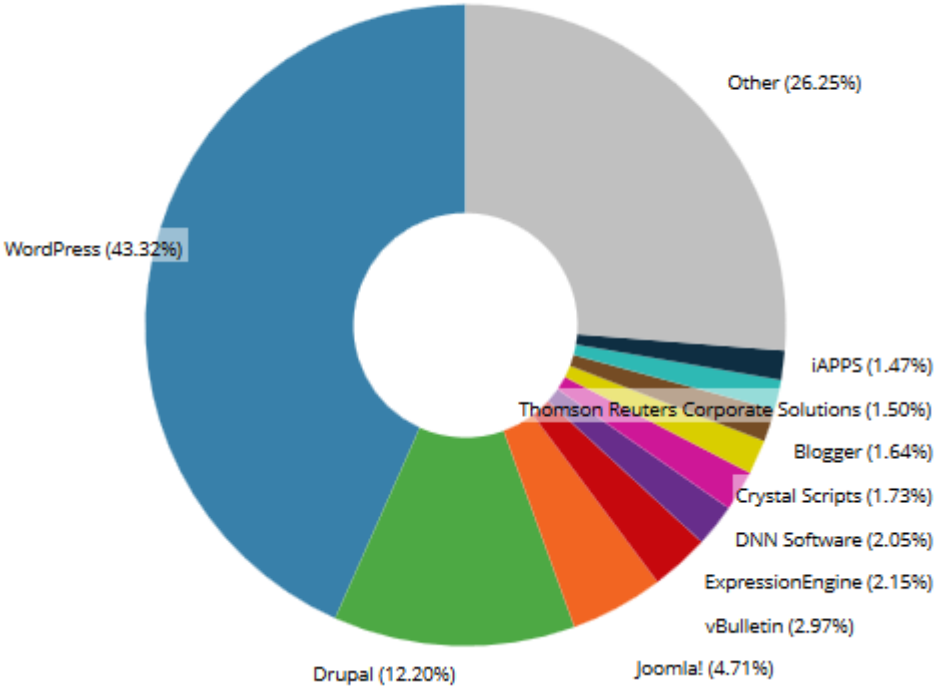
Viimastel aastatel on sarnaseid juhtumeid olnud veel palju, nende seas valitsuse koduleht, Eesti.ee ning Keskerakonna ja IRL-i kodulehed 2012. aastal (Kahu 2013), Tartu Maratoni veebileht 2010. aastal (Jüriso 2013), Tõrva linna ja Õru valla koduleheküljed samuti 2010. aastal (Väikenurm 2013), Põhjaranniku veebileht 2007. aastal (Rannajõe 2013).

Riigiasutused võivad saada rünnete sihtmärgiks ka näiteks väheste finantsvõimaluste tõttu – puudub võimalus palgata kompetentset IT-spetsialisti, kes oskaks turvaprobleeme ennetada, ja kasutatakse vabavaralist tarkvara. Näiteks Ivari Horm kaitses 2012. aastal Tallinna Tehnikaülikoolis bakalaureusetöö „Vaba tarkvara kasutamine võrgu teenuste osutamisel Tallinna munitsipaalkoolides". Tema uurimusest selgub, et enamik koole kasutavad koduleheküljede haldamiseks tõesti just vabavaralist tarkvara. Hormi (2012, 41) sõnul „61% ehk 19 küsitluses osalenud kooli kinnitas, et nende koduleheküljel kasutab vabal tarkvaral põhinevat administreerimiskeskkonda (WordPress, Joomla!, Drupal vms.)".

3. Populaarsemad CMS-id ja nende turvalisus

Autori seminaritöö käigus uuritakse kui palju ja missuguseid turvalisusega seotud probleeme on populaarsete sisuhaldustarkvarade puhul täheldatud ja püütakse anda hinnang, milline neist on kõige turvalisem.

Seminaritöös käsitlemiseks otsustas autor valida kõige populaarsemad tasuta CMS-id. Valiku tegemisel lähtub autor ühe väga autoriteetse statistika koguja Web Technology Surveysi (W3Techs n.d.) andmetest, mille järgi on kolm kõige enam kasutatavat sisuhaldustarkvara WordPress, Joomla ja Drupal. Need kolm on ka Builtwithi statistika (2013) järgi kõige populaarsemad. Nad teevad veebilehtede kasutuse kohta pidevat statistikat. Joonis 1 on näha graafik selle kohta, mis CMS-e miljoni kõige vaadatuma lehe seas kasutatakse (juhul kui kasutatakse CMS-i).

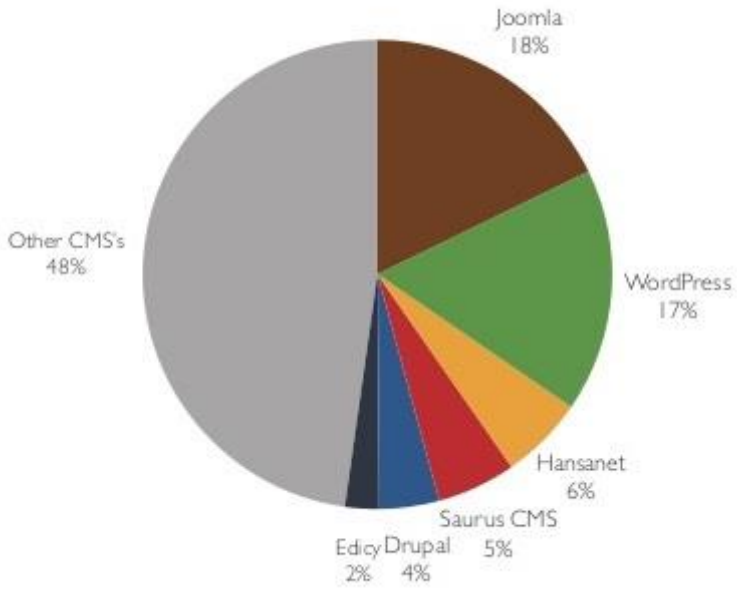


Joonis 1. Miljoni kõige enam vaadatava veebilehe CMS-i kasutus (Builtwith 2013)

Kuna käesoleva seminaritöö seisukohalt on oluline vaadelda kõige populaarsemaid CMS-e Eestis, siis lisaks toetub autor veel Mekaia poolt välja antud statistikale.

2013. aasta märtsis andis Eesti kõige suurem Drupali arendaja Mekaia meie riigi CMS-i turuosast ülevaate. Nad uurisid 23 000 .ee domeeniga kodulehte. CMS-i äratundmiseks kasutati selleks valmistatud skripti, mis arvab ära kasutatud tehnoloogia uurides HTML-i eri mustreid. 2013. aasta seisuga tunti ära 44% CMS-idest. 2009. aastal oli vabavaraliste CMS-ide osakaal võrreldes tasulistega 25% ning aastaks 2013 oli see kasvanud 50%-ni. (Mekaia 2013) Nende uuringus selgus, et kolm maailma populaarseimat CMS-i kuuluvad kõige populaarsemate hulka ka siin. Maailma populaarsuselt kolmas on Eestis küll viies, aga Mekaia tehtud uuringust selgub ka (Mekaia 2013), et Drupali osakaal on viimase aastaga jõudsalt tõusnud. Kolmandal kohal paiknev Hansanet on tasuline tarkvara ja seda autor valimisse ei võtnud.

Nendele kolmele lisaks võetakse vaatluse alla ka üks peamiselt vaid Eestis levinud tarkvara Saurus, mis on siin välja töötatud. Seda pakub ka üks suurimaid serveriteenuse pakkujaid Veebimajutus.



Joonis 2. Mekaia uuring eri CMS-ide Eesti turuosadest (Mekaia 2013)

3.1. WordPress

WordPress alustas tööd 2003. aastal. Sellest on kiiresti kasvanud üks populaarsemaid ise majandatavaid blogikeskkondi. WordPressi enda sõnul kasutab seda platvormi miljon veebilehestikku, millel on iga päev omakorda kümneid miljoneid külastajaid. See on loodud vabavaraliseks kasutamiseks.

