VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

ZABEZPEČENÝ PŘÍSTUP PRO WEBOVÉ APLIKACE

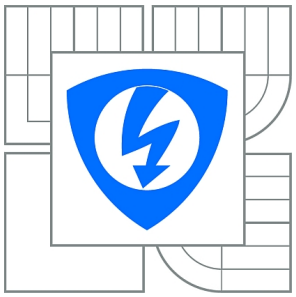
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE JAN HUMPOLÍK
AUTHOR

BRNO 2010

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

ZABEZPEČENÝ PŘÍSTUP PRO WEBOVÉ APLIKACE

SECURED ACCESS FOR WEB APPLICATIONS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

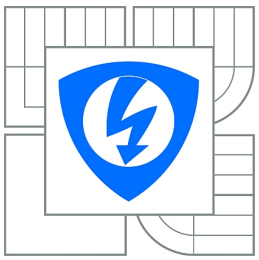
AUTOR PRÁCE — JAN HUMPOLÍK

AUTHOR

VEDOUCÍ PRÁCE — Ing. RADEK DOLEŽEL

SUPERVISOR

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Jan Humpolík ***ID:*** 106483
Ročník: 3 ***Akademický rok:*** 2009/2010

NÁZEV TÉMATU:

Zabezpečený přístup pro webové aplikace

POKYNY PRO VYPRACOVÁNÍ:

Definujte prostředí webové aplikace a současné trendy ve vývoji. Prostudujte obecný princip autentizace a autorizace ve spojení s webovou aplikací. Zaměřte se na zjištění možného nebezpečí, které souvisí se zabezpečeným přístupem k webové aplikaci, ale i webové aplikace samotné. Vytvořte laboratorní prostředí webové aplikace představující systém pro správu obsahu a implementujte bezpečnostní pravidla. Ověřte zabezpečení na základě zjištěného nebezpečí a v případě nalezení bezpečnostních nedostatků navrhněte jejich nápravu. Využívejte nástroje z oblasti Open Source Software.

DOPORUČENÁ LITERATURA:

[1] Endorf, C. Detekce a prevence počítačového útoku. Praha: Grada, 2005. 355s. ISBN 80-247-1035-8.
[2] Garfinkel, S. Web security commerce. USA: O´Reilly, 1997. 483s. ISBN 1-56592-269-7.

Termín zadání: 29.1.2010 ***Termín odevzdání:*** 2.6.2010

Vedoucí práce: Ing. Radek Doležel

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

Tato práce se zabývá zejména často zanedbávaných součástí zabezpečení každé webové aplikace, ale i bezpečným přístupem samotných uživatelů. Popisuje teoreticky i prakticky moderní techniky zabezpečení, na vytvořené webové aplikaci testuje a ukazuje možný způsob obrany. Dává návod na instalaci vlastního webového serveru.

KLÍČOVÁ SLOVA

autentizace, autorizace, bezpečný kód, CAPTCHA, ClickJacking, CSRF, DNSSEC, hašovací funkce, HTTP, HTTPS, IDN, instalace webového serveru, SQL injection, SSL/TLS, šifrovaná komunikace, WAMP, XSS

ABSTRACT

This thesis mainly concerns often neglected security part of each web application, but also secure access users themselves. Describes theoretically and practically modern security technology, on a web application being tested and shows a possible way of defense. Gives instructions for installing its own web server.

KEYWORDS

authentication, authorization, CAPTCHA, ClickJacking, CSRF, DNSSEC, encrypted communication, hash function, HTTP, HTTPS, IDN, safe code, SQL injection, SSL/TLS, WAMP, web server setup, XSS

HUMPOLÍK, J. *Zabezpečený přístup pro webové aplikace*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2010. 56 s. Vedoucí bakalářské práce Ing. Radek Doležel.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Zabezpečený přístup pro webové aplikace“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ....

(podpis autora)

PODĚKOVÁNÍ

Děkuji mému vedoucímu bakalářské práce Ing. Radku Doleželovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování této práce.

V Brně dne ....

(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Dnes, ve věku internetu se vyvíjí mnoho webových aplikací, které jsou dostupné prakticky celému světu. Uživatelé chtějí, aby taková webová aplikace vykonávala jen to, co vykonávat má. Naneštěstí je ale dostupná taky těm, jejichž největší zábavou je zabezpečení prolomit nebo jinak řečeno donutit aplikaci, aby dělala to, co jste při jejím vývoji nepředpokládali.

Tato práce se zabývá zejména často zanedbávaných součástí zabezpečení každé webové aplikace, ale i bezpečným přístupem samotných uživatelů. Popisuje moderní techniky zabezpečení a dává návod na instalaci vlastního webového serveru.

Vysvětlení některých pojmů, zmíněných v této práci

webová stránka – statický webový dokument,
webová aplikace – aplikace přístupná pomocí webového prohlížeče,
vývojář – osoba sdružující funkci programátora, kodéra a designéra,
uživatel – člověk, návštěvník webové stránky/aplikace,
klient – webový prohlížeč.

Kapitoly

1. Webová aplikace – zabývá se historií samotných webových stránek, úvodem do protokolu HTTP a webovými prvky s tím spojených. Na závěr jsou zmíněny nadějné technologie pro budoucí webové aplikace.
2. Moderní techniky zabezpečení – společná kapitola pro uživatele i vývojáře.
3. Zranitelnosti webových aplikací – na praktických ukázkách popisuje chyby, kterých je nutné se při vývoji vyhnout.
4. Nebezpečí na internetu – upozorňuje na rizika na internetu, kterými mohou uživatelé ohrozit bezpečnost svých dat i peněž (bankovní transakce).
5. Přehled bezpečnostních pravidel – všechna uváděná pravidla pohromadě, opět jak pro uživatele, tak pro vývojáře.
6. Instalace webového serveru – návod na instalaci vlastního webového serveru.
7. Praktická část – v této části bylo ověřeno a ukázáno, jak se bránit před všemi webovými zranitelnostmi popsanými v této práci.

Upozornění

Veškeré níže uvedené detailní informace a postupy slouží pouze ke studijním účelům. Jakékoliv zneužití může být nezákonné a kvalifikováno jako trestní čin, za něž není autor zodpovědný. Podle § 230 zákona č. 40/2009 Sb. [1] až na 8 let odnětí svobody.

1 WEBOVÁ APLIKACE

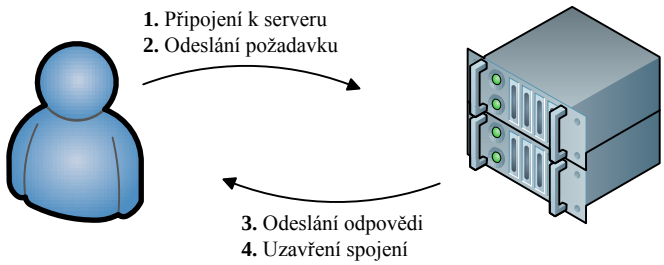
Autorem webových stránek je fyzik Tim Berners-Lee, který je začal v březnu roku 1989 vyvíjet při svém působení v CERNu (European Organization for Nuclear Research). Cílem organizace je spolupráce evropských států v oblasti čistě vědeckého a základního výzkumu, nezabývá se činností pro vojenské účely. Výsledky jejich experimentálních a teoretických prací se zveřejňují nebo jinak zpřístupňují veřejnosti. Záměrem bylo spojit hypertext s internetem a osobními počítači, což má tvořit jednotný informační systém na pomoc fyziků CERNu sdílet všechny informace uložené na počítačích v laboratořích.

Tim Berners-Lee navrhl jazyk HTML (HyperText Markup Language) a protokol HTTP (Hypertext Transfer Protocol), vytvořil první webový prohlížeč na světě WorldWideWeb a koncem roku 1990 spustil také vůbec první webový server [2]. Neexistují žádné snímky z této původní stránky, ale je možné se podívat na pozdější verzi z roku 1992, viz zdroj [3]. V říjnu roku 1994 založil W3C (World Wide Web Consortium), které dohlíží na další vývoj.

I když webové stránky začaly jako nástroj na pomoc fyzikům zodpovědět těžké otázky týkající se vesmíru, dnes se jejich používání vztahuje na různé aspekty globální komunity a ovlivňuje náš každodenní život.

1.1 Ukázka komunikace

Webové stránky jsou ve své podstatě jen dokumenty propojené hypertextovými odkazy, ty umožní uživatelům snadno procházet mezi texty na webových stránkách pomocí odkazů. Pro přesun mezi nimi byl vytvořen protokol HTTP [4, 5], v současné době se používá ve verzi 1.1, která přináší podporu pro více virtuálních serverů v rámci jednoho fyzického serveru. Praktická ukázka viz následující strana, která popisuje komunikaci protokolu z obr. 1.1.



Obr. 1.1: Model klient–server používaný na webových stránkách.

Požadavek klienta

```
GET / HTTP/1.1
Host: www.example.com
Accept: text/html, text/plain, image/*
Accept-Language: cs
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
```

Odpověď serveru

```
HTTP/1.1 200 OK
Date: Wed, 07 Oct 2009 17:40:01 GMT
Server: Apache/1.3.41
Last-Modified: Tue, 15 Nov 2005 13:24:10 GMT
Content-Length: 84
Connection: close
Content-Type: text/html; charset=UTF-8

<html>
<head>
  <title>Example</title>
</head>
<body>
  <p>Hello, world!</p>
</body>
</html>
```

Nejčastější návratové kódy

`200 OK` – v pořádku
`301 Moved Permanently` – trvalé přesměrování
`303 See Other` – zabraňuje opakovanému odeslání formuláře při znovunačtení stránky
`304 Not Modified` – soubor se od předchozího GET požadavku nezměnil (cache)
`403 Forbidden` – požadavek byl odmítnut
`404 Not Found` – požadovaný zdroj nebyl nalezen
`500 Internal Server Error` – chyba serveru
`503 Service Unavailable` – přetížení serveru

Návratové kódy se dělí do těchto skupin: `1xx` – informační, `2xx` – potvrzení, `3xx` – přesměrování, `4xx` – chyba klienta, `5xx` – chyba serveru.

1.2 Relace

V angličtině známé jako Session ID [4], je jedinečný identifikátor, který identifikuje relaci na serveru. Protože je protokol HTTP bezestavový, neexistuje žádné pojítko mezi různými požadavky jednoho klienta. Klient zašle požadavek, server odpoví a spojení se ukončí. Pojítko se vytváří pomocí souborů cookies, na požadavek serveru (webové aplikace) se do jednoho z nich uloží identifikátor, který pak klient v dalších požadavcích posílá serveru. K posílání i přijímání souborů cookies se používají HTTP hlavičky. Cookies se nepoužívají pouze na identifikace relace, ale např. i pro statistiky přístupů uložením času posledního přístupu. Doporučený limit [6] je 20 cookies na doménu, 4 KB na cookie, u různých webových prohlížečů se tyto limity mohou měnit směrem nahoru. V případě, že klient nepodporuje cookies, nebo je má zakázané, je možné předávat identifikátor relace pomocí metody GET v URL dotazu. Tato metoda ale není příliš vhodná, protože se u ní objevuje identifikátor relace v záznamech webových serverů a historii webového prohlížeče, odkud může být relace zneužita. Alternativou pro přenos Session ID bez použití cookies je přenos přes JavaScript, viz [7, 8].

1.3 Web 2.0

Tímto, poněkud *buzzword* termínem, se označuje současný trend tvůrců webových aplikací, kteří dávají uživatelům nástroje (rozhraní) a ti tvoří obsah [9]. Příkladem takových webových aplikací mohou být Wikipedie [10] (otevřená encyklopedie), YouTube [11] (video portál), Facebook [12] (sociální/společenská síť), Twitter [13] (mikroblogovací služba), Digg [14] (sdílení odkazů), last.fm [15] (hudba a personalizované rádio), MySpace [16] (komunitní server a hudba), LinkedIn [17] (profesní síť životopisů).

Specifické znaky

- změna komunikačního modelu z one-to-many (jediný tvůrce obsahu) na many-to-many (mnoho tvůrců),
- rozmlžení hranice tvůrce/uživatel,
- webové služby nahrazující desktopové aplikace,
- možnost využívat API (Application Programming Interface) jiných webů,
- velká koncentrace dat, u nichž je často důležitější kvantita, než kvalita.

1.4 Budoucnost

Ačkoliv je tzv. internetová horečka z přelomu tisíciletí již za námi, online aplikace prožívají bouřlivý rozvoj i dnes. Původně desktopové aplikace se dostávají i na web, jen čas ale ukáže, které si lidé oblíbí a uchytí se. Jejich hlavní výhoda je být k dispozici vždycky a všude, kde je rychlé připojení k internetu.