3.1.1. WordPressi üldiseloomustus

Sellest blogikeskkonnast on kujunenud ulatuslik sisuhaldustarkvara, millel on tuhandeid pluginaid ehk pistikprogramme (*plugins*) ja vidinaid (*widgets*) ning kujundusteemasid (*themes*). WordPressi versioonide nimed tulenevad kuulsate *jazz*-muusikute järgi, nagu näiteks Mingus. (WordPress n.d.)

WordPressi tegemist alustati mõttega luua elegantne, hästi ülesehitatud personaalne avaldamise keskkond, mis on ülesehitatud PHP ja MySQL keeles ning toetub GPLv2 või hilisemale litsentsile. WordPress on b2/cafelogi järeltulija. Selle loomist alustati 2001. aastal. (WordPress n.d.)

3.1.2. WordPressi turvalisus

WordPress uuendab oma sisuhaldustarkvara keskmiselt kaks korda kuus. Uutes versioonides on tähtsal kohal turvaaukude parandamine ja kõrvaldamine. Kuna häkkerid arenevad kogu aeg ja õpivad järjest rohkem rünnatavaid süsteeme tundma, siis peavad sisuhaldustarkvarade arendusmeeskonnad süsteemi pidevalt täiustama ja proovima rünnakuid ennetada. Sellest tulenevalt on olemas palju versioone, käesoleva seminaritöö esitamise hetkel on uusim Parker, mis tehti avalikuks 12. detsembril 2013 (Mullenweg 2013).

Siit aga tulenebki WordPressi platvormile loodud veebilehestike suurim turvaprobleem – WordPressi tarkvara vajab serverites pidevat turvalisematele versioonidele uuendamist. Nii WordPressi kui ka teiste CMS-ide suurimaks turvariskiks on see, et häkkerid on loonud automatiseeritud vahendid, mis aitavad väga kiiresti leida erinevate veebilehtede turvaauke ja nõrkusi. Turvaaukude leidmist aitab väga oluliselt vähendada sisuhaldustarkvarade õigeaegne uuendamine.

2007. aasta mais tehtud uuringust selgus, et 98% WordPressi sisuhaldust kasutavates veebilehtedes on privaatne informatsioon, kas vähemal või rohkemal määral kättesaadav mitte selleks volitatud isikutele (Kierznowski 2013). Põhjuseks oli uuendamata või mitte enam toetatud WordPressi sisuhalduse versioon. 2008. aasta detsembris väljastati versioon 2.7, kuhu oli sisse ehitatud ühe-kliki uuendamise protsess, mis tegi kodulehe korraliku

haldamise palju lihtsamaks ja võimaldas uuendada oma tarkvara palju mugavamalt. (WordPress n.d.)

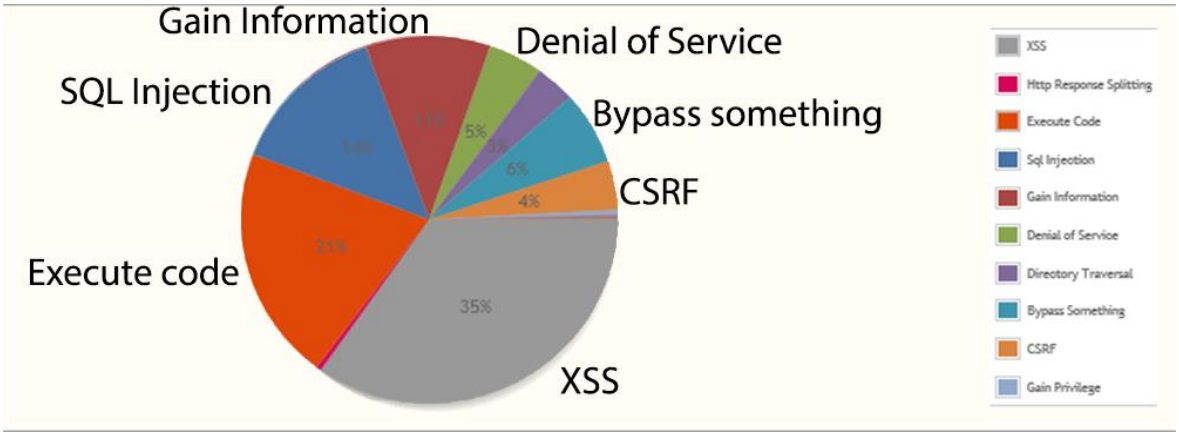
Mathew J. Schwartz (2013) kirjutas artikli WordPressi häkkimise kohta. Ta kirjeldab selles EnableSecurity tegevdirektori Sandro Gauci 2013. aasta septembris koostatud WordPressi turvalisuse kohta tehtud uuringut, milles osales 42 106 WordPressil tehtud veebilehte. Uuringust järeldati, et 73,2% WordPressi lehtedest on ohustatud automatiseeritud lehe nõrkasid kohti leidvate vahendite poolt. Selgus, et osalenud veebilehtede seas oli 74 eri WordPressi versiooni, millest 11 olid vigased (*invalid*) (nt versioon 6.6.6.). 18-l lehel oli vigane versioon, mida ei ole isegi enam olemas. 769 veebilehte (1,82% uuritutest) töötasid ikka veel WordPressi 2.0 vaheversioonil. Kõigest 7 814 (18,55%) lehte olid selleks ajaks uuendatud versioonile 3.6.1. 13 034 veebilehte (30,95%) töötasid ikka veel nõrgal WordPressi versioonil 3.6. Kõige enam kasutatavatest WordPressi versioonidest tehti ka kokkuvõte (vt tabel 1).

Tabel 1. WordPressi versioonide kasutamise statistika seisuga september 2013 (Abela 2013)

WordPressi versioon	Kasutajate arv	Tuntud riskide arv
3.6	13 034	5
3.6.1	7 814	0
3.5.1	6 859	8
3.5.2	4 031	0
3.4.2	2 204	12
3.5	1 655	10
3.3.1	820	24
3.2.1	820	10
3.3.2	732	14
3.4	295	15

WordPressi turvalisuse probleemid on välja toodud CVE kodulehel aastast 2004. Andmetest tuleb välja, et kõikide aastate ja versioonide peale kokku on kõige suurem probleem XSS-i ehk murdskriptimisega (vt Joonis 3), mis moodustab 35% kõikidest tuvastatud turvalisust puudutavatest probleemidest.

Teisel kohal on omavoliline koodi käivitamine ja kolmandal kohal privaatse informatsiooni kättesaamine kolmandate isikute poolt. Wikipedias on kirjas, et 2007. aasta alguses rünnati suuri blogilehti, mis olid optimeeritud otsingumootorite jaoks ja väiksemaid tavalisi blogikeskkondi, mille lehel kasutati reklaamide kuvamiseks AdSense'i programmi. Seoses nende uute veebirakenduste laialdase kasutusele võtmisega, levisid nendega kaasnenud turvaohud paljudele veebilehtedele, mis muutusid seetõttu kergeks sihtmärgiks.



Joonis 3. WordPressi nõrkuste statistika CVE järgi (CVE Details 2013d)

Mathew J. Schwartz toob artiklis „WordPress Site Hacks Continue" välja ühe väga olulise turvalisusega seotud aspekti, milleks on see, et selliseid sisuhaldustarkvarasid on väga lihtne kasutada, aga sellega kaasneb tihti olukord, et mitte ükski IT-alane professionaal ei vaata tarkvara installimist ega selle kasutamist üle. Probleeme võivad tekitada ka juba kõige lihtsamad valikud, nagu näiteks tugeva salasõna valimine. Administraatori kasutajanimeks näiteks „admin" valimine muudab lehe kohe väga heaks sihtmärgiks. (Schwartz 2013) Väga suur turvarisk seisnebki selles, et CMS-ide installimine on tehtud lihtsaks, mugavaks ning igaühele jõukohaseks.

Internetist võib leida palju vihjeid selle kohta, et WordPressil on olnud suuri probleeme rämpspostiga. Ühel lehel (Everyday Drupalist n.d.) kirjutatakse näiteks, et WordPressi 3.0 versiooni kasutajaks registreerimise vormil puudub valideerimine ja spämmi kaitse.

WordPressi vorm on nii ebakindel, et spämmirobotid (*spam bot*) suudavad luua tuhandeid libakasutajaid, mida kasutatakse spämmteadete ja lehele libasisu loomiseks.

3.2. Drupal

Drupal on vabavaraline paljude kohandamisvõimalustega tarkvarapakett, mille abil saab hõlpsasti infot organiseerida, hallata ja avaldada. Drupali tarkvara loomist alustati 1999. aastal sõnumikeskkonnana (*message board*). Juba umbes aastaga kasvas kasutajate huvi ja keskkond tehti vabavaraliseks. 2001. aastal jõudis Drupal.org veebi. (Drupal n.d.)

3.2.1. Drupali üldiseloomustus

Seda tarkvara hooldab umbes 630000 vabatahtlikku kasutajat ja arendajat. Seda arendatakse pidevalt edasi, et see platvorm toetaks kõige uuemaid tehnoloogiaid. Ka Drupali kodulehel on kirjas, et see on arendajate ja kodulehe valdajate eelistatuim. (Drupal n.d.)

Drupal ei uuenda oma versioone sama tihti kui WordPress. Viimane versioon on 2013. aasta detsembri seisuga 7.24 ja avaldati 2011. aasta alguses. Praegu on oodata uut 8.0 versiooni, aga selle avaldamise aega ei ole täpsustatud.

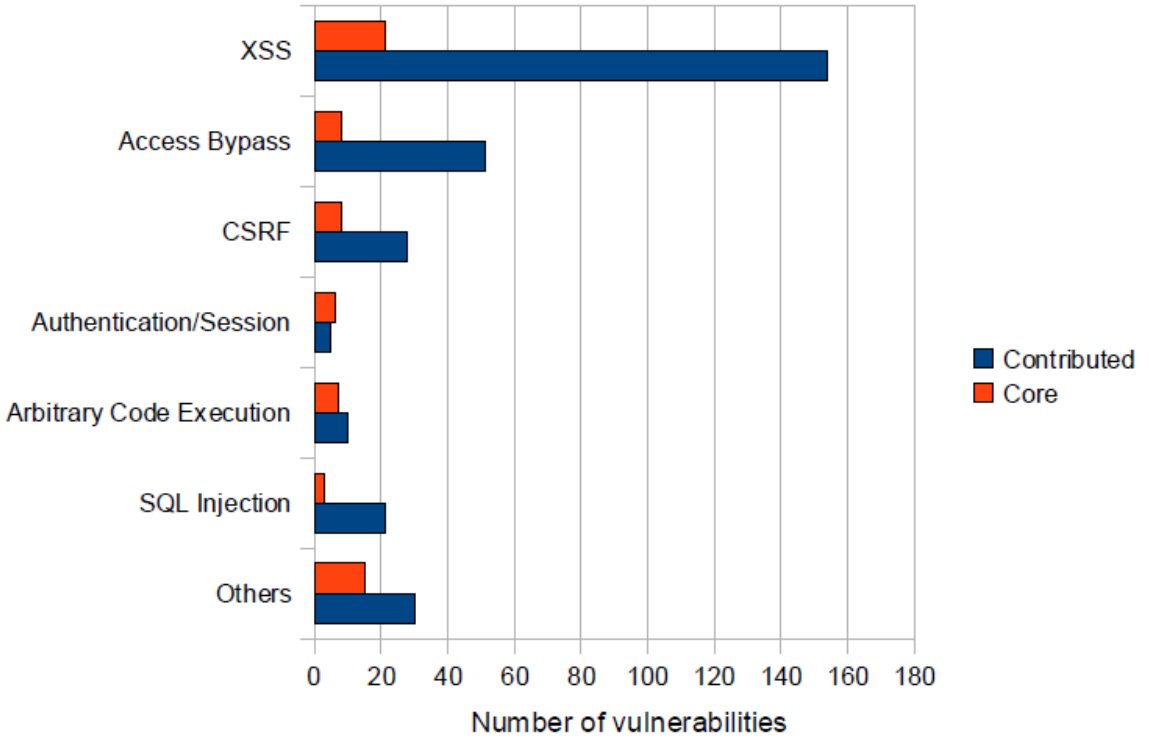
Drupalile on ehitatud isegi valitsusasutuste lehti, nende seas lausa whitehouse.gov ja data.gov.uk.

Drupali suureks eeliseks loetaks seda, et see talub väga suurt päringute hulka – nimelt 20000 päringut sekundis. (Frankk n.d.)

3.2.2. Drupali turvalisus

Drupali meeskond võtab OWASPi programmi turvariskide hinnanguid tõsiselt arvesse. Ka näiteks 2010. aasta turvaraportis on see välja toodud (Jeavons ja Knaddison 2010, 2). Raportis hinnatakse eraldi Drupali turvaprobleeme ning leitakse, et kõige enam levinud

nõrkus on XSS ehk murdskriptimine (vt Joonis 4). Sinisega on joonisel tähistatud probleemid Drupali tuuma koodis ja punasega vabatahtlike arendused.

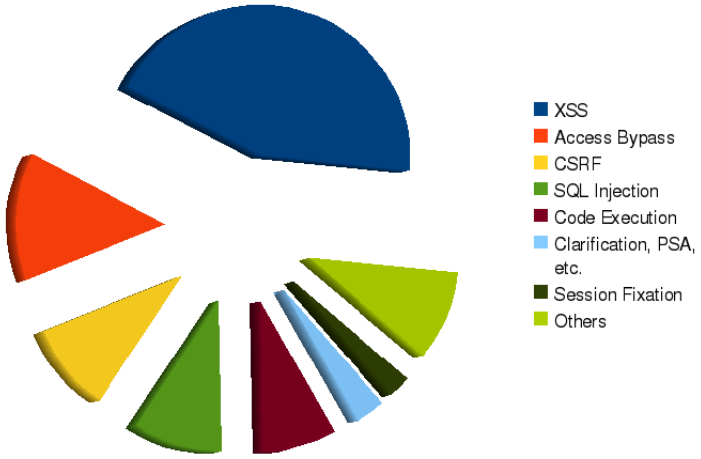


Joonis 4. Drupali turvaprobleemid (Jeavons ja Knaddison 2010, 4)

Raportis väidetakse, et Drupali tuum vaadatakse koodijupi haaval üle ja kõik peab muu seas vastama koodimis- ja turvastandarditele. Näiteks Drupal 7 puhul osaleb selles töös umbes 700 inimest, kellest vaid kahel on koodi üles laadimise luba (*commit access*).

Suuremad probleemid võivad esineda individuaalsete lehtede kohandatud koodiga. Drupali WhiteHat Security raporti kohaselt (Jeavons ja Knaddison 2010, 5) oli 90% 120st leitud nõrkusest seotud just kohandatud kujundusteemadega.

Ka Cracking Drupali (Knaddison n.d.) lehel jõuti tulemuseni, et kõige enam on levinud XSSi turvaprobleemid. Seal on tulemused toodud välja Drupali lehel olevate turvaprobleemide teadaannete analüüsi järgi (vt Joonis 5).

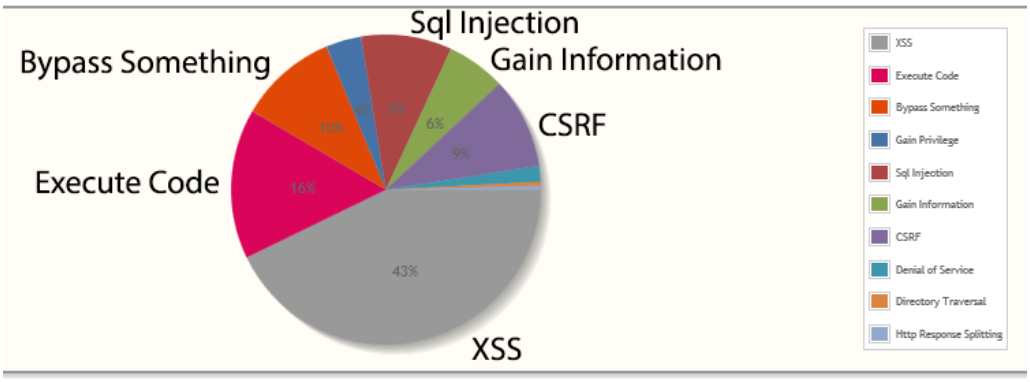


Joonis 5. Drupali turvaprobleemid Cracking Drupali analüüsi järgi (Knaddison n.d.)