HTML 5

Dnešní nové webové stránky jsou tvořeny podle standardu XHTML 1.1 (Extensible HyperText Markup Language) nebo HTML 4.01, kromě toho jsou však webdesignéři nuceni pro požadovaný efekt znát a používat mnoho dalších jazyků (JavaScript, ActionScript). Změnit by to mělo HTML 5 [18], ačkoliv je od roku 2004 stále ve fázi vývoje, tak byla počátkem roku 2008 vydána první testovací verze specifikace a některé webové prohlížeče (Google Chrome 2, Firefox 3.5, Apple Safari 4) ji už částečně implementovaly.

Současná verze HTML 4 z konce 90. let je zaměřena na strukturu webové stránky, nijak neulehčuje tvorbu interaktivního obsahu. Proto v nové verzi přibudou nové sémantické prvky, které výrazně pomohou k přehlednému popisu struktury (ubude nadměrného používání tříd a identifikátorů). Výběr z nových prvků:

- záhlaví <`header`>, resp. zápatí stránky <`footer`>,
- multimediální objekty <`audio`> a <`video`> bez potřeby mít Flash apod.,
- <`figure`> pro svázání multimediálního obsahu a jeho textového popisku,
- značka <`canvas`> schopná vykreslovat grafiku, dokonce i 3D.

Již dnes se dají na internetu nalézt ukázkové aplikace, ale o praktickém používání se ještě mluvit nedá, na to si budeme muset počkat alespoň do roku 2012.

SPDY protokol

SPeeDY [19] je experimentální protokol pro rychlejší web, návrh vytvořený společností Google jako náhrada za protokol HTTP. Stále sice používá HTTP metody, hlavičky a další sémantiku, ale potlačuje jiné části protokolu, jako řízení spojení a formát přenosu dat.

Cílem je snížit čas nutný k načtení webové stránky. To je dosaženo prioritami, odstraněním zbytečných hlaviček protokolu HTTP, kompresí hlaviček, možností serveru iniciovat komunikaci s klientem (přednačítání) a multiplexováním (mnoho souběžných HTTP požadavků po jedné TCP relaci). S prototypem webového prohlížeče Google Chrome a speciálně upraveného Google serveru byla doba načítání stránky až o 64 % kratší oproti stávajícímu protokolu HTTP.

2 MODERNÍ TECHNIKY ZABEZPEČENÍ

V dnešním světě je na internetu množství našich dat, které byly dříve pouze offline. Musíme proto přístup k webovým aplikacím a jejich obsah aktivně chránit před zneužitím, či přímo odcizením. Tato kapitola se zabývá současnými technikami ochrany, jak z hlediska uživatele (návštěvníka), tak vývojáře webových aplikací.

2.1 Šifrovaná komunikace

Běžně se používá např. u administrace, internetového bankovnictví či obchodu, nebo jen pro proces přihlášení do e-mailové schránky a dalších služeb. Využívá jedné z nejobtížnějších úloh ve výpočetní technice, jenž je považováno nalezení účinného algoritmu pro faktorizaci (rozklad celého čísla na součin prvočísel) velkých čísel. Nejlepší dosud nalezený algoritmus má ale exponenciální obtížnost, možným řešením by bylo použití kvantového počítače, který by měl v tomto případě lineární časovou obtížnost. Podle vize budoucnosti společnosti Cisco Systems [20] by se mohl první komerční kvantový počítač objevit na pultech obchodů již v polovině roku 2020.

Při nemožnosti použít šifrované spojení je alternativou pro zabezpečený přenos hesla technika „výzva-odpověď“ [21], ta ale vyžaduje na klientské straně podporu JavaScriptu pro vypočet haše se solí (pomocí externí knihovny).

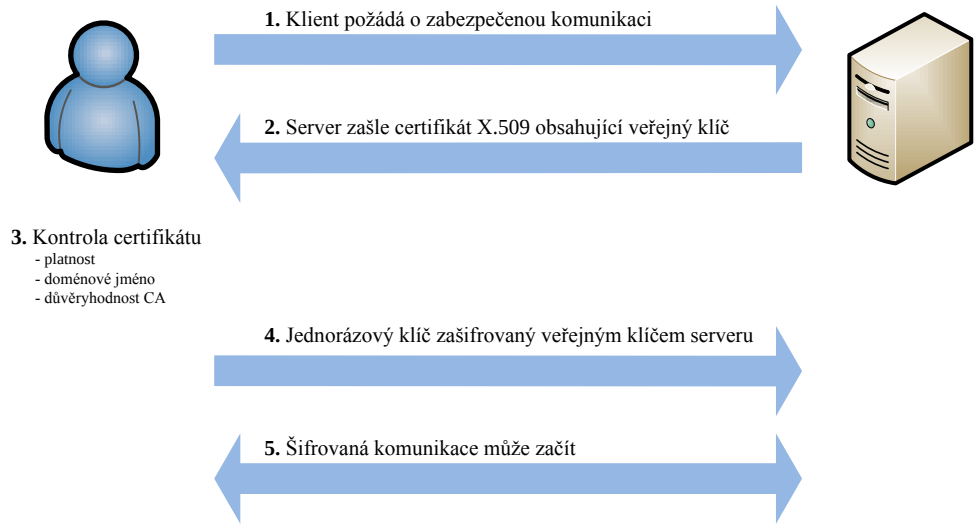
Prvotní navázání komunikace

Komunikace pomocí veřejného a soukromého klíče se používá pouze při navazování spojení, pak si klient se serverem dohodnou z důvodu rychlosti symetrický klíč, který dále používají. Grafické znázornění viz obr. 2.1.

Klient (webový prohlížeč) si stáhne certifikát uložený na serveru, zkontroluje jeho podpis s podpisy CA (Certification Authority) uložených ve své databázi (bez autentizace serverového certifikátu je uživatel vystaven riziku útoku MITM – Man In The Middle), dále časovou platnost a doménové jméno s doménovým jménem na něž je certifikát vydán. V případě že je vše v pořádku, webový prohlížeč z certifikátu vyčte veřejný klíč, kterým zašifruje jím náhodně vygenerovaný jednorázový klíč, který následně zašle serveru. Ten pomocí svého soukromého klíče dešifruje zprávu, nyní mají obě strany symetrický klíč.

Serverový certifikát

K ustavení spojení stačí pouze certifikát serveru, certifikát klienta není požadován, ale může podstatně zvýšit úroveň důvěry. Toho se využívá u internetového ban-



Obr. 2.1: Ustavení šifrovaného spojení (SSL handshake, tedy potřesení rukou).

kovnictví, kde mívá uživatel s certifikátem větší limit na bezhotovostní transakce. Nevýhodou však je, že některé webové aplikace jsou navrženy tak, že nestačí mít jen naimportovaný certifikát do prohlížeče, ale je taky nutné mít nainstalované prostředí Java Virtual Machine.

Certifikát X.509 je standard pro jednoduché podepisování založeném na veřejném klíči (PKI, Public Key Infrastructure). Termínem X.509 je dnes běžně označován standard X.509 v3, ten např. obsahuje verzi certifikátu, sériové číslo, dobu platnosti, informace o vlastníkovi a mnohé další detailní informace [22], které uživateli umožní ověření jeho pravosti a platnosti. U společnosti VeriSign a GeoTrust začíná cena za standardní serverový certifikát na $399 [23, 24], u Thawte pak na $249 [25] za 1 rok. Při prodlužování na další rok, nebo při koupi na více let dopředu, může být cena nižší. Varianta EV (Extended Validation) je certifikát s rozšířeným ověřením, který je v novějších prohlížečích označen zelenou barvou s informací o organizaci, která jej vlastní. Za ten si ale nechávají CA řádně zaplatit (VeriSign $995), není proto divu, že je takových webových aplikací velmi málo – převážně internetová bankovnictví.

Doba ochrany	**Do roku 2010**	**2010 až 2030**	**2030 a později**
Minimální symetrický klíč	80 bitů	112 bitů	128 bitů
Minimální RSA klíč	1024 bitů	2048 bitů	3072 bitů

Tab. 2.1: Doporučená minimální délka symetrického a RSA klíče. [26]

Druhy protokolů

TLS (Transport Layer Security) a jeho předchůdce SSL (Secure Sockets Layer), jsou protokoly resp. vrstvy vložené mezi vrstvu transportní (např. TCP) a aplikační (např. HTTP) [27]. Mezi prohlížeči nejvíce podporovanými protokoly SSL 3.0 (1996) a TLS 1.0 (1999) jsou drobné rozdíly, ale v podstatě jsou stejné, nové prohlížeče podporují už i TLS 1.2 (2008). Velkou výhodou TLS je podpora SNI (Server Name Indication), nebo-li podpora více doménových jmen na jediné IP adrese – typický případ u běžných webhostingů, kde existuje mnoho virtuálních webových serverů na jednom fyzickém serveru.

2.2 Technologie AAA

V oblasti počítačové bezpečnosti AAA znamená Authentication, Authorization and Accounting protocol [28, 29], tj. česky autentizační, autorizační a účtovací protokol.

Autentizace je proces, kdy jeden subjekt ověřuje jinému subjektu nárok na držení konkrétní digitální identity. Obyčejně jeden subjekt je klient (uživatel, počítač klienta, atd.) a jiný subjekt je server (počítač). Autentizace provádí žadatel pomocí předložení identifikačního a odpovídajícího pověření ověřovateli. Příklady typů oprávnění jsou hesla, jednorázová pověření, digitální certifikáty nebo telefonní čísla.

Autorizace je rozhodnutí o povolení přístupu uživateli, na základě jeho autentizace, služeb, které požaduje a aktuálního stavu systému. Může být založena na omezeních, například omezení na určité hodiny v rámci dne, omezení na fyzickou polohu, nebo omezení vícenásobného přihlášení jednoho uživatele. Autorizace určuje povahu služby, která je poskytnuta uživateli.

Účtování znamená sledování využívání služeb uživateli. Tyto informace mohou být použity pro správu, plánování, účtování, nebo další účely. Typické informace, které jsou shromážděny jsou identita uživatele, povaha dodaných služeb a časy počátků a konců dodaných služeb.

2.3 Hašovací hesel

Hašovací funkce, vytváří otisk ze zprávy na vstupu (možný i obsah souboru), a ten bude pokaždé stejný, ale neexistuje způsob, jakým bychom mohli z haše vypočítat zpět původní zprávu – fungují pouze jedním směrem. Pravděpodobnost, že dvě různé zdrojové zprávy budou mít stejný haš, je natolik malá, že se neuvažuje. Aby se nedaly použít databáze předpočítaných hašů (Rainbow tables), doporučuje se před hašováním připojovat k heslu náhodný řetězec – tzv. *solení* (optimálně algoritmem HMAC – Hash-based Message Authentication Code [30]), který je uložený

mimo databázi se zahašovaným heslem a je v čase stejný, ale zároveň jiný pro každou instalaci. Útočník by po té musel vygenerovat vlastní Rainbow tables. Velikost potřebných tabulek pro nejpoužívanější algoritmy můžete nalézt v příloze A. Mezi typické hašovací algoritmy patří MD5 (prolomena), SHA-1 (náročnost snížena na 2^{61}) a SHA-256 (zvolna nastupující). [31]

2.4 CAPTCHA

Completly Automated Public Turing test to tell Computers and Humans Apart [32] se používá ve webových aplikacích pro automatické odlišení skutečného uživatele od robotů. Test zpravidla spočívá v zobrazení obrázku s deformovaným textem, přičemž úkolem uživatele je zobrazený text opsat do příslušného vstupního políčka. Nemusí jít vždy jen o obrázek s textem, ale i o zvukový záznam rušného prostředí, mezi kterým někdo v pozadí diktuje čísla (video ukázka prolomení viz [33]). Modifikace myšlenky v podobě systému reCAPTCHA [34] navíc pomáhá při digitalizaci knih. Slouží mimo jiné jako ochrana před:

- komentářovým spamem,
- hromadnou registrací e-mailových schránek a dalších služeb, které by později mohly být zneužity ke spamování, resp. jiným činnostem,
- hrubou silou slovníkového útoku do přihlašovacích systémů.

2.5 Bezpečné DNS

DNS (Domain Name System) [35] je protokol aplikační vrstvy (TCP/53 a UDP/53) umožňující vzájemný převod doménových jmen a jim odpovídajících IP adres.

Komunikace probíhá metodou klient–server, v případě více současných požadavků musí systém rozeznat, ke kterému požadavku patří odpověď. Při použití TCP protokolu by to nebyl problém, tam je komunikace spojově orientovaná a dotaz i následná odpověď se přenáší v rámci jednoho identifikovatelného TCP spojení, ale u protokolu UDP taková možnost není. Proto se v hlavičce DNS požadavku vyskytuje položka ID (16bitové číslo), kterou generuje klient při požadavku a DNS server poté uvádí v hlavičce odpovědi stejné. Hlavní slabina protokolu je právě v této identifikaci odpovědí (klient akceptuje první odpověď, která obsahuje příslušné ID v hlavičce) a v tom, že komunikace neprobíhá šifrovaně. Útočník může změnou DNS záznamu ovlivnit fungování dalších internetových služeb:

- pomocí podvržených webových stránek získávat údaje o platebních kartách, přihlašovací údaje do e-mailové schránky apod.,
- přesměrovávat, nebo odposlouchávat VoIP hovory (podvržená ústředna),

– obcházet antispamovou kontrolu fungující na principu směrování příchozí pošty pomocí DNS záznamů na kontrolní servery,
– získávat cizí e-maily.