Cameron & Wildingi vabavaraliste veebilahenduste kodulehel tutvustatakse DrupalConi konverentsil toimunud ettekannet, mis oli ehitatud üles just OWASPi hinnangule.

Konverentsil mainitud kõige kriitilisemate Drupali turvaprobleemide hulgas olid lisaks nendele, mis on kirjeldatud peatükis 2.1, puudulik krüptograafiline mälu (*Insecure cryptographic storage*), internetiaadressi piiramatus (*Failure to restrict URL Access*) ja puudulik failiedastus kaitse (*Insufficient transport protection*).

CVE andmebaasis oleva info põhjal (vt Joonis 6) on ka näha, et peaaegu poole kõikidest turvaaukudest moodustab murdskriptimine. Peamine põhjus, miks nii palju juhtumeid on seotud murdskriptimisega, on Drupali tuuma mooduli ebapiisav kontroll enne kuvamist erinevates rakendustes. Moodulite nimed ja kirjeldused, mis asuvad metaandmete failides (tuvastatakse nende .info laiendustega) on väga vähesel määral läbi vaadatud. Üks selline moodul on kontekstimoodul, mis võimaldab kasutajal jagada oma veebilehe erilaadseteks osadeks. Ründajad saavad sisestada suvalist koodi, et rünnata veebilehe administreerimise poolt. See võib muuta kasutaja konto turvalisust ja omakorda viia kolmandate isikute poolt käivitatud PHP-koodini või kasutatakse administraatori õigusi, et rünnata teisi süsteeme pahavaraga. (Klein Keane 2013)



Joonis 6. Drupali nõrkuste statistika CVE järgi (CVE 2013a)

Sama mooduli raames on tuvastatud ka möödapääsemine teatud turvapiirangutest. Selle põhjuseks on programmiviga kui Drupal käsitleb sessiooni informatsiooni. Seda saab ära kasutada, et välja kuvada teatud plokkide sisu, mis muidu on keelatud. Selleks kasutab ründaja spetsiaalset päringukoodi. Mööda pääsemine piirangutest moodustab ka arvestatava osa Drupali turvaaukudest. CVE andmetel moodustab see 10% kõikidest rünnakutest.

Versiooniga 7.24 parandati mitu kriitilist nõrkust Drupali süsteemis. Üheks paranduseks oli ristpäringuvõltsinguga seotud turvaauk. Drupalil on sisse ehitatud CSRF-i valideerimine, aga kuni selle versioonini tehti valideerimine eraldi moodulites vormi peal. Teatud juhtudel võib vormi valideerimise funktsioon käivitada ohtlikke operatsioone.

3.3. Joomla!

Joomla! (edaspidi Joomla) kodulehel on kirjas, et see on auhinnatud sisuhaldustarkvara, mille abil saab luua veebilehti ja võimsaid *online*-rakendusi. Oma lihtsuse ja laiaulatuslikkuse tõttu on see Joomla arendajate väitel üks kõige populaarsemaid veebilehetarkvarasid. See on samuti vabavaraline. Joomla tuuma uuendatakse keskmiselt kaks korda aastas ja 2013. aasta lõpuga olid toetatud versioonid 2.5 ja 3.2. (Joomla n.d.)

3.3.1. Joomla üldiseloomustus

Joomla arenes välja Mambo sisuhaldustarkvarast, mille viimane versioon ilmus 2008. aasta juunis. Joomla arendusmeeskond tahtis luua uut brändi, kuna Mambo arendajate arusaam

avatud lähtekoodiga programmist oli hoopis teistsugune nende omast. Joomla esimene versioon 1.0 anti välja 22. septembril 2005. aastal. (Joomla! 2008)

Süsteem sisaldab muuhulgas puhverdamist süsteemi kiiruse parandamiseks, RSS uudisvoogude loomist, lehekülgede prinditavaid versioone, blogivaadet, küsitlusi, veebilehelt otsimist. Süsteem on kirjutatud PHP skriptikeeles ja kasutab info salvestamiseks MySQL andmebaasisüsteemi.

3.3.2. Joomla turvalisus

Dean Marshall, üks juhtivaid Joomla eksperte ja arendajaid on oma ettevõtte, mis tegeleb igasuguse Joomla toega, lehel toonud välja väga levinud Joomla kasutamisega tekkinud turvalisusega seotud probleemid.

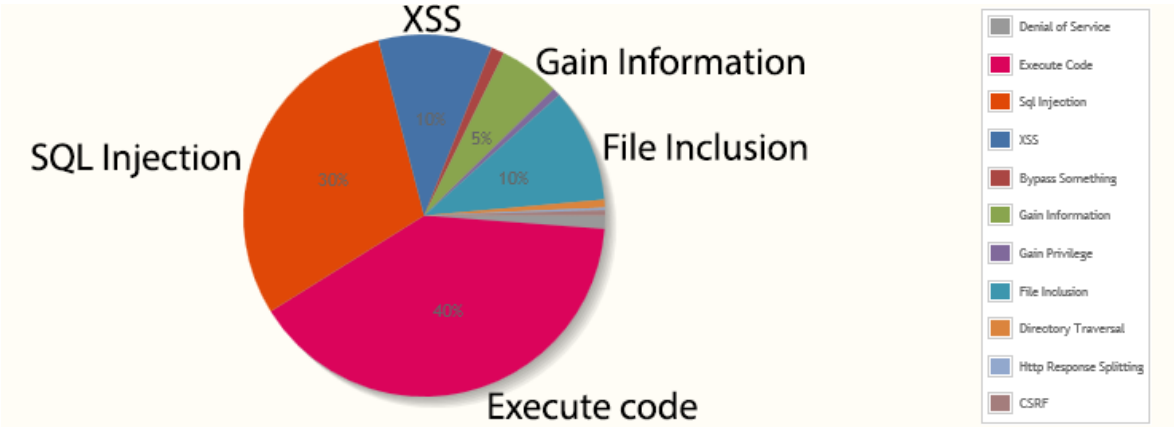
Nendeks on näiteks:

- Nõrk administraatori parool
- Paroolide teistega jagamine ja kättesaadavasse kohta jätmine
- Vananenud tuumfailid (*core files*)
- Odava majutusteenuse pakkujad jagavad oma serverit paljude klientide vahel, seega ei pruugi nad suuta tagada selle piisavalt head kiiruse ja turvalisuse konfiguratsiooni.
- Joomla laseb faile üles laadida ilma igasuguste piiranguteta. Kui keegi valdab foorumit või mingil muul põhjusel laseb teistel faile üles laadida, siis see tekitab turvaprobleeme, kuna kuritahtlik kasutaja võib üles laadida liiga suuri faile, mis nõrgestab kogu lehe tööd või mistahes tüüpi faile, mille avamise tagajärgi ei oska ette näha.
- Puudub korralik kontroll Joomla laienduste arenduse üle. (DM Consultancy n.d.)

Nendest esimesed kolm on inimfaktorid, mis tähendab, et suur osa probleemidest tuleb kasutajate oma tegemata jätmisest, teadmatusest või hoolimatusest. Samad probleemid on ka teiste sisuhaldustarkvaradega. Järgnevalt toon ma paar näidet tehnilistest turvaaukudest, mis on tuvastatud konkreetsetel Joomla versioonidel.