DNSSEC

DNSSEC (Domain Name System Security Extensions) [36] je zabezpečená verze překladu doménových jmen, které umožňuje na některých doménách ověřit pravost informací získaných z DNS. Zavádí do DNS asymetrickou kryptografii, tedy používání jednoho klíče na zašifrování (veřejný klíč) a jiného klíče na dešifrování obsahu (soukromý klíč).

Držitel domény vygeneruje dvojici soukromého a veřejného klíče. Svým soukromým klíčem pak elektronicky podepíše technické údaje, které o své doméně do DNS vkládá. Pomocí veřejného klíče je pak možné ověřit pravost tohoto podpisu. Aby byl tento klíč dostupný všem, publikuje jej držitel ke své doméně u nadřazené autority, kterou je pro všechny domény .cz registr domén .cz. I na úrovni registru domén .cz jsou technická data v DNS podepsána a veřejný klíč k tomuto podpisu je opět správcem registru předán nadřazené autoritě. Pokud není v žádném svém článku řetěz porušen a všechny elektronické podpisy souhlasí, tak se tímto zajistí důvěryhodnost údajů.

Veřejné DNS

V případě problémů s používanými DNS servery (např. časté výpadky, blokování obsahu, přetížení, či jen malá důvěryhodnost u malých ISP – Internet Service Provider) je možné je jednoduše nahradit změnou primárního a sekundárního DNS serveru v operačním systému, nebo na DHCP (Dynamic Host Configuration Protocol) serveru, případně přímo změnou na Výchozí bráně (typicky ADSL modem).

Google DNS [37] – `8.8.8.8, 8.8.4.4`
OpenDNS [38] – `208.67.222.222, 208.67.220.220`

Google DNS je výrazně jednodušší, nabízí jen samotnou službu veřejného DNS, za to OpenDNS umí i blokovat Vámi definovaný obsah (přes 50 kategorií), white-list/blacklist jednotlivých domén a mnohé další funkce.

3 ZRANITELNOSTI WEBOVÝCH APLIKACÍ

Na zabezpečení dat webových aplikací nestačí jen firewall a instalace aktuálních bezpečnostních záplat. Webové aplikace pracující na straně serveru (PHP – Hypertext Preprocessor, ASP.NET – Active Server Pages, Java, Perl a mnohé další) a docela často zdrojové texty píší vývojáři, kteří považují zabezpečení za úkol správců systému, jednoduše se zajímají jen o samotnou funkčnost aplikace. Důsledkem toho pak je množství dynamických webových míst, obsahujících ohromné trhliny, které je činí zranitelnými všemi druhy útoků.

Tato kapitola ukazuje nejčastější chyby, kterých se můžete ve svých webových aplikacích dopustit. Vyžaduje alespoň základní znalosti HTML, CSS a JavaScriptu. Pro serverově orientované příklady byl zvolen jazyk PHP pro jeho velkou rozšířenost a jednoduchost.

3.1 Ošetření neplatného vstupu

Většina webových aplikací očekává nějaký vstup dat od uživatelů, ať už ve formě formulářů, parametrů v URL, souborů atp. Při nedůsledné kontrole vstupů by se útočník mohl dostat na místa, na které by se dostat neměl. Přijímání dat od uživatele patří z hlediska bezpečnosti mezi to nejrizikovější.

SQL injection

Pomocí SQL (Structured Query Language) Injection [4] je útočník schopen měnit nebo zadávat dotazy, které se odesílají do databáze prostřednictvím vstupů webové aplikace. Metoda zneužívá apostrofů a dalších znaků k vsunutí (odtud „injection“) vlastního kódu přes neošetřený vstup, tomuto se zabraňuje pomocí *escapování* [39] těchto nebezpečných znaků. Detailní popis problematiky viz [40].

Obrana: V případě doporučeného rozšíření MySQLi pro připojení k MySQL serveru >= 4.1, používat funkci `mysqli_real_escape_string` na každý vstup do databáze, případně můžete využít *Vázání proměnných* [41] (Prepared Statements). Z bezpečnostních důvodů je doporučeno používat UTF-8 kódování (pomocí funkce `mysqli_set_charset`), protože v některých jiných vícebajtových znakových sadách se dá ochrana teoreticky obejít [42].

3.2 Krádež relace

Jak už víme (kapitola 1.2), HTTP je bezestavový protokol, ale je potřeba alespoň určitou formu stavu přenášet, proto se používá mechanizmus relace (Session). Uži-

vateli je serverem při prvním přístupu na webovou aplikaci vytvořen identifikátor relace, který je při každém dalším přístupu odesílán klientem zpět na server ve formě cookie, nebo pomocí předávání v URL (Uniform Resource Locator).

Session hijacking

Nebo-li ukradení Session ID oprávněnému uživateli. Mechanizmus relace je možné zneužít, stačí získat jeho identifikátor a následně můžeme předstírat korektně přihlášeného uživatele. Proto je nutné identifikátory relace generovat opravdu náhodně a ještě lépe kontrolovat i IP adresu uživatele (pozor, může se měnit). [43] Doporučenou ochranou je ukončovat relaci se serverem skrze odhlášení ve webové aplikaci, tj. neřešit to pouhým ukončením (zavřením) webového prohlížeče.

Obrana: Příznakem HttpOnly [44] u cookies zamezíte dostupnost Session ID pomocí JavaScriptu, např. pomocí funkce `session_set_cookie_params`. Ideálně v kombinaci s šifrovaným spojením.

Session fixation

Útočník nastaví klientovi nějakou hodnotu Session ID (podstrčením odkazu s předáním relace pomocí metody GET, JavaScriptem, nebo přímou editací uživatelových cookies) a jakmile se uživatel přihlásí, tak tuto Session ID použije pro sebe. Obrana je poměrně jednoduchá, stačí po přihlášení vygenerovat nové Session ID.
Příklad JS: `javascript:void(document.cookie = "PHPSESSID=hodnota");`

Obrana: Po přihlášení (a jiných změn role) volat funkci `session_regenerate_id`, která způsobuje změnu Session ID, původní ID tak bude pro útočníka bezcenné. Sekundárně je žádoucí mít ošetřené všechny vstupy aplikace před XSS.

3.3 Cross-Site Scripting (XSS)

Ve většině případů je záměrem tohoto útoku ukradnout informaci o probíhající relaci (Session ID) z cookies. To se může stát záměrným vložením nebezpečného skriptu zpracovávaném na straně klienta (např. JavaScript, VBScript) do obsahu webové stránky. Protože je jazyk HTML schopný zpracovávat klientské skripty i jako reakce na události jednotlivých značek (viz následující příklad), je potřeba převést < a > na jejich HTML entity a tím zabránit jejich provedení webovým prohlížečem.

```
<img src="obrazek.gif" width="0" height="0" onload="skript(this);" />
```

Další možností je HTML kód a klientské skripty při vypisování úplně odstranit.

V případě neošetření všech těchto metaznaků, může dojít zobrazováním této upravené stránky k jeho provedení. Skript má přístup k obsahu cookies (a tím pádem Session ID), protože je spouštěn z dané domény.

Obrana: Používat funkci `htmlspecialchars` pro všechny vstupy od uživatele (vč. metod POST, GET a proměnné $_SERVER) a výstupy z databáze (je praktičtější použít funkci až na data při výpisu, než při ukládání do databáze), které webová aplikace vypisuje. Riziko XSS [43] lze výrazně snížit oddělením systému (např. administrace) jinou doménou (doména 3. řádu stačí, je ovšem třeba počítat s omezením [6] 20 cookies na doménu) od dalších okrajových služeb (např. blog či fórum).

3.4 Cross-Site Request Forgery (CSRF)

Předpokládejme, že je uživatel právě přihlášen ve webové aplikaci (to není problém, protože řada webových aplikací nabízí trvalé přihlášení pomocí dlouhodobě platné cookie). Nyní si představme, že útočník přiměje uživatele navštívit (např. odkazem v e-mailu) webovou stránku, která navíc obsahuje následující HTML kód:

```
<img src="http://www.example.com/admin/?clanek=123&smazat"
width="0" height="0" alt="" />
```

Pokud webová aplikace není odolná proti CSRF [43], tak bude požadavek úspěšný. Obranou není ani vyžadování příjmu dat metodou POST, protože i to lze provést pomocí JavaScriptu vloženého v `<iframe>` nulové velikosti. Pro obranu je tedy nutné používat některou z následujících metod:

- ve formulářích generovat unikátní kód (ten může být stejný po celou dobu relace) do skrytého prvku `<input>`, a ten při odeslání formuláře kontrolovat webovou aplikací,
- potvrzovat operace pomocí CAPTCHA nebo SMS,
- mít co nejkratší platnost relace (nepoužívat trvalé přihlášení), protože odhlášenému uživateli žádné riziko nehrozí.

3.5 ClickJacking

Zranitelnost spočívá v tom, že útočník na pozadí za svojí stránkou zobrazí skrytě v `iframe` naši webovou aplikaci. Následně přiměje oběť, aby do své stránky klikal, ve skutečnosti to jsou ale požadavky do naší aplikace. [45] Ukázkové video viz [46], praktická ukázka viz [47].

Obrana: Nejspolehlivějším řešením je zasílat hlavičku `X-Frame-Options: deny`, tu ale podporují jen moderní webové prohlížeče.

4 NEBEZPEČÍ NA INTERNETU

Internet je nejen zdrojem informací a zábavy, ale může se stát i nebezpečím pro naše data a finanční prostředky. V této kapitole se podíváme, na jaká nebezpečí si dát jako uživatel (návštěvník) pozor.

4.1 Sociální inženýrství

Často je mnohem jednodušší věnovat se lidským slabinám, než se probourávat zabezpečením, které navíc bývá „neprůstřelné“. Této problematice se detailně věnuje kniha Umění klamu [48], obsahující spoustu ukázkových příběhů, které přiblíží o co jde a na co si dát pozor. Autorem je Kevin Mitnick, výborný sociotechnik, který je považován za nejslavnějšího počítačového hackera na světě a který se právě díky sociálnímu inženýrství dokázal v 90. letech dostat téměř do každého systému.

Phishing aneb rhybaření

Souhrnně označuje jakékoliv aktivity, které vedou k získávání osobních informací, zejména přihlašovacích údajů a čísel platebních karet. Phishing [49] v e-mailech, instant messagingu a na webu využívá hlavně možnosti zveřejnit odkaz, který ve skutečnosti vede někam jinam, než jak vypadá.

Existuje několik úrovní boje s phishingem, na uživatelské úrovni zejména je to dodržování bezpečnostních pravidel, na aplikační úrovni umožňují některé nové webové prohlížeče automaticky porovnávat známé phishingové domény s databází a upozorňovat na ně. Toto už dělají i některé vyhledávače ve výsledcích vyhledávání a veřejné DNS servery.

4.2 Spyware

Je to označení pro software, který využívá internetu k odesílání dat z počítače bez vědomí jeho uživatele. Autoři obhajují jejich existenci tím, že odesílají pouze informace za účelem zjištění potřeb nebo zájmů uživatele a ty dále využívají pro cílenou reklamu. Spyware může odesílat:

- historii prohlížených stránek,
- seznam otevíraných souborů,
- IP adresu uživatele,
- celé dokumenty,
- uživatelova hesla,
- informace o souborech nebo nainstalovaných programech.

Ochrana je poměrně jednoduchá, neinstalovat podezřelé programy z internetových stránek s podezřelým obsahem (warez) a používat nejnovější verze programů, od kterých se očekává ošetření starých zranitelností. Tyto neopravené díry, zejména v internetovém prohlížeči, by mohly znamenat nainstalování Spyware [50] i pouhým vstupem na nebezpečnou stránku, tj. bez typického odklepnutí dialogového okna.

Podezřelá je automaticky změněná domovská stránka, nové ikony na ploše (které se záhadně objevují) či např. přesměrování telefonní linky u vytáčeného připojení. V případě pochybností je možné systém otestovat, mezi nejznámější programy pro Microsoft Windows patří Lavasoft Ad-Aware [51], nebo volně dostupný Spybot Search & Destroy [52] (s pomocí Wine je podpora i v GNU/Linux).

4.3 IDN homograph attack

IDN (Internationalized Domain Names) [53, 54] jsou internetové domény, které mohou obsahovat i znaky národních abeced (diakritiku, čínské znaky, cyrilice atd.), díky čemuž se přes ně dostanou na webové stránky i lidé, kteří latinku neznají. Na druhou stranu ale poskytuje další možnost doménovým spekulantům, vytváří další náklady na držení všech kombinací doménových jmen majitelům webových stránek a v neposlední řadě je tady také nevýhoda zaměnitelnosti některých stejně nebo podobně vypadajících znaků v IDN použitém kódování Unicode, důsledkem čehož se uživatel může stát obětí phishingového útoku (kapitola 4.1).

Jak se tvoří překlad IDN domény?

1. název domény začíná „xn–“,
2. následují znaky domény vyjádřené standardní latinkou, znaky s diakritikou jsou vynechány,
3. za pomlčkou jsou vyjádřené znaky s diakritikou ve zkráceném Unicode kódování pomocí algoritmu Punicode.

Překladem domény „www.háčkyčárky.cz“ vznikne „www.xn–hkyrky-ptac70bc.cz“. Doména www.háčkyčárky.cz je jediná CZ doména, která umožňuje zobrazení diakritiky. Je spravována sdružením CZ.NIC, které na ni publikuje informace o IDN, spuštění IDN na celé doméně .cz se zatím neplánuje.

4.4 DNS spoofing

Je založen na generování falešných DNS (kapitola 2.5) odpovědí, které dorazí klientovi dříve, než od původně dotazovaného DNS serveru. Klient v takovém případě první (podvrženou) odpověď přijme a další zahodí, protože již jinou neočekává.

Odposlech sítě – hrozí v lokální síti, či u nezabezpečeného směrovače (routeru). Proti tomuto útoku není odolný žádný DNS server, protože útočník již má vše potřebné (ID dotazu, IP adresy a porty zdroje i cíle). Jedinou možnou ochranou by bylo šifrovaní spojení (kapitola 2.1) s DNS serverem anebo použití DNSSEC (kapitola 2.5).

Hádání ID dotazu – pokud útočník nemá přístup do sítě pro odposlech DNS komunikace, tak musí nějak jinak zjistit port a ID dotazu. DNS server náchylný k tomuto útoku, se vyznačuje tím, že používá vždy stejný zdrojový port pro odchozí DNS dotazy. K jeho zjištění se využívá rekurzivních dotazů, kterými se DNS server snaží zjistit z autoritativního DNS (útočníkem vytvořeného serveru) informace o jemu neznámé doméně. Detailnější popis viz [35].

DNS cache poisoning – jde o chybnou implementaci v systému DNS zveřejněnou v srpnu 2008 [55]. Princip opět spočívá v rekurzivních dotazech DNS serveru na zjištění neznámé domény od autoritativního DNS, ale nyní útočník nemusí čekat na vypršení doby TTL (Time To Live), po kterou je záznam ve vyrovnávací paměti (cache) serveru. Útočník se totiž neptá přímo na IP adresu webového serveru, ale na libovolně zvolený neexistující záznam a IP adresu webového serveru podvrhne pomocí sekce `AUTHORITY` a `ADDITIONAL`, které nejsou kontrolovány.

5 PŘEHLED BEZPEČNOSTNÍCH PRAVIDEL

5.1 Pro tvůrce webových aplikací

- používejte nejnovější bezpečnostní záplaty k vámi použitým serverům,
- všechna data přicházející v odpovědi od klienta, můžou mít naprosto neočekávané hodnoty. Útočník může např. pozměnit HTTP hlavičku Referer, soubory cookie, skrytá HTML pole nebo položky seznamů,
- pro testování zabezpečení nepoužívejte skripty běžící na straně klienta,
- při přihlášení uživatele vždy vygenerujte nový identifikátor relace, vyhnete se tak možnému útoku typu Session Fixation,
- nezobrazujte podrobná chybová hlášení, jsou ukazatelem slabých míst aplikace,
- ošetřete všechny metaznaky (např. zpětná lomítka) předávané do subsystému,
- nepoužívejte metodu GET v souvislosti s posíláním identifikátoru relace a jiných tajných informací, které se nesmí objevit v záznamech webových serverů a historii webového prohlížeče,
- vytvářejte záznamy i na úrovni aplikace, o přihlášení uživatele apod.,
- neukládejte hesla v nešifrované podobě, ukládejte pouze haše hesel,
- při filtrování dávejte přednost whitilistingu před blacklistingem,
- nedůvěřujte zabezpečení hodnot vrácených webovou API (Application Programming Interface) a vytvořte si vlastní kontrolu těchto hodnot,
- pro vstup vytvořený serverem použijte nepřímý přístup k datům vždy (pomocí indexů nebo názvů), kdy je to možné,
- snažte se vytvořit vícestupňové zabezpečení, např. omezením práv pro zápis do databáze jen tam, kde je to nevyhnutelně nutné.

5.2 Základní pravidla bezpečného internetu

- samozřejmostí je instalace nejnovějších verzí webového prohlížeče a bezpečnostních záplat k operačnímu systému,
- používejte firewall, nejlépe i pro odchozí komunikaci,
- neinstalujte nedůvěryhodné pluginy do webových prohlížečů, mohou odposlouchávat komunikaci, či být zdrojem phishingových útoků,
- přihlašovací hesla volte nejméně z 8 znaků, jejichž kombinace velkých, malých písmen a číslic nejsou slova nebo zkratky, které se dají najít ve slovníku,
- pokud to bude možné, používejte šifrované spojení, ale nepokračujte dále při varování prohlížeče na nedůvěryhodný certifikát serveru,
- v operačním systému nebo na DHCP nastavte důvěryhodné DNS servery.

6 INSTALACE WEBOVÉHO SERVERU

Pro vyzkoušení skriptů je potřeba mít webový server, pro návod byla zvolena nejčastější konfiguraci pro vývoj – Open Source Software na platformě Microsoft Windows. V bezpečnosti je třeba se držet nejnovějších verzí programů, proto i tento návod popisuje instalaci posledních verzí, pokud možno již v 64bitové variantě.

MySQL 5 doplnila podporu uložených procedur, pohledů (views), kurzorů, spouštěčů (triggers) a ve verzi 5.1 i partitioningu. Stala se tak použitelnou i pro náročnější, alternativou je PostgreSQL [56]. Předností systému PostgreSQL (oproti MySQL) je možnost používat různé programovací jazyky v uložených procedurách, avšak nevýhodou je málá rozšířenost na hostingových serverech.

6.1 Použité technologie

Microsoft Windows 7 x64
Apache 2.2.15 [57] (httpd-2.2.15-win32-x86-openssl-0.9.8m-r2.msi)
OpenSSL 0.9.8m – součástí instalace Apache
MySQL 5.1.47 x64 [58] (mysql-essential-5.1.47-winx64.msi)
PHP 5.3.2 [59] (php-5.3.2-Win32-VC6-x86.zip)
phpMyAdmin 3.3.3 [60] (phpMyAdmin-3.3.3-all-languages.zip)

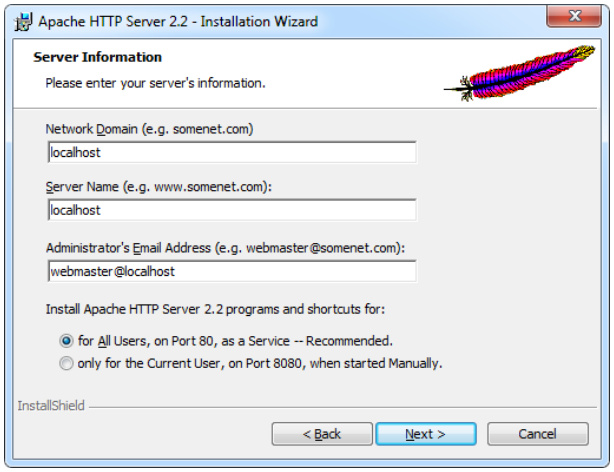
6.2 Vlastní instalace

Osobně volím instalaci s následující strukturou, pokud použijete jinou, budete muset konfiguraci podle toho upravit. Jde o umístění přímo v kořenovém adresáři, protože server Apache vyžaduje oprávnění Zobrazení obsahu složky už odtud. Můžete tedy využít i vnořené složky, ale Apache (a tím i PHP skripty – v případě modulu) budou moci zobrazit seznam souborů v těchto složkách. Názvy vnořených složek nesmí obsahovat háčky a čárky, mezera je možná.

C:\Web\htdocs
C:\Web\system\Apache
C:\Web\system\MySQL
C:\Web\system\PHP
C:\Web\system\phpMyAdmin

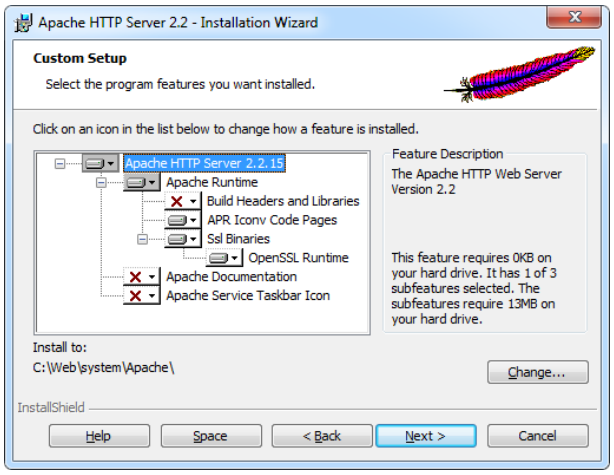
Vzhledem k tomu, že server Apache nemůže sdílet stejný port s jinou TCP/IP aplikací, měli byste zastavit či přenastavit aplikace využívající porty 80 a 443.

1. Spustíme instalaci serveru Apache (httpd-2.2.15-win32-x86-openssl-0.9.8m-r2.msi), instalační balík naleznete pod odkazem v části 6.1.
2. Přesuneme se až na stránku Server Information (Obr. 6.1), kde bude v našem případě stačit vyplnit požadované informace na localhost. Tyto informace se dají později změnit v souboru httpd.conf a jsou zobrazitelné např. přes PHP funkci phpinfo().



Obr. 6.1: Uživatelské informace.

3. Zvolíme uživatelskou (Custom) instalaci.
4. Podle obr. 6.2 nastavíme u první položky cestu „C:\Web\system\Apache“.

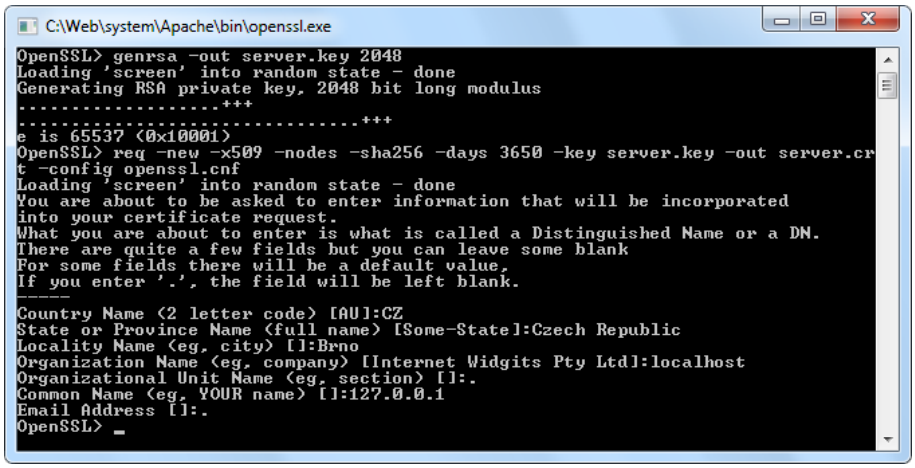


Obr. 6.2: Podrobná instalace Apache.

5. V textovém editoru otevřeme soubor „Apache\conf\httpd.conf“ a provedeme následující nahrazení. Čísla označují čísla řádků v konfiguračním souboru, jejich umístění se může s novými verzemi lišit. Tip: Pro zobrazení čísla řádku v aplikaci Poznámkový blok je nutné zrušit Zalamování řádků a povolit Stavový řádek.

036: `PHPIniDir "C:/Web/system/PHP/"`
047: `ServerTokens Prod` *(omezená identifikace serveru)*
104: `#LoadModule log_config_module modules/mod_log_config.so`
117: `LoadModule rewrite_module modules/mod_rewrite.so`
120: `LoadModule ssl_module modules/mod_ssl.so`
128: `LoadModule php5_module "C:/Web/system/PHP/php5apache2_2.dll"`
171: `ServerName localhost:80`
178: `DocumentRoot "C:/Web/htdocs"`
205: `<Directory "C:/Web/htdocs">`
225: `AllowOverride All` *(povolení konfiguračních souborů .htaccess)*
240: `DirectoryIndex index.html index.htm index.php`
325: `Alias /db "C:/Web/system/phpMyAdmin/"`
341: `<Directory "C:/Web/system/phpMyAdmin/">`
342: `AllowOverride All`
383: `AddType application/x-httpd-php .php`
384: `AddType application/x-httpd-php-source .phps`
474: `Include conf/extra/httpd-ssl.conf`

6. Otevřeme soubor „Apache\conf\extra\httpd-ssl.conf“ a v něm změníme:
   077: `DocumentRoot "C:/Web/htdocs"`
   081: `#TransferLog`
   228: `#CustomLog`
7. Vytvoříme složku „C:\Web\htdocs“.
8. V souboru „Apache\conf\openssl.cnf“ změníme následující řádek:
   233: `basicConstraints = CA:false` *(certifikační úřad nebo koncová entita)*
   A následně soubor zkopírujeme do „Apache\bin“.
9. Spustíme „Apache\bin\openssl.exe“ (Obr. 6.3) a zadáme následující 2 příkazy:
   ```
   genrsa -out server.key 2048
   req -new -x509 -nodes -sha256 -days 3650 -key server.key \
       -out server.crt -config openssl.cnf
   ```
10. Soubory server.key a server.crt z „Apache\bin“ přesuneme do „Apache\conf“.
11. PHP ve verzi VC6 Thread Safe (php-5.3.2-Win32-VC6-x86.zip) rozbalíme do „C:\Web\system\PHP“.
12. V této složce přejmenujeme soubor „php.ini-production“ na „php.ini“ a v něm provedeme následující úpravy:
    ```
    379: open_basedir = "C:/Web/htdocs/;C:/Users/Web/AppData/Local/ \
         Temp/;C:/Web/system/phpMyAdmin/"
    431: expose_php = Off
    514: error_reporting = E_ALL & ~E_NOTICE
    728: post_max_size = 100M
    ```



Obr. 6.3: Nastavení OpenSSL.