CVE kodulehel asuvalt graafikult (vt Joonis 7) on näha, et arvuliselt kõige rohkem rünnakuid on tehtud Joomlale omavolilise koodi käivitamise teel. See moodustab 40%

kõikidest rünnakutest aastate jooksul. Teisel kohal on SQL-süstimine, mille väga sagedane põhjus on see, et laienduspaki on teinud kolmandad osapooled, kes ei järgi nii rangeid turvareegleid kui Joomla enda arendajad. Üks hullemaid olukordi saab olla näiteks see, kui keegi süstib sellise koodi URL-i reale: http://www.minujoomlaveebileht.ee?userid=5′; DROP DATABASE minujoomlaandmebaasinimi– . See kustutab terve andmebaasi koos selle sisuga. Kuna URL-i lõpus on MySQL keele kommenteerimismärk, siis andmebaas ei anna veatõrget ja arvab, et kõik läks nii nagu pidi. (Itoctopus 2012)



Joonis 7. Joomla nõrkuste statistika CVE andmetel (CVE Details 2013a)

SecurityFocus kodulehel on välja toodud Joomla 3.1.5 ja sellele eelnevate versioonide turvalisuse probleem. Sisu failide PHP-koodis on viga, mis laseb murdskriptimise teel veebisaidile koodi lisada, mis saab hallata küpsiseid, sessiooni andmeid või muud informatsiooni, mille veebilehitseja on salvestanud selle veebilehe kohta. Vigane fail asub juurkaustas /libaries/idna_convert/example.php. (Pinna 2013)

JoomlaCode.org on veebiarendajatele mõeldud tuge pakkuv keskkond. Sellel kodulehel on kirjutatud Joomla 2.5.x versioonide puudusest. Kui sisuhalduse administraatori poole peal muuta üldkonfiguratsioone ja sisseloginud kasutaja on valinud, et veebilehitseja salvestaks tema kasutajanime ja parooli, siis automaatselt lisatakse need väärtused SMTP kasutajanime ja parooli väljadesse ning lihttekstina kirjutatakse need configuration.php faili. Häkkeril on pärast konfiguratsioonifailile ligipääsemist võimalus logida kasutaja administraatori poole peale sisse. (Jardin 2013)

Eelpool kirjeldatud juhtumitele sarnaseid rünnakuid on väga palju ja suure osa moodustavad just ründajate poolt kirjutatud koodi sisestamine ja süstimine veebilehele. Üks põhjus on kindlasti see, et need meetodid on väga tõhusad suuremahulise kahju tegemiseks.

3.4. Saurus

Saurus on tasuta, avatud lähtekoodiga, sisuhaldustarkvara veebisaitide loomiseks ja haldamiseks. Selle tarkvara loojad on eestlased ja nende eesmärk on olla edukas tarkvara eksportija ning viia sellega Eesti e-edu maailmakaardile. (Saurused OÜ 2013)

3.4.1. Sauruse üldiseloomustus

Saurusel, nagu ka teistel eelpool kirjeldatud sisuhaldustarkvaradel, on kõik vajalikud võimalused olemas veebilehtede loomiseks ja toimetamiseks, muude võimaluste hulgas *on-site* toimetamine, tekstiredaktor, failihaldus, kasutajate lisamine, mitmekeelsus, kommenteerimine ja palju muud. 2010. aastal tuli Saurus välja oma seni viimase stabiilse versiooniga (4.7.1) ja pärast seda on vigade ilmnemisel tehtud pidevalt sellele versioonile uuendusi. (Saurused OÜ 2013)

3.4.2. Sauruse turvalisus

Nagu teistelgi CMS-idel, esineb kahjuks ka Saurusel turvaprobleeme. Tuntud Taani ettevõte Secunia (2013), mis tegeleb IT turvalisuse ökosüsteemiga ja turvanõrkuste haldusega, on juhtinud tähelepanu Sauruse probleemidele.

Secunia kodulehel on toodud välja teiste seas Sauruse nõrkused ja turvaprobleemid, mida saab pahatahtlikult ära kasutada avalikustades privaatset informatsiooni ning pääseda läbi turvapiirangutest. Neid nõrkusi ära kasutades saab veebilehte murdskriptida, päringuid võltsida ja andmebaasi tahtmatut informatsiooni lisada või seda kustutada.

- Sauruse rakendus lubab kasutajal sooritada teatud tegevusi läbi HTTP päringute ilma, et neid valideeritakse. Seda saab ära kasutada, et vahetada administreerimise paroole, tühjendada prügikasti, seadeid muuta ja kustutada suvalisi vorme.

- Kui failide üleslaadimisel antakse neile ID, siis enne nende kasutamist veebilehel ei kontrollita sisendeid piisavalt.
- Kui sisend saadetakse läbi „sites" parameetri index.php lehele, siis ei kontrollita seda päringut piisavalt, enne kui sooritatakse SQL-päring. Seda viga saab ära kasutada sisestades vabalt valitud SQL-lauseid päringute sisse.
- Sisselogimisel, läbi „user" parameetri, ei kontrollita sisendit piisavalt. Seda ära kasutades, saab veebilehele sisestada igasugust HTML koodi ja skripte, mis avanevad sessiooni käigus kasutaja veebilehitsejas, kui sisestatakse nakatunud saidil privaatseid andmeid. Niimoodi saab isiklikke andmeid näha ainult siis kui „Lülita sisse saidi logi" väärtuseks on pandud „Jah"
- Sama probleem nagu eelmises punktis juhtub ka sisendi läbimisel „pg" parameetrist index.php lehele. See viga saab tekkida kui seadete all on aktiivseks tehtud „PHP ja SQL vead salvestatakse andmebaasi" link.
- Kui sisend edastatakse mitmele parameetrile paljudes skriptides, siis Saurus ei kontrolli sisendeid piisavalt enne kasutajale tagastamist. Selle tulemusena saab avada suvalist HTML-koodi ja skripti kasutaja veebilehitseja sessiooni ajal.
- Pärast sisendi andmist läbi GET-parameetri toimub ümbersuunamine redirect.php ja editor/redirect.php läbi. Saurus ei kontrolli piisavalt nende päringute õigsust ja tulemus võib olla see, et kasutaja suunatakse mingile teisele veebilehele.
- Saurus ei kontrolli piisavalt korralikult kasutaja õigusi kui soovitakse teada saada Sauruse installi seadeid – admin/check_requirements.php. Seda turvaauku saab ära kasutada teada saamaks kasutaja veebiserveri infot. (Secunia 2010)

Need loetletud probleemid on tuvastatud kõikidel 4.7 versioonidel 6. juuni seisuga 2013. Kui kasutaja uuendab oma sisuhalduse juulis 2013 välja antud viimase stabiilse versiooni peale, siis peaks olema kõik loetletud vead peale kahe viimase parandatud. Viimastele probleemidele ei ole praeguse ajani lahendust leitud.

Turvaauk murdskriptimisele on leitud admin/edit.php skriptis Sauruse 4.7.0 versioonis. Selle ära kasutamine eeldab, et kasutaja on sisselogitud ja omab „Article list" muutmise õigusi. Ründaja saab läbi parameetri „pealkiri" avada suvalist skripti või HTML-koodi kasutaja veebilehitsejas nakatunud lehel. Viimasele versioonile uuendamine lahendab selle probleemi. (CVE Details 2013a)

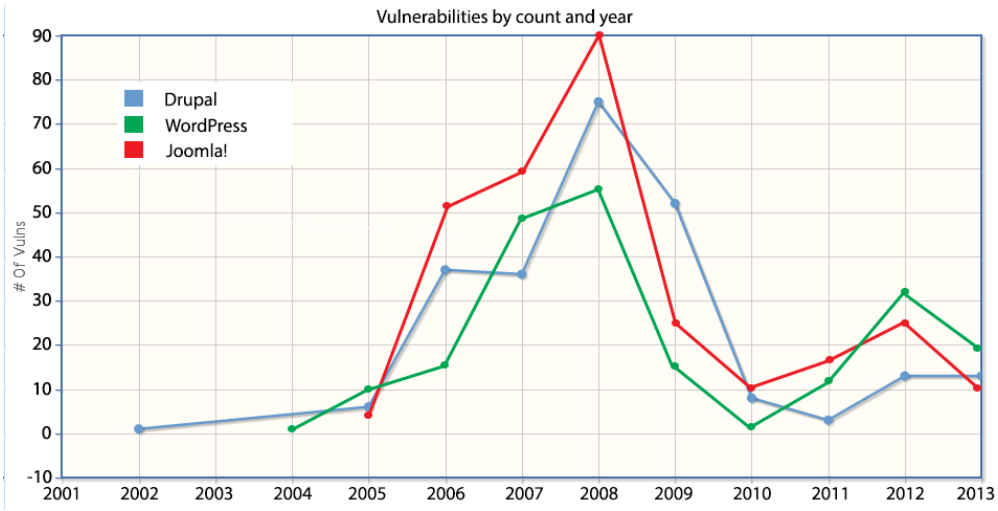
IBMi interneti turvasüsteemide kodulehel on välja toodud Sauruse sisuhaldussüsteemi turvarisk, mis seisneb mitme faili lisamisel veebilehele. Sisuhaldus lubaks pahatahtlikul ründajal sisestada suvalisi faile. Häkker võib saata erinevaid URL-i päringuid file.php ja com_del.php skriptidesse kasutades class_path parameetrit, et kasutada enda faili kaugvõrgust. See lubaks häkkeril käivitada suvalist koodi nakatunud veebi serveril.