`809: extension_dir = "C:/Web/system/PHP/ext/"`

`879: upload_max_filesize = 100M`

`952: extension=php_gd2.dll` *(knihovna pro generování obrázků)*

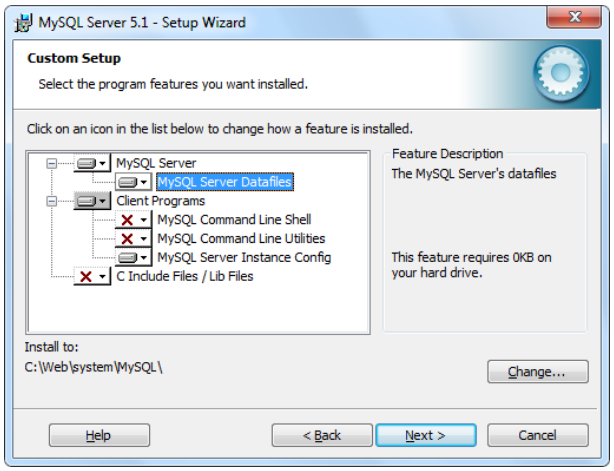
`959: extension=php_mbstring.dll` *(funkce pro multibytové řetězce, např. UTF-8)*

`963: extension=php_mysqli.dll` *(připojení k databázi MySQL)*

`996: date.timezone = Europe/Prague`

`1495: session.name = ID` *(změna názvů relací, původně PHPSESSID)*

13. Restartujeme službu „Apache2.2“ přes Ovládací panely, či rovnou vyhledáním programu „Restart“ z nabídky *Start*.
14. Spustíme instalaci serveru MySQL (mysql-essential-5.1.47-winx64.msi) a zvolíme uživatelskou (Custom) instalaci.
15. Podle obr. 6.4 nastavíme u položky MySQL server, tak u podřazené MySQL server Datafiles „C:\Web\system\MySQL“.
16. Potvrdíme dokončení instalace, spustí se průvodce MySQL Server Instance Config Wizard a registrace produktu na webu (nepovinná).
17. V následujících krocích budeme postupně volit Detailed Configuration, Server Machine, Multifunctional Database, Decision Support (DSS)/OLAP.
18. Povolení TCP/IP komunikace ponecháme beze změny.
19. Změníme kódování na Best Support For Multilingualism (podpora UTF-8).
20. Instalaci MySQL jako služby ponecháme beze změny.
21. Vytvoříme heslo pro uživatele root s administrátorskými právy pro MySQL server.
22. phpMyAdmin (phpMyAdmin-3.3.3-all-languages.zip) rozbalíme do složky „C:\Web\system\phpMyAdmin“.
23. Tam také vytvoříme soubor „config.inc.php“, do něhož vložíme následující nastavení:



Obr. 6.4: Podrobná instalace MySQL.

```
<?php
$cfg['Servers'][1]['host'] = '127.0.0.1';
$cfg['Servers'][1]['connect_type'] = 'tcp';
$cfg['Servers'][1]['extension'] = 'mysqli';
$cfg['Servers'][1]['auth_type'] = 'cookie';
$cfg['Servers'][1]['hide_db'] = '(mysql|information_schema)';
$cfg['Servers'][1]['ssl'] = TRUE;
$cfg['ForceSSL'] = TRUE;
$cfg['PmaNoRelation_DisableWarning'] = TRUE;
$cfg['ShowPhpInfo'] = TRUE;
$cfg['DefaultLang'] = 'cs-utf-8';
$cfg['DefaultCharset'] = 'utf-8';
$cfg['DefaultConnectionCollation'] = 'utf8_czech_ci';
$cfg['blowfish_secret'] = '4ad5ec1c871e74.36100050';
?>
```

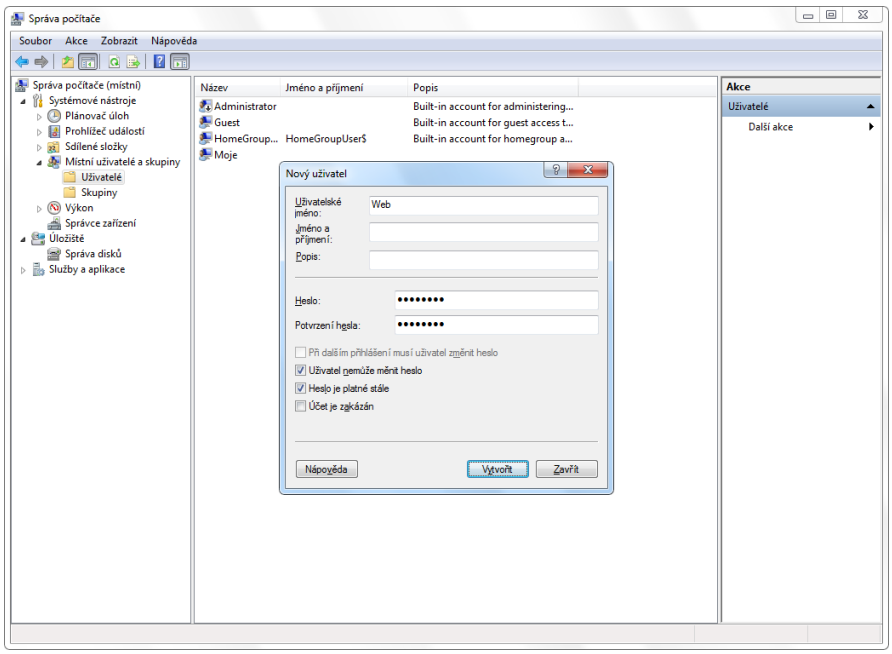
24. Proměnnou `$cfg['blowfish_secret']` naplníme nějakým jiným náhodným řetězcem, je to nastavení šifrovacího klíče pro cookies.

Tip: Můžete omezit přístup k administraci databáze na IP adresy – vytvořením souboru „.htaccess“ ve složce „system\phpMyAdmin“ s následujícím obsahem:

```
deny from all
allow from 127.0.0.1
```

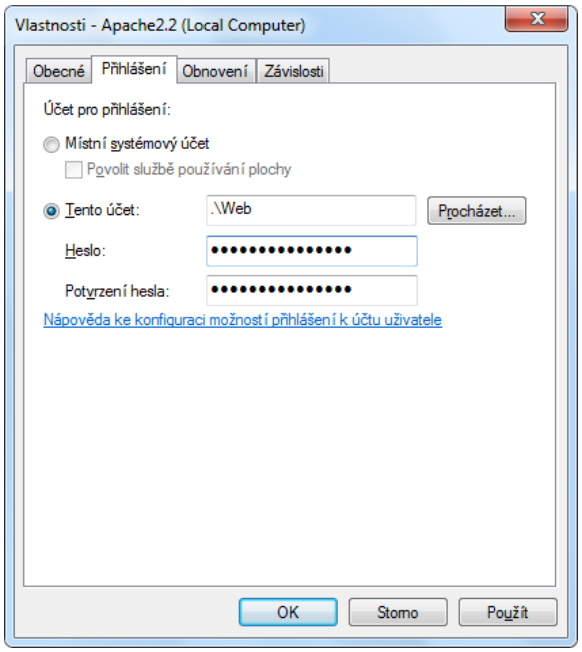
25. Nyní vytvoříme oprávnění na úrovni operačního systému (s menšími pravomocemi), aby webový server neměl tak velké oprávnění, jako uživatel pod kterým byl nainstalován (typicky administrátorská). V nabídce *Start* si otevřeme *Správa počítače* (`compmgmt.msc`) a vytvoříme nového uživatele „Web“ (Obr. 6.5) v *Místní uživa-*

telé a skupiny tak, aby nebyl členem žádné uživatelské skupiny (nebude se zobrazovat v dialogu pro přihlášení).



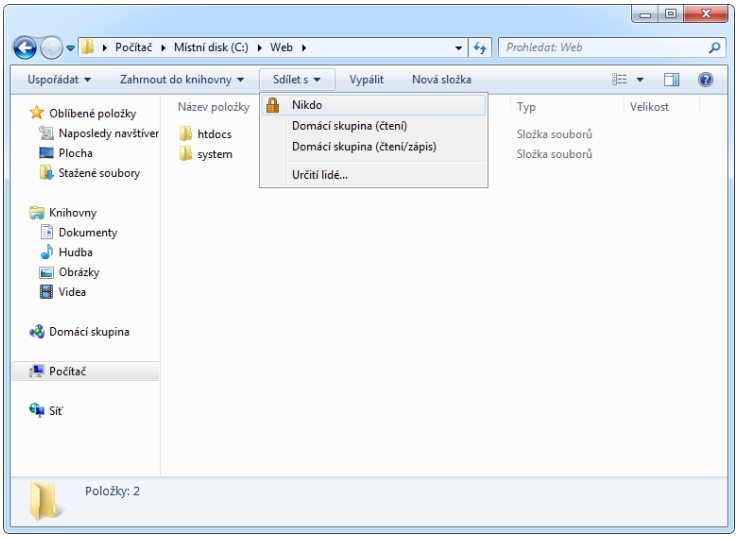
Obr. 6.5: Vytvoření uživatele Web.

26. Následně nově vytvořeného uživatele připojíme přes *Služby* (`services.msc`) službám „Apache2.2“ a „MySQL“ (Obr. 6.6).



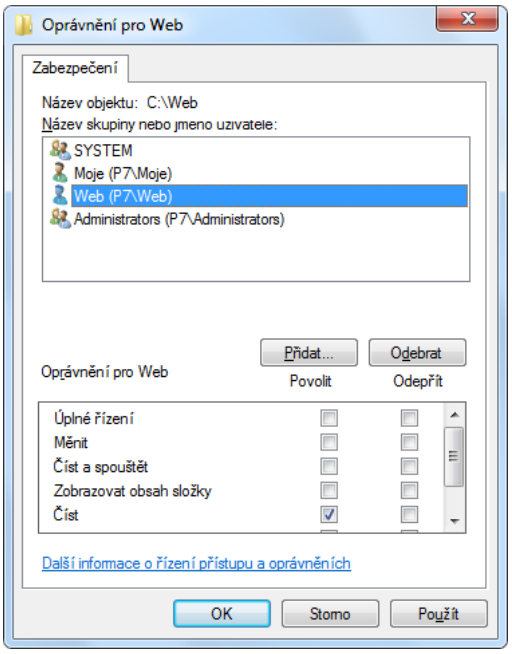
Obr. 6.6: Změna spouštění služby Apache.

27. V případě, že restartujeme služby, měly by se úspěšně spustit, i když vlastně uživatel „Web“ nemá žádná oprávnění. Ve složce „C:\Web“ (Obr. 6.7) vybereme z hlavní nabídky „Sdílet s“ a následně „Nikdo“, nyní by se už služby po restartu neměly spustit, protože nemají oprávnění pro přístup ke spouštěcím souborům.



Obr. 6.7: Omezení oprávnění pro složku Web.

28. Podle obr. 6.8 postupně vytvoříme následující oprávnění pro jednotlivé složky.
    C:\Web
        `Číst`
    C:\Web\system\Apache\bin
    C:\Web\system\Apache\modules
    C:\Web\system\MySQL\bin
    C:\Web\system\PHP
        `Číst a spouštět`
        `Zobrazovat obsah složky`
        `Číst`
    C:\Web\htdocs
    C:\Web\system\Apache\logs
    C:\Web\system\MySQL\data
        `Měnit`
        `Číst a spouštět`
        `Zobrazovat obsah složky`
        `Číst`
        `Zapisovat`
    C:\Users\Web\AppData\Local\Temp
        Odepřít `Zobrazovat obsah složky`



Obr. 6.8: Vytvoření oprávnění pro složku Web.