See viga on tuvastatud Saurus 4.7.0 versioonil ja kuni 2013. aasta 1. detsembrini ei olnud sellele probleemile lahendust. (IBM Internet Security Systems 2013)

SecuriTeami kodulehel on kirjeldatud Saurus 4.7.1 versiooni viga, mis võib viia autoriseerimata informatsiooni avalikustamiseni. Probleem tekib siis kui ründaja saadab otsepäringu admin/templates.php, admin/edit_pilt.php või admin/edit_object.php skriptile. See avalikustab tarkvara installatsiooni failide asukohad. (Securiteam 2013)

3.5. Kokkuvõte vaadeldud CMS-ide turvalisusest

Kahjuks on kõikidel sisuhaldussüsteemidel turvaauke ja nõrkusi, millele ei ole veel lahendusi leitud. Saurus on veel maailma mastaabis suhteliselt vähe kasutatud ning sellest tingituna ei ole CVE-l selle kohta võrreldavat informatsiooni. Samas ei ole Saurusel sellist pluginate võimalust nagu teistel sisuhaldustarkvaradel ja seetõttu on leitud turvaaukude arv väiksem. Ülejäänutel sisuhaldussüsteemidel on aga suureks probleemitekitajaks just allalaaditavad pluginad ja kohandatavad kujundusteemad. Kui CMS-ide tuuma koodi on võimalik üsna hästi kontrolli all hoida, siis igasuguseid kujundusteemasid ja muud saavad kasutajad ise luua ja teistele jagada.



Joonis 8. CMS-ide rünnakud aastate lõikes (CVE Details 2013b)

Nagu graafikult näha, tõusis rünnakute arv 2008. aastal kõige enam Joomla lehtedel ja kõige vähem said kannatada WordPressi lehed, samas 2013. aastaks on WordPressi rünnakute arv siiski suurem kui Drupali ja Joomla oma. Drupali üldiselt heale turvalisusele viitab ka näiteks see, et sellele on ehitatud isegi suurte valitsusasutuste lehti, nende seas lausa whitehouse.gov ja data.gov.uk. Drupali lehte on raske ka päringutega üle koormata, sest see talub väga suurt päringute hulka – nimelt 20 000 päringut sekundis. (Frankk n.d.) WordPressi kasuks ei räägi ka pidev uute versioonide väljalaskmine, sest nendega on raske sammu pidada ja versioonid jäävad uuendamata. Uue versiooniga tuuakse aga vana versiooni vead välja WordPressi kodulehel ja nende teadmine muudab ründamise pahatahtlikele eriti lihtsaks.

Kui vaadata graafikut (vt Joonis 8) kõikide aastate peale kokku tuvastatud turvaprobleemidest, siis on märgata, et 2010. aastal oli väga vähe raporteeritud turvaauke. Arvatavasti näiteks WordPressi puhul on selle põhjuseks see, et nimetatud aastal ilmus ainult üks WordPressi versioon – 3.0. Selle versiooniga tuli WordPressi palju muudatusi nii sisu osas kui ka uue kujundusteema näol. Suurimaid muudatusi oli arvatavasti see, et enam ei olnud kahte erinevat sisuhaldussüsteemi – WordPress ja WordPress MU (*multiuser*), mis tähendas, et MU tarkvaraga sai mitu eraldiseisvat veebilehte luua, aga andmebaas on originaalsaidiga sama. Selle versiooniga sai WordPressist ühtne süsteem ehk *multisite*.

Seda, milline nendest neljast süsteemist kõige turvalisem võiks olla, on siiski raske hinnata. CVE statistika järgi (vt Joonis 8), kus on vaadeldud kõikide raporteeritud intsidentide arvu, võib näha, et 2008. aastal oli kõikide CMS-ide rünnakute arv tõusnud tippu. Seda võib põhjendada näiteks üldine majandusseis. Rahaliste raskuste tõttu võidi hakata veebilehti rohkem vabavaralistele CMS-idele ehitama. Samuti võis sellest tuleneda tahe omal käel hakkama saada ja mitte palgata asjatundjat. Mida rohkem veebilehti neile süsteemidele loodi, seda enam võisid selguda nõrkused.

2013. aasta statistikat vaadates (vt Joonis 8) on näha, et kõige enam turvaprobleeme on avastatud WordPressil. Teisel kohal on Drupal ning kolmandal Joomla. Siin võib märgata selget korrelatsiooni nende sisuhaldussüsteemide populaarsusega samal aastal. Nimelt Builtwithi statistika järgi (vt Joonis 1) oli just WordPress (43,32 %) kõige enam kasutatud CMS, sellele järgnes Drupal (12,20%) ning kolmas oli Joomla (4,71%).

Uurimisalustest sisuhaldustarkvaradest on raske valida, mis on kõige turvalisem. Neil kõigil on erinevaid turvalisusega seotud probleeme. Autor valiks kõige turvalisemaks Drupali, kuna lisaks eelmainitud omadustele on ka Drupal kirjutatud väga puhtas ja paindlikus koodis, mis teeb selle kasutamise mugavaks ja turvalisemaks.

4. Soovitused turvariskide vähendamiseks

Arvestades seda kui palju on 2013. aasta jooksul ka Eestis veebilehti rünnatud, on sisuhaldussüsteemide turvalisuse teema kindlasti aktuaalne. Paljusid rünnakuid saab tihtipeale ennetada. Järgnevalt antakse mõningaid soovitusi veebilehe haldajatele, kuidas riske vähendada. Soovitused põhinevad nii isiklikul kogemusel kui kirjandusel.

Mitmed õnnestunud rünnakud on tingitud inimfaktoritest. Veebilehe haldaja teadlik ja aktiivne tegutsemine aitaks riske vähendada. Üheks ohumärgiks võiks pidada näiteks seda, et vabavaralised CMS-id on kõigile lihtsasti kättesaadavad. Sellest võib tuleneda see, et installimist ja kasutamist ei kontrolli mitte ükski IT-alane professionaal, vaid lihtsalt asjast huvitatu. Tasuta kättesaadava CMS-i puhul võib kasutajale tunduda, et ega kaotada pole midagi ja proovib omal käel hakkama saada. IT-teenuseid peetakse üldjuhul väga kulukateks.

Inimfaktorite alla kuuluvad ka näiteks nõrga parooli valimine, samuti parooli jagamine ja teistele kättesaadavasse kohta jätmine. Samuti võiks inimfaktorite alla liigitada vead konfiguratsioonil, CMS-i turvalisemale versioonile uuendamise ja varunduse tegemata jätmise. Varunduse tegemata jätmist ei saa iseenesest pidada nõrkuseks, kuid see oluline faktor muudab lehe rünnakujärgse taastamise palju keerulisemaks. Nagu selgus WordPressi statistikast (Schwartz 2013), on versioonide uuendamata jätmine kahjuks väga levinud. Enamasti tulevad sellised hooletusest tingitud vead teadmatusest.

Siinkohal võikski juhtida tähelepanu sellele, kui oluline on tõsta selles valdkonnas inimeste teadlikkust. Enam ei kuulu veebilehtede haldamine ja turvalisuse tagamine vaid arendajate pädevusse, sellega tegelevad ka täiesti teiste valdkondade inimesed. Riigi Infosüsteemi Amet (RIA) peab veebilehtede seisukorra suurimaks probleemiks ebapädevaid arendajaid ja sisutoimetajaid. RIA tellitud uuringu kohaselt (Tohvelmann 2011, 13, 16) ei vasta lausa 95,91% 293st uuritud avaliku sektori veebilehtedest WCAG 2.0 kriteeriumite isegi miinimumnõuetele ning 87% HTML-kood on ebakorrektne.