29. Jestliže se přihlašujete do Microsoft Windows bez zadaní hesla (bez dialogu pro přihlášení), bude po Vás systém chtít (kvůli novému uživateli) přihlašovací dialog. Pro jeho zrušení v programu *Spustit* otevřete „control userpasswords2“ a zrušte zaškrtnutí volby *Před použitím počítače musí uživatelé zadat uživatelské jméno a heslo*, následně budete vyzváni k zadání přihlašovacích údajů.
30. Tímto je instalace hotova, nejsnadnější možnost otestování všech součástí najednou je přihlášení se do správy databází přes HTTPS připojení. Adresa je v tomto případě `https://127.0.0.1/db/` (doporučuji použít „127.0.0.1“, než „localhost“), uživatelské jméno pro přihlášení je „root“ a heslo se volilo na konci instalace MySQL.

Tip: Pro využití webového serveru v internetu je většinou nutné nastavit směrování portů TCP 80 a 443 na NATu (Network Address Translation), a také povolit tyto porty na firewallu. V případě použití vestavěného firewallu v Microsoft Windows 7 je nejrychlejší vyhledat v nabídce *Start* program *Povolit program v bráně Windows Firewall* a v něm povolit „C:\Web\system\Apache\bin\httpd.exe“.

7 PRAKTICKÁ ČÁST

Na nainstalovaném webovém serveru z kapitoly 6 můžete vyzkoušet webovou aplikaci z přiloženého CD, nazvanou poměrně univerzálním názvem MyWeb. Jde o sdílení souborů uživateli uzavřené sítě (jen pro přihlášené), jejíž vývojem bylo ověřeno a ukázáno, jak se bránit před všemi webovými zranitelnostmi popsanými v této práci. Přihlášení k aplikaci bylo zvoleno trvalé (není se třeba znovu přihlašovat po každém zavření webového prohlížeče), ale pro případ přihlášení z veřejného počítače je zároveň možnost se ihned odhlásit. Trvalé přihlášení je užitečná funkce, která nezvyšuje riziko neoprávněného přístupu, protože když už má někdo přístup k PC, tak může jednoduše nainstalovat trojského koně nebo mezi PC a klávesnici zapojit HW keylogger. Snímky aplikace můžete nalézt v příloze C.

I přes snahu vytvořit tuto webovou aplikaci tak, aby byla co nejméně závislá na konfiguraci webového serveru, je nutné splnit alespoň následující podmínky: Apache 2 s funkčním HTTPS, podporou *mod_rewrite* a souborů *.htaccess*; PHP 5.3 s rozšířeními *php_gd2.dll*, *php_mbstring.dll*, *php_mysqli.dll*; MySQL 5 se zapnutým uložištěm InnoDB.

7.1 Bezpečnostní vlastnosti

- přihlašovací stránka automaticky přesměrovává na šifrované HTTPS spojení, použitý *self-signed* serverový certifikát je podepsaný SHA-256 a zaručuje až 256bitové AES šifrování (záleží na použitém webovém prohlížeči) s veřejným klíčem RSA o délce 2048 bitů,
- hesla jsou uložena jako SHA-256 haše (64 hexadecimálních znaků) se solí vloženou algoritmem HMAC a umístěnou mimo databázi – v PHP skriptu,
- po úspěšném přihlášení aplikace zaznamená IP adresu uživatele (podpora i IPv6), identifikátor jeho webového prohlížeče a čas přihlášení,
- v případě 3 neúspěšných pokusů o přihlášení se spustí ochrana CAPTCHA, jejíž výstup uživatel zapisuje dohromady s heslem do stejné kolonky. Nízký počet možných pokusů je zvolen pro zastrašující efekt při „ručním“ zkoušení, při užití robota nemá nízký počet smysl. Neúspěšné pokusy se zaznamenávají do databáze k přihlašovanému uživateli, nejsou tady závislé na cookies uživatele,
- zakázáno předávání session ID v URL, předávání je možné jen pomocí cookies,
- po úspěšném přihlášení je změněno Session ID – ochrana před Session Fixation,
- skryta přítomnost PHP na serveru – v hlavičkách odpovědí neposílá informaci o PHP, jen zmínku o použití HTTP serveru Apache (bez verze). Aplikace neodkazuje na soubory s koncovkou .php na stejné doméně a nepoužívá výchozí session identifikátor PHPSESSID (používá „ID“),

– jediná relace – při novém přihlášení dojde k odhlášení původní relace,
– ochrana proti XSS na úrovni převádění speciálních znaků na HTML entity také při vypisování z databáze, tj. má-li útočník přístup do databáze, je jeho vložení škodlivého kódu mezi data (např. k uživatelskému jménu) neškodné,
– zabezpečení cookies – HttpOnly, přenos pouze přes šifrované spojení,
– obrana před CSRF (podvržení požadavku přihlášeného uživatele) ověřováním vygenerovaného unikátního kódu u každého formuláře,
– uživatelé s moderními prohlížeči jsou ochráněni také před útokem ClickJacking, za pomoci hlavičky `X-Frame-Options: deny`,
– Session se na serveru ukládají do vlastní zabezpečené složky. Není tedy problém jako v případě sdílených webových hostingů, kde se ve výchozím nastavení ukládají Session do složky `Temp` vlastněnou operačním systémem, ke které mají přístup všichni uživatelé webového serveru,
– výpis chybových hlášení je vypnut (ukazují útočníkovi slabá místa aplikace), ale zároveň se chyby zapisují do logovacího souboru v zabezpečené složce.

Známe nedostatky

– serverový certifikát je podepsaný sám sebou (*self-signed*) pomocí SHA-256, to vyžaduje operační systém Microsoft Windows XP SP3 či novější. A protože není podepsaný známou certifikační autoritou, tak webové prohlížeče zobrazují varování, což může některé uživatele v reálném nasazení odradit,
– testovací „doména“ není zabezpečena prostřednictvím DNSSEC,
– zapomene-li se uživatel odhlásit z webové aplikace na veřejně přístupném počítači (např. škola, internetová kavárna), bude přihlášen i po uzavření webového prohlížeče do té doby, než se znovu přihlásí (např. již doma).

7.2 Vlastnosti systému

– automatická ochrana před uživateli nahrávající nežádoucí soubory – ostatní uživatelé (kteří soubor sdílí) mohou hlasovat, jestli se jim soubor líbí (či nikoliv) a uživateli, který soubor nahrál se to promítne do hodnocení systému Aura (rozsah 0–100, začíná na 50). V případě, že se uživatel dostane na hodnotu 0, nemůže nahrávat další soubory (systémově se z něj stane „host“). Nahrávání mu může opět povolit jen „admin“, on sám Auru nemá,
– 100 MiB/soubor, 1 GiB na účet (velikost účtu muže administrátor měnit). Soubory se ukládají do souborového systému (nezatěžují SQL databázi), název každého souboru je SHA-256 haš obsahu. To umožňuje do aplikace nahrát více stejných souborů s různými názvy jako jen jeden soubor v souborovém systému

a k němu více symbolických odkazů se skutečným názvem v databázi. Název souboru jako haš zároveň slouží jako další stupeň ochrany – obrana utajením názvu (64 hexadecimálních znaků), vedle dalších ochran jako jsou práva na úrovni operačního systému a omezení přístupu přes soubor *.htaccess*,
- referenční integrita databáze zaručena pomocí *Cizích klíčů* (Foreign Keys), datovou integritu zajišťují *Spouště* (Triggers),
- pro hezká URL je použit `mod_rewrite`, zhoršila se tím ale přenositelnost na jiný webový server, než je dominantní Apache,
- výstupem webové aplikace je dokument typu `text/html` (pro větší kompatibilitu) v kódování UTF-8. Využívá prvky standardů XHTML 1.0 Strict, CSS 2.1 a JavaScriptu (pro lepší *použitelnost*), je *přístupný* i v textové formě (bez CSS),
- zabráněno opakovanému odeslání formulářových dat (POST) při obnovení webové stránky, pomocí HTTP 301/303 s přesměrováním na stejnou stránku,
- oddělení procesů autentizace a autorizace.

Známe nedostatky

- webový prohlížeč Opera nepracuje s cookies na „127.0.0.1“ („localhost“ a jiné domény fungují). I přes to doporučuji používat pro účely testování „127.0.0.1“ než „localhost“, s tím je kvůli hybridní podpoře IPv4 a IPv6 v Microsoft Windows 7 více problémů,
- ve vzdálené budoucnosti, kdy pravděpodobně bude objevena kolize hašovací funkce SHA-256, by útočník mohl nahrát do webové aplikace škodlivý soubor bez toho, aby ho nasdílel jiným a tím přepsat původní soubor v souborovém systému. Čímž by najednou mohlo vzniknout, že důvěryhodná osoba (např. Administrátor) nasdílela virus, ačkoliv původně šlo o neškodný soubor,
- při nenastavení PHP direktiv *post_max_size* a *upload_max_filesize*, na mnou doporučených 100 MiB, bývá maximální velikost nahrávaného souboru 2 MiB. Další věcí je nastavení dostatečně velké paměti direktivou *memory_limit*, ta má ve výchozím nastavení sice dostatečných 128 MiB, ale provozovatelé sdíleného webového hostingu ji často omezují až na 8 MiB,
- IP adresa přihlášeného uživatele je pro kompatibilitu s nastupující IPv6 ukládána jako VARCHAR(39), tj. v databázi zabírá každý znak 1 B + 1 B navíc (celkem tedy od 8B u jednoduché IPv4, do 40B u plné IPv6). Je to tak ukládáno z důvodu neexistence současných interních funkcí PHP či MySQL pro převod IPv6 do binárního vyjádření a zpět, na které by stačilo pevných 16 B. Poznámka, pro funkčnost IPv6 je nutné mít k doméně v DNS zaveden AAAA záznam, ten je 128bitovou obdobou A záznamu pro IPv4.

8 ZÁVĚR

Zabezpečení webových aplikací je poměrně novým tématem, u něhož není v porovnání se zabezpečením počítače jako takového až tak moc ucelených a aktuálních informací, tato práce se tento stav pokusila vylepšit.

Kombinací zabezpečení `open_basedir`, oprávnění na úrovni operačního systému a omezení odesílání informací o použitém programovém vybavení, byl vytvořen jeden z nejzabezpečenějších návodů na instalaci webového serveru (Apache 2.2, PHP 5.3, MySQL 5.1) na platformě Microsoft Windows 7.

V praktické části byl vytvořen systém pro správu souborů, které mohou uživatelé sdílet mezi sebou (v uzavřené komunitě) a k nim přidávat komentáře. Účelem bylo vytvořit a ověřit zabezpečení této aplikace. Kromě základních ochraných mechanismů bylo použito také šifrované spojení a byla vytvořena speciální automatická ochrana před uživateli nahrávající, podle ostatních uživatelů, nežádoucí soubory. Webová aplikace by na reálné nasazení potřebovala uživatelské testování použitelnosti, čímž by byly odstraněny chyby zpětnou vazbou přímo od reálných uživatelů, jenž nebyly nalezeny při tak krátkém vývoji.

Závěrem bych chtěl touto formou upozornit začínající vývojáře, že v zabezpečení webových aplikací je třeba udělat kompromis mezi bezpečností a mírou obtěžování uživatelů, ti jsou na nadstandardní prvky (certifikát na straně klienta, opisování CAPTCHA kódu apod.) velmi citliví a mohli by přejít ke konkurenční službě, která to po nich nevyžaduje. Podobně tak je otázkou, jestli vyžadovat pravidelnou povinnou obměnu hesla anebo vyžadovat jeho minimální délku, je potom reálnou hrozbou, že uživatelé si píší hesla na papírky apod. Pamatujte, že zabezpečení není jednorázový úkol, ale trvalý proces.

REFERENCE

[1] SOKOL, T.; SMEJKAL V. *Postih počítačové kriminality podle nového trestního zákona* [online]. 2009 [cit. 2010-05-10]. Dostupný z WWW: <http://pravniradce.ihned.cz/c4-10077480-37865090-F00000_d-postih-pocitacove-kriminality-podle-noveho-trestniho-zakona>.

[2] *The website of the world's first-ever web server* [online]. c2008 [cit. 2009-10-11]. Dostupný z WWW: <http://info.cern.ch/>.

[3] *The World Wide Web project* [online]. [2009] [cit. 2009-11-06]. Dostupný z WWW: <www.w3.org/History/19921103-hypertext/hypertext/WWW/TheProject.html>.

[4] HUSEBY, S. H. *Zranitelný kód* 1. vyd. Brno: Computer Press, 2006. 208 s. ISBN 80-251-1180-6.

[5] *HTTP protokol - požadavky a odpovědi* [online]. [2005] [cit. 2009-11-22]. Dostupný z WWW: <http://http.stylove.com/>.

[6] *Zabezpečení session proměnných* [online]. [2005] [cit. 2009-11-22]. Dostupný z WWW: <http://php.vrana.cz/zabezpeceni-session-promennych.php>.

[7] FRANK, T. *Session variables without cookies* [online]. 2008 [cit. 2010-03-15]. Dostupný z WWW: <http://www.thomasfrank.se/sessionvars.html>.

[8] *How to save session values in JavaScript* [online]. c2003-2010 [cit. 2010-03-15]. Dostupný z WWW: <http://www.daniweb.com/forums/thread19283.html>.

[9] ZBIEJCZUK, A. *Web 2.0 – charakteristika a služby.* Fakulta sociálních studií, Masarykova univerzita v Brně, 2007. 71 s. Diplomová práce.

[10] *Wikipedie : otevřená encyklopedie* [online]. [2009] [cit. 2009-11-06]. Dostupný z WWW: <http://cs.wikipedia.org/>.

[11] *YouTube : Broadcast Yourself* [online]. c2009 [cit. 2009-11-06]. Dostupný z WWW: <http://www.youtube.com>.

[12] *Facebook* [online]. c2009 [cit. 2009-11-06]. Dostupný z WWW: <http://cs-cz.facebook.com/>.

[13] *Twitter* [online]. c2009 [cit. 2009-11-06]. Dostupný z WWW: <http://twitter.com/>.

[14] *Digg : The Latest News Headlines, Videos and Images* [online]. c2009 [cit. 2009-11-06]. Dostupný z WWW: <http://digg.com/>.

[15] *Last.fm : Listen to internet radio and the largest music catalogue online* [online]. c2009 [cit. 2009-11-06]. Dostupný z WWW: <http://www.last.fm>.

[16] *MySpace* [online]. c2003-2009 [cit. 2009-11-06]. Dostupný z WWW: <http://www.myspace.com>.

[17] *LinkedIn : Relationships Matter* [online]. c2009 [cit. 2009-11-06]. Dostupný z WWW: <http://www.linkedin.com>.

[18] DVOŘÁK, J. *Novinky v HTML 5.* Connect!, roč. 14, č. 12, s. 42–43. ISSN 1211-3085.

[19] *SPDY: An experimental protocol for a faster web* [online]. [2009] [cit. 2009-11-14]. Dostupný z WWW: <http://sites.google.com/a/chromium.org/dev/spdy/spdy-whitepaper>.

[20] *Top 25 Technology Predictions* [online]. 2009 [cit. 2010-02-16]. Dostupný z WWW: <http://newsroom.cisco.com/dlls/2009/ekits/Top25_Technology_Predictions.pdf>.

[21] *Bezpečné přihlašování uživatelů* [online]. 2006 [cit. 2010-01-25]. Dostupný z WWW: <http://www.root.cz/clanky/bezpecne-prihlasovani-uzivatelu/>.

[22] BITTO, O. *Ukryto pod rouškou X.509. LUPA* [online]. 2005 [cit. 2009-11-11]. Dostupný z WWW: <http://www.lupa.cz/clanky/ukryto-pod-rouskou-x-509/>.

[23] *Secure Site SSL Services : Compare Features of SSL Certificates from VeriSign, Inc.* [online]. c1995-2009 [cit. 2009-11-06]. Dostupný z WWW: <http://www.verisign.com/ssl/buy-ssl-certificates/secure-site-services/index.html>.

[24] *GeoTrust : Compare SSL Certificates* [online]. [2009] [cit. 2009-11-06]. Dostupný z WWW: <http://www.geotrust.com/ssl/compare-ssl-certificates.html>.

[25] *SSL digital certificates with extended validation from thawte the global SSL certificate authority* [online]. c1995-2009 [cit. 2009-11-06]. Dostupný z WWW: <http://www.thawte.com/pricing/>.

[26] *RSA Laboratories : TWIRL and RSA Key Size* [online]. 2003 [cit. 2010-03-20]. Dostupný z WWW: <http://www.rsa.com/rsalabs/node.asp?id=2004>.

[27] MAJ, A. *Apache 2 with SSL/TLS: Step-by-Step* [online]. 2005 [cit. 2009-11-06]. Dostupný z WWW: <http://www.securityfocus.com/infocus/1818>.

[28] THOMAS, M. T. *Zabezpečení počítačových sítí.* 1. vyd. Brno: Computer Press, 2005. 338 s. ISBN 80-251-0417-6.

[29] *Bezpečnost ve webových aplikacích* [online]. 2010 [cit. 2010-05-09]. Dostupný z WWW: <http://kore.fi.muni.cz:5080/wiki/index.php/Bezpe%C4%8Dnost_ve_webov%C3%BDch_aplikac%C3%ADch>.

[30] VALÁŠEK, M. *HMAC - Hash Message Authentication Code* [online]. 2007 [cit. 2010-01-25]. Dostupný z WWW: <http://www.aspnet.cz/Articles/146-hmac-hash-message-authentication-code.aspx>.

[31] KMENT, V. *Soumrak SHA-1, ohrožení elektronických podpisů i rozpočtů IT. LUPA* [online]. 2009 [cit. 2009-10-05]. Dostupný z WWW: <http://www.lupa.cz/clanky/soumrak-sha-1-ohrozeni-el-podpisu-i-rozpoctu-it/>.

[32] *The Official CAPTCHA Site* [online]. c2000-2009 [cit. 2009-12-13]. Dostupný z WWW: <http://www.captcha.net/>.

[33] *Breaking Gmail's Audio Captcha* [online]. 2008 [cit. 2009-12-13]. Dostupný z WWW: <http://vimeo.com/1301905>.

[34] *reCAPTCHA: Stop Spam, Read Books* [online]. 2009 [cit. 2010-05-09]. Dostupný z WWW: <http://recaptcha.net/>.

[35] ŠŤASTNÝ, P. *Útoky s využitím protokolu DNS* [online]. 2007 [cit. 2009-12-07]. Dostupný z WWW: <http://www.pweb.cz/a/38/utoky-s-vyuzitim-protokolu-dns-1-dns-spoofing-cache-poisoning.html>.

[36] *O DNSSEC* [online]. c2009 [cit. 2009-10-05]. Dostupný z WWW: <http://www.nic.cz/dnssec/>.

[37] *Google Public DNS* [online]. c2009 [cit. 2009-12-13]. Dostupný z WWW: <http://code.google.com/intl/cs-CZ/speed/public-dns/>.

[38] *OpenDNS* [online]. c2009 [cit. 2009-12-13]. Dostupný z WWW: <http://www.opendns.com/>.

[39] GRUDL, D. *Escapování - definitivní příručka* [online]. 2009 [cit. 2010-05-09]. Dostupný z WWW: <http://phpfashion.com/escapovani-definitivni-prirucka>.

[40] VEČEŘA, Z. *Jak na to: SQL injection, magic_quotes_gpc, addslashes() a stripslashes()* [online]. 2009 [cit. 2010-05-09]. Dostupný z WWW: <http://blog.zdenekvecera.cz/item/jak-na-to-sql-injection-magic_quotes_gpc-addslashes-a-stripslashes>.

[41] VRÁNA, J. *Vázání proměnných v MySQLi* [online]. 2006 [cit. 2010-05-09]. Dostupný z WWW: <http://php.vrana.cz/vazani-promennych-v-mysqli.php>.

[42] SHIFLETT, CH. *addslashes() Versus mysql_real_escape_string()* [online]. 2006 [cit. 2010-05-09]. Dostupný z WWW: <http://shiflett.org/blog/2006/jan/addslashes-versus-mysql-real-escape-string>.

[43] FERSCHMANN, P. *Bezpečnost na webu - přehled útoků na webové aplikace* [online]. 2008 [cit. 2009-12-13]. Dostupný z WWW: <http://zdrojak.root.cz/clanky/prehled-utoku-na-webove-aplikace/>.

[44] TICHÝ, J. *Předávání SID pomocí cookies* [online]. 2009 [cit. 2010-05-09]. Dostupný z WWW: <http://www.phpguru.cz/clanky/predavani-sid-pomoci-cookies>.

[45] VRÁNA, J. *ClickJacking* [online]. 2009 [cit. 2010-05-09]. Dostupný z WWW: <http://php.vrana.cz/clickjacking.php>.

[46] *YouTube : Webcam ClickJacking* [online]. 2008 [cit. 2010-05-09]. Dostupný z WWW: <http://www.youtube.com/watch?v=gxyLbpldmuU>.

[47] HASSMAN, M. *Ukázka clickjackingu* [online]. 2009 [cit. 2010-05-09]. Dostupný z WWW: <http://i.iinfo.cz/urs-att/clickjack-123565951275948.html>.

[48] MITNICK, K.; SIMON, W. *Umění klamu* HELION, 2003. 348 s. ISBN 83-7361-210-6.

[49] DOČEKAL, D. *Jak se dělá phishing* [online]. 2008 [cit. 2009-12-13]. Dostupný z WWW: <http://www.lupa.cz/clanky/jak-se-dela-phishing/>.

[50] POSEY, B. *How Spyware And The Weapons Against It Are Evolving* [online]. 2004 [cit. 2009-11-06]. Dostupný z WWW: <http://www.windowsecurity.com/articles/Spyware-Evolving.html>.

[51] *Lavasoft Ad–Aware* [online]. c2009 [cit. 2009-11-06]. Dostupný z WWW: <http://www.lavasoft.com/products/ad_aware.php>.

[52] *Spybot – Search & Destroy* [online]. c2000-2009 [cit. 2009-11-06]. Dostupný z WWW: <http://www.safer-networking.org/cz/spybotsd/index.html>.

[53] *CZ.NIC : Internationalized domain names* [online]. c2009 [cit. 2009-11-10]. Dostupný z WWW: <http://háčkyčárky.cz/page/451/>.

[54] *IDN aréna : Co to je IDN?* [online]. c2009 [cit. 2009-11-28]. Dostupný z WWW: <http://www.idnarena.cz/inpage/co-je-to-idn/>.

[55] *DNS útok podle Kaminského* [online]. 2008 [cit. 2009-12-07]. Dostupný z WWW: <http://blog.nic.cz/2008/08/08/dns-utok-podle-kaminskeho/>.

[56] *PostgreSQL : The world's most advanced open source database* [online]. c1996-2009 [cit. 2009-11-06]. Dostupný z WWW: <http://www.postgresql.org/>.

[57] *Index of /dist/httpd/binaries/win32* [online]. [2010] [cit. 2010-03-10]. Dostupný z WWW: <http://www.apache.org/dist/httpd/binaries/win32/>.

[58] *Download MySQL Community Server* [online]. c2010 [cit. 2010-05-20]. Dostupný z WWW: <http://dev.mysql.com/downloads/mysql/>.

[59] *PHP For Windows : Binaries and sources Releases* [online]. c2001-2010 [cit. 2010-03-10]. Dostupný z WWW: <http://windows.php.net/download/#php-5.3-ts-VC6-x86>.

[60] *phpMyAdmin : Download* [online]. c2003-2010 [cit. 2010-05-11]. Dostupný z WWW: <http://www.phpmyadmin.net/home_page/downloads.php>.

SEZNAM ZKRATEK

AAA	Authentication, Authorization and Accounting
AES	Advanced Encryption Standard
API	Application Programming Interface
ASP	Active Server Pages
CA	Certification Authority
CAPTCHA	Completly Automated Public Turing test to tell Computers and Humans Apart
CERN	European Organization for Nuclear Research
CSRF	Cross-Site Request Forgery
CSS	Cascading Style Sheets
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
EV	Extended Validation
GNU	GNU's Not Unix!
HMAC	Hash-based Message Authentication Code
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDN	Internationalized Domain Names
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISP	Internet Service Provider

JS	JavaScript
MD5	Message-Digest algorithm 5
MITM	Man In The Middle
MySQLi	MySQL Improved
NAT	Network Address Translation
PC	Personal Computer
PHP	Hypertext Preprocessor
PKI	Public Key Infrastructure
RSA	Rivest, Shamir, Adleman
SHA	Secure Hash Algorithm
SNI	Server Name Indication
SPDY	SPeeDY
SQL	Structured Query Language
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TTL	Time To Live
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VBS	Visual Basic Scripting
W3C	World Wide Web Consortium
WAMP	Windows, Apache, MySQL, PHP
WWW	World Wide Web
XHTML	Extensible HyperText Markup Language
XSS	Cross-Site Scripting

SEZNAM PŘÍLOH

A ÚTOK NA HESLA HRUBOU SILOU

Délka	Kombinací [mil.]	Velikost Rainbow tables [GiB]		
		MD5	SHA1	SHA256
5	12	0	0	0
6	309	5	6	9
7	8 032	120	150	239
8	208 827	3 112	3 890	6 224
9	5 429 504	80 906	101 132	161 812
10	141 167 096	2 103 554	2 629 442	4 207 107

Tab. A.1: Rainbow tables: Znaky z množin a–z (celkem 26 znaků).

Délka	Kombinací [mil.]	Velikost Rainbow tables [GiB]		
		MD5	SHA1	SHA256
5	60	1	1	2
6	2 177	32	41	65
7	78 364	1 168	1 460	2 335
8	2 821 110	42 038	52 547	84 076
9	101 559 957	1 513 361	1 891 702	3 026 723
10	3 656 158 440	54 481 006	68 101 258	108 962 013

Tab. A.2: Rainbow tables: Znaky z množin a–z, 0–9 (celkem 36 znaků).