Teiselt poolt tulenevad rünnakud tehnilistest faktoritest. Siia võiks lugeda seda, kui CMS-id on nõrkuste kaudu lihtsasti rünnatavad. WordPressi ja Drupali puhul on olnud kõige levinumaks ründetüübiks XSS ehk murdskriptimine (vastavalt 35% ja 43%). Siinkohal kordaks veelkord, et esimene samm kõikide nõrkuste ja ohtude minimaliseerimiseks oleks

oma sisuhaldustarkvara hoidmine uuendatuna ja teiseks käia tihti oma veebisaidil, et hoida silma peal kui peaks midagi korrast ära olema. Rääkides aga konkreetsemalt ründetüübist XSS, siis kõige efektiivsem, aga mitte lõplikult kindel moodus ennetada XSS rünnet, on lubada ainult JavaScripti, mis tuleb kasutaja poolt usaldatud saidilt. Soovitav on kasutada oma veebilehitsejate võimalusi ja lisasid, mis kaitsevad selliste rünnakute eest. Näiteks Firefoxil on olemas plugin nimega NoScript, mis lubab läbi ainult valideeritud koodi. Teistel populaarsetel veebilehitsejatel on ka sarnased lisad olemas. (NSA 2011)

Üks enim kasutatavaid ja ohtlikumaid ründemeetodeid on SQL-süstimine ja peamine eksimus, mida tehakse, et ründajad saaksid süstida SQL-koodi on see, et domeenil lubatakse kasutada kontrollimata ja valideerimata sisendeid. Selleks, et sellist rünnakut ära hoida, peaks rakendus näiteks kõik erilise tähendusega tähemärgid – nagu ühekordsed ja topelt jutumärgid, semikoolon, kommenteerimismärgid – ükshaaval valideerima.

Teine võimalus SQL-süstimise ärahoidmiseks on mitte kunagi liita kasutajate sisendit rakenduse SQL-lausega, mis saadetakse andmebaasi päringuks. Lihtne moodus selle saavutamiseks on kasutada parameetritega päringuid. Need päringud on sellised, kus SQL-i muutujate osad asendatakse määrajatega (tavaliselt „?”). Näiteks selle asemel, et päringut sooritada selliselt: SELECT email FROM users WHERE email = *'<user_input>'*, võiks sooritada nii: SELECT email FROM users WHERE email = *?*. (OWASP 2013e)

Joomla puhul on omavoliline koodi käivitamine (40%) aga kõige levinum (Kerner 2013). See võib tuleneda sellest, et enne 2011. aastat oli Joomla suurteks nõrkusteks pluginad. Kuna Joomla puhul ei olnud nende arendamine piisavate turvanõuetega, siis oli pluginate kaudu ründeid väga palju ning just nende abil on mugav kasutada koodi käivitamist ja SQL-süstimist. Siinkohal võibki tuua kõige paremaks lubamatu koodi käivitamise ründe ära hoidmiseks selle, et pluginate allalaadimisel ja enne CMS-iga liitmist, tuleks veenduda plugina päritolus ja sobivuses.

Kuid ükskõik, mis CMS-i poolt kasutaja otsustab, oleks põhiline, et ta vähendaks võimalikult palju inimfaktoritest tulenevaid turvariske ja oleks teadlik valitud sisuhaldustarkvara turvaaukudest ning muudest ohtudest.

Kokkuvõtlikult on kõige olulisemad soovitused CMS-i turvariskide vähendamiseks järgnevad:

- Veendu, et kasutad alati kõige uuemat CMS-i versiooni

- Kasuta tugevaid paroole, vaheta neid regulaarselt ning ära jaga neid teistega
- Konfigureeri veebisaidi seadeid korrektselt
- Konsulteeri veebilehe loomisel IT-alase professionaaliga
- Käi probleemide märkamiseks tihti oma veebisaidil
- Veendu pluginate päritolus ja sobivuses
- Veendu, et HTML-kood on korrektne
- Luba ainult JavaScripti, mis tuleb kasutaja poolt usaldatud saidilt
- Kasuta veebilehitsejate lisasid, mis lubavad läbi ainult valideeritud koodi
- Kasuta parameetritega määratletud SQL-päringuid
- Kasuta ainult valideeritud ja kontrollitud SQL-sisendeid
- Varunda regulaarselt serveris olevaid faile

Kokkuvõte

Veebilehtede turvalisus on viimasel ajal väga aktuaalne teema. Eesti meedias on leidnud aina enam kõlapinda rünnakud riigiasutuste, näiteks Eesti Kaitseministeeriumi, Eesti Kunstimuusemi ja paljude teiste veebilehtede vastu.

Käesoleva seminariöö eesmärgiks oli anda ülevaade tüüpilistest turvariskidest, erinevatest rünnakutüüpidest veebilehestike vastu ning sellest, kuivõrd need on aktuaalsed kõige populaarsemate vabavaraliste CMS-ide puhul. Üheks eesmärgiks on anda autori poolt hinnang, milline populaarsetest sisuhaldustarkvaradest võiks olla kõige turvalisem, ja mis võimalusi on rünnakute vältimiseks. Töö eesmärk sai täidetud.

Seminaritöös käsitlemiseks otsustas autor valida kõige populaarsemad tasuta CMS-id. Nende valimisel lähtus autor ühe väga autoriteetse statistika koguja Web Technology Surveysi (W3Techs n.d.) andmest, mille järgi on kolm kõige enam kasutatavat sisuhaldustarkvara WordPress, Joomla ja Drupal. Nendele kolmele lisaks võeti vaatluse alla ka üks Eestis levinud tarkvara Saurus, mis on siin välja töötatud.

CMS-ide turvariskid on seotud veebilehe haldajate hoolimatusest või teadmatusest tingitud inimfaktoritega (näiteks nõrkade paroolide valimine, tarkvara uuendama jätmine) ning tehniliste vigadega (valideerimata ümbersuunamised, SQL-sisendite vähene kontrollimine). Riske saab maandada järgides seminaritöö viimases peatükis välja toodud soovitusi.

Turvaprobleeme esineb kõikidel valitud CMS-idel ning keeruline on öelda, mis neist on kõige turvalisem. Autori hinnangul võib kõige turvalisemaks pidada Drupalit. Selle üldiselt heale turvalisusele viitab näiteks see, et sellele on ehitatud isegi valitsusasutuste lehti, nende seas lausa whitehouse.gov ja data.gov.uk. Drupali lehte on raske ka päringutega üle koormata, sest see talub teiste CMS-idega väga suurt päringute hulka.

Käesolev seminaritöö andis CMS-ide turvaprobleemidest üldise ülevaate. Autoril on plaan teemaga bakalaureusetööd kirjutades edasi tegeleda ning edaspidi uurida lähemalt töös käsitletud ründetüüpe ja nende mõju veebikeskonnale. Samuti oleks kasulik tegeleda põhjalikumalt rünnakute ennetamist võimaldavate tehniliste lahendusega.

Kasutatud kirjandus

Aasmäe, Kalev 2013a. Kunstimuuseum: veebilehe töö taastub turvalise lahenduse olemasolul. http://www.e24.ee/2587438/kunstimuuseum-veebilehe-too-taastub-turvalise-lahenduse-olemasolul (Accessed November 11, 2013).

Aasmäe, Kalev 2013b. RIA: oleme kaitseministeeriumi tabanud tehnilistest probleemidest teadlikud. http://www.e24.ee/2582224/ria-oleme-kaitseministeeriumi-tabanud-tehnilistest-probleemidest-teadlikud (Accessed November 11, 2013).

Abela, Robert 2013. Statistics Show Why WordPress is a Popular Hacker Target. http://www.wpwhitesecurity.com/wordpress-news/statistics-70-percent-wordpress-installations-vulnerable/ (Accessed August 11, 2013).

Builtwith 2013. CMS Usage Statistics. http://trends.builtwith.com/cms# (Accessed December 19, 2013).

CVE 2013a. Drupal: Security Vulnerabilities. http://www.cvedetails.com/vendor/1367/Drupal.html (Accessed November 11, 2013).

CVE 2013b. Frequently Asked Questions. http://cve.mitre.org/about/faqs.html (Accessed December 16, 2013).

CVE Details 2013a. Saurus: Security Vulnerabilities. http://www.cvedetails.com/vulnerability-list/vendor_id-10865/Saurus.html.

CVE Details 2013b. Vulnerability Distribution. http://www.cvedetails.com/ (Accessed December 10, 2013).

DM Consultancy The Most Commonly Known Joomla Security Issues. http://www.webdevelopmentconsultancy.com/joomla-security/common-security-issues.html?start=1 (Accessed November 27, 2013).

Drupal Drupali kirjeldus. https://drupal.org/about (Accessed November 9, 2013).

Everyday Drupalist Drupal versus Word Press - A brief security and usability comparison. http://www.wy-designs.net/content/drupal-versus-word-press-brief-security-and-usability-comparison (Accessed December 3, 2013).

Frankk, David Security Issues in Drupal Content Management System. http://www.examiner.com/article/security-issues-drupal-content-management-system (Accessed March 12, 2013).

Horm, Ivari 2012. *Vaba tarkvara kasutamine võrgu teenuste osutamisel Tallinna munitsipaalkoolides*. Tallinn: Tallinna Tehnikaülikool.

IBM Internet Security Systems 2013. Saurus CMS mutiple file include. http://xforce.iss.net/xforce/xfdb/61076 (Accessed December 14, 2013).

Itoctopus 2012. How to Prevent SQL Injection in Joomla. http://www.itoctopus.com/how-to-prevent-sql-injection-in-joomla (Accessed December 16, 2013).

Jardin, David 2013. Joomla! http://joomlacode.org/gf/project/joomla/tracker/?action=TrackerItemEdit&tracker_item_id=32369&start=0 (Accessed December 16, 2013).

Jeavons, Benjamin James ja Gregory James Knaddison 2010. *Drupal Security White Paper*. http://drupalsecurityreport.org/sites/drupalsecurityreport.org/files/drupal-security-white-paper-1-1.pdf (Accessed December 1, 2013).

Joomla Joomla kirjeldus. http://www.joomla.org/about-joomla.html (Accessed November 8, 2013).

Joomla! 2008. Migrating to Joomla! from Mambo. http://help.joomla.org/content/view/818/181/ (Accessed December 15, 2013).

Jüriso, Katrin 2013. Tartu Maratoni veebileht langes häkkerite rünnaku ohvriks. http://uudised.err.ee/index.php?06192672 (Accessed November 11, 2013).

Kahu, Oliver 2013. Eile õhtul rünnati nelja Eesti veebilehte. http://uudised.err.ee/index.php?06263316 (Accessed October 11, 2013).

Kerner, Sean Michael 2013. HP Security Report: What is the Most Insecure CMS? http://www.esecurityplanet.com/news/article.php/3929901/HP-Security-Report-What-is-the-Most-Insecure-CMS.htm (Accessed December 15, 2013).

Kierznowski, David 2013. Survey Finds Most WordPress Blogs Vulnerable. http://blogsecurity.net/wordpress/articles/article-230507/ (Accessed November 12, 2013).

Klein Keane, Justin C. 2013. Drupal core XSS vulnerability. http://seclists.org/fulldisclosure/2013/Aug/158 (Accessed December 17, 2013).

Knaddison, Greg James What Kinds of Security Problems Exist in Drupal? http://crackingdrupal.com/blog/greggles/what-kinds-security-problems-exist-drupal (Accessed December 1, 2013).

Lehesalu, Uku 2007. *PHP/SQL süstimine*. Tartu Ülikool. Matemaatika-informaatikateaduskond. Arvutiteaduse instituut.

Mekaia 2013. Estonian web CMS market overview in March 2013. http://www.saurus.info/estonian-web-cms-market-overview-in-march-2013/ (Accessed November 15, 2013).

Mullenweg, Matt 2013. WordPress 3.8 “Parker.” http://wordpress.org/news/2013/12/parker/ (Accessed December 11, 2013).

NSA 2011. Protect Against Cross Site Scripting (XSS) Attacks. http://www.nsa.gov/ia/_files/factsheets/xss_iad_factsheet_final_web.pdf (Accessed January 1, 2014).

OWASP 2013a. About The Open Web Application Security Project. https://www.owasp.org/index.php/About_OWASP (Accessed November 28, 2013).

OWASP 2013b. Broken Authentication and Session Management. https://www.owasp.org/index.php/Broken_Authentication_and_Session_Management (Accessed October 1, 2014).

OWASP 2013c. Cross-Site Request Forgery (CSRF). https://www.owasp.org/index.php/Cross-Site_Request_Forgery_%28CSRF%29 (Accessed January 10, 2014).

OWASP 2013d. OWASP Top Ten Project. https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project#tab=Project_Details (Accessed December 4, 2013).

OWASP 2013e. SQL Injection Prevention Cheat Sheet. https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet#Defense_Option_1:_Prepared_Statements_.28Parameterized_Queries.29 (Accessed December 25, 2013).

OWASP 2013f. Top 10 2013-A5-Security Misconfiguration. https://www.owasp.org/index.php/Top_10_2013-A5-Security_Misconfiguration (Accessed January 12, 2014).

OWASP 2013g. Top 10 2013-A7-Missing Function Level Access Control. https://www.owasp.org/index.php/Top_10_2013-A7-Missing_Function_Level_Access_Control (Accessed January 10, 2014).

OWASP 2013h. Top 10 2013-A9-Using Components with Known Vulnerabilities. https://www.owasp.org/index.php/Top_10_2013-A9-Using_Components_with_Known_Vulnerabilities (Accessed January 10, 2014).

Pinna, Emilio 2013. Joomla core <= 3.1.5 reflected XSS vulnerability. http://www.securityfocus.com/archive/1/527765 (Accessed December 5, 2013).

Rannajõe, Maria-Elisa 2013. Põhjaranniku veebileht langes kräkkerite rünnaku alla. http://epl.delfi.ee/news/eesti/pohjaranniku-veebileht-langes-krakkerite-runnaku-alla.d?id=51072362 (Accessed November 11, 2013).

Roon, Maarja 2013. Kaitseministeeriumi veebilehe ründajad jäävad ilmselt karistuseta. http://uudised.err.ee/index.php?06290809 (Accessed November 11, 2013).

Saurused OÜ 2013. Firmast. http://www.saurus.ee/firmast (Accessed December 13, 2013).

Schwartz, Mathew 2013. WordPress Site Hacks Continue. http://www.informationweek.com/security/attacks/wordpress-site-hacks-continue/240162060 (Accessed August 11, 2013).

Secunia 2010. Saurus CMS Multiple Vulnerabilities. https://secunia.com/advisories/39773/ (Accessed December 13, 2013).

Secunia 2013. Committed to eliminating vulnerability threats. https://secunia.com/company/ (Accessed December 13, 2013).

Securiteam 2013. Saurus CMS Multiple Script Direct Request Path Disclosure Vulnerability. http://www.securiteam.com/securitynews/6G03A0A8WA.html (Accessed December 14, 2013).

Tohvelmann, Ivar 2011. Avaliku sektori veebilehtede käideldavuse uuring 2010. https://www.ria.ee/public/Programm/veebideuuring_aruanne_final.pdf (Accessed December 18, 2013).

Väikenurm, Marge 2013. Ülemaailmselt tuntud häkker ründas kodulehekülgi. http://www.valgamaalane.ee/316994/ulemaailmselt-tuntud-hakker-rundas-kodulehekulgi/ (Accessed September 11, 2013).

W3Techs Usage of content management systems for websites. http://w3techs.com/technologies/overview/content_management/all (Accessed November 9, 2013).

Wikipedia 2013. Murdskriptimine. http://et.wikipedia.org/wiki/Murdskriptimine (Accessed December 1, 2013).

WordPress Updating WordPress. http://codex.wordpress.org/Updating_WordPress (Accessed November 15, 2013a).

WordPress WordPressi kirjeldus. http://wordpress.org/about/ (Accessed November 8, 2013b).