Délka	Kombinací [mil.]	Velikost Rainbow tables [GiB]		
		MD5	SHA1	SHA256
5	916	14	17	27
6	56 800	846	1 058	1 693
7	3 521 615	52 476	65 595	104 952
8	218 340 106	3 253 521	4 066 901	6 507 042
9	13 537 086 546	201 718 309	252 147 886	403 436 617
10	839 299 365 868	12 506 535 141	15 633 168 926	25 013 070 281

Tab. A.3: Rainbow tables: Znaky z množin a–z, A–Z a 0–9 (celkem 62 znaků).

B OBSAH PŘILOŽENÉHO CD

Konfigurační soubory webového serveru

Pro návod z kapitoly 6.

Zdrojové texty webové aplikace

Návod na jejich zprovoznění:

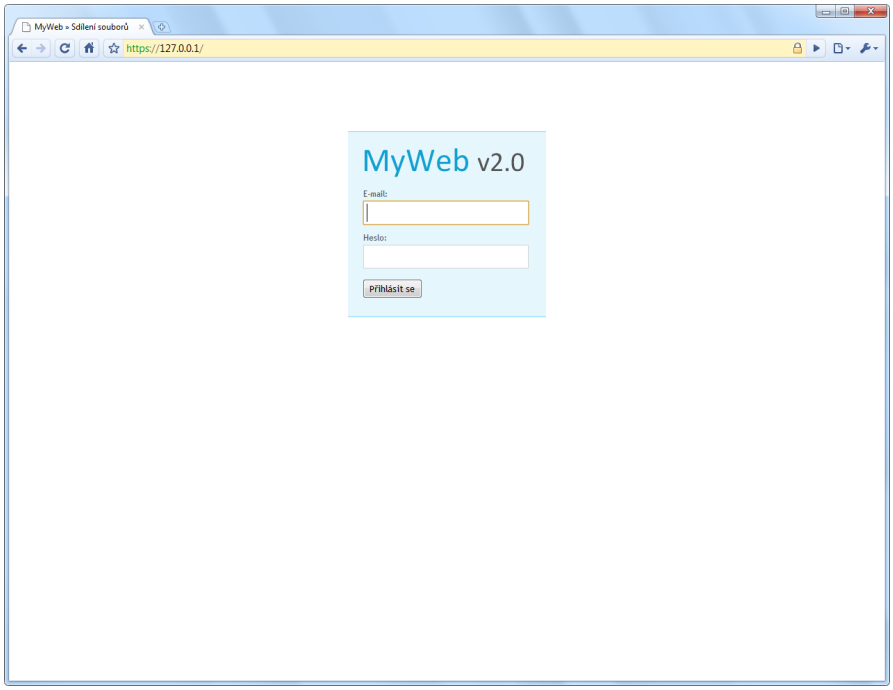
1. Importujte (např. pomocí nástroje phpMyAdmin) do předem vytvořené databáze (řazení `utf8_czech_ci`) obsah souboru „skryte\databaze.sql“.
2. V souboru „index.php“ nastavte připojení k databázovému serveru MySQL.
3. Nyní se můžete přihlásit, e-mail: `root@example.com`, heslo: `root`.

Snímky vytvořené webové aplikace

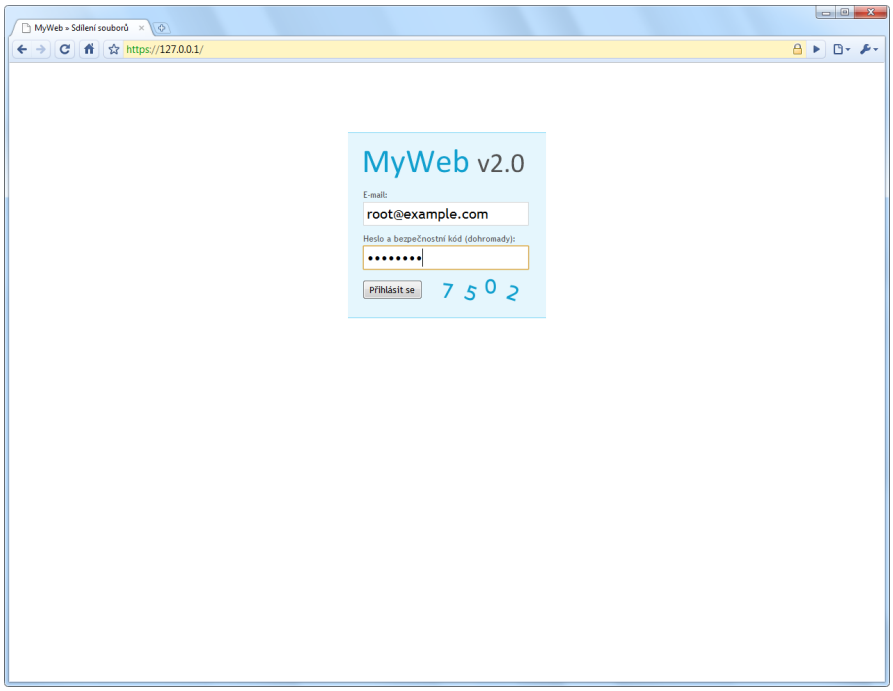
Ve vyšším rozlišení, pro možnost přiblížení detailů či malého textu.

Elektronická verze této práce

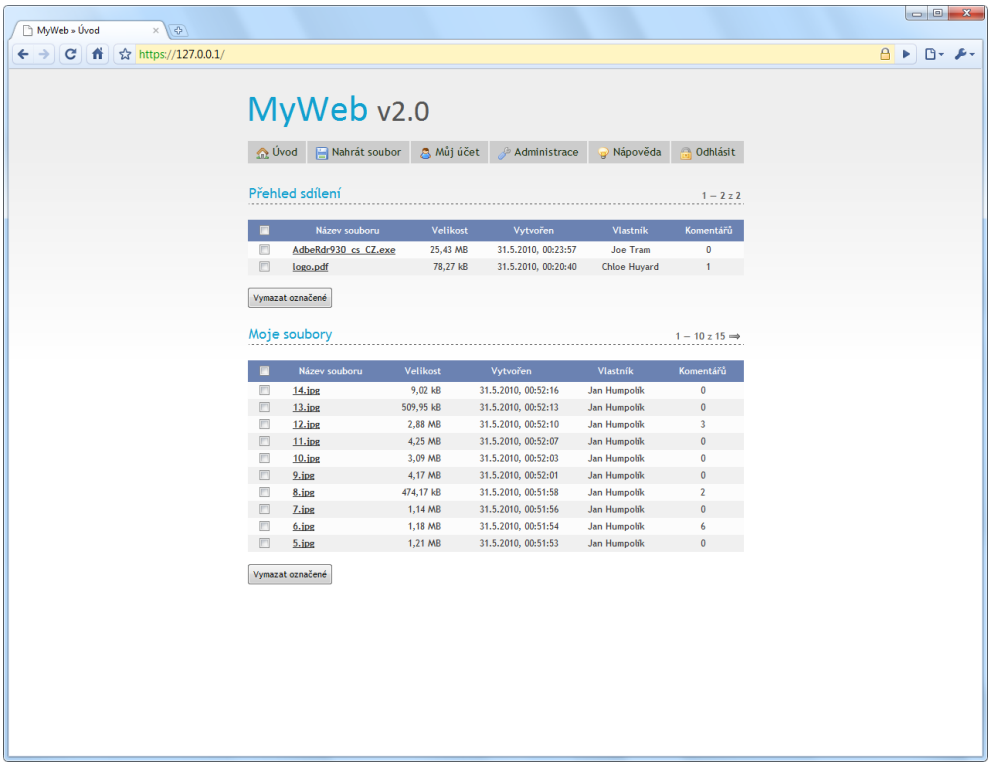
C SNÍMKY APLIKACE



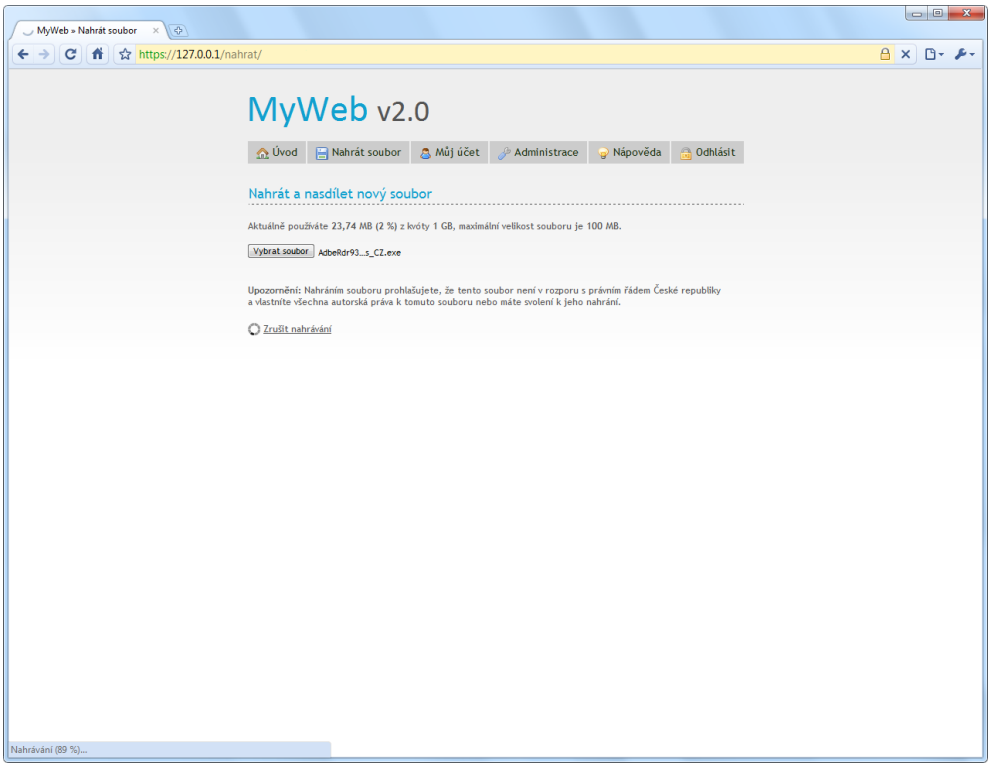
Obr. C.1: Úvodní přihlášení do aplikace.



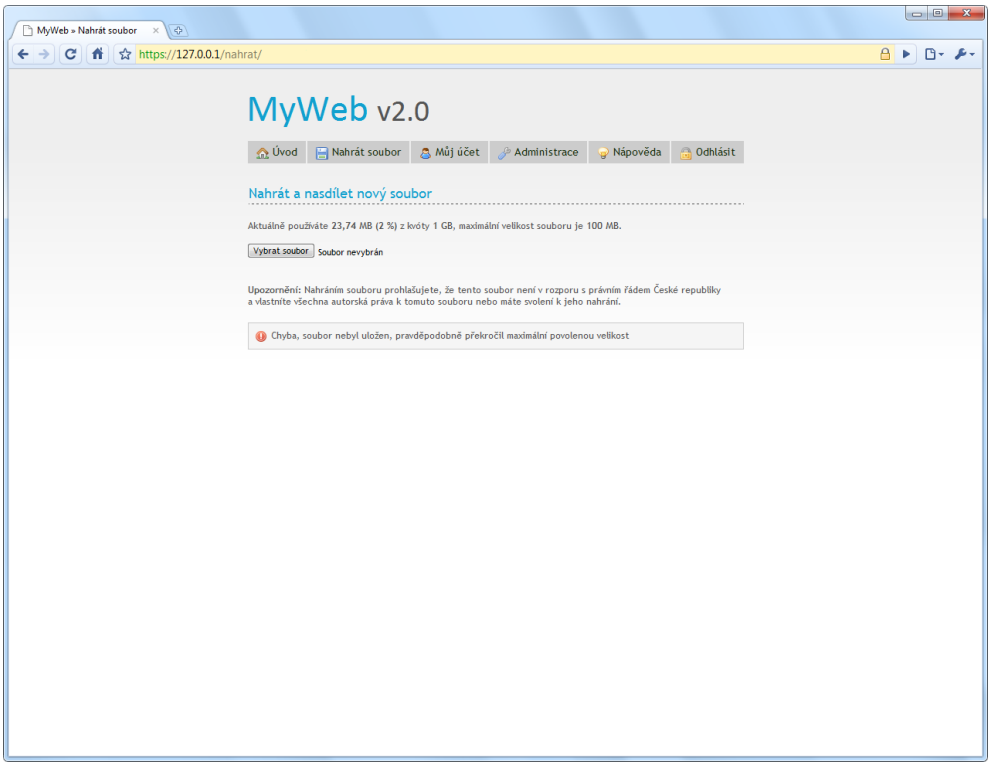
Obr. C.2: Přihlášení po 3 neúspěšných pokusech (systém CAPTCHA).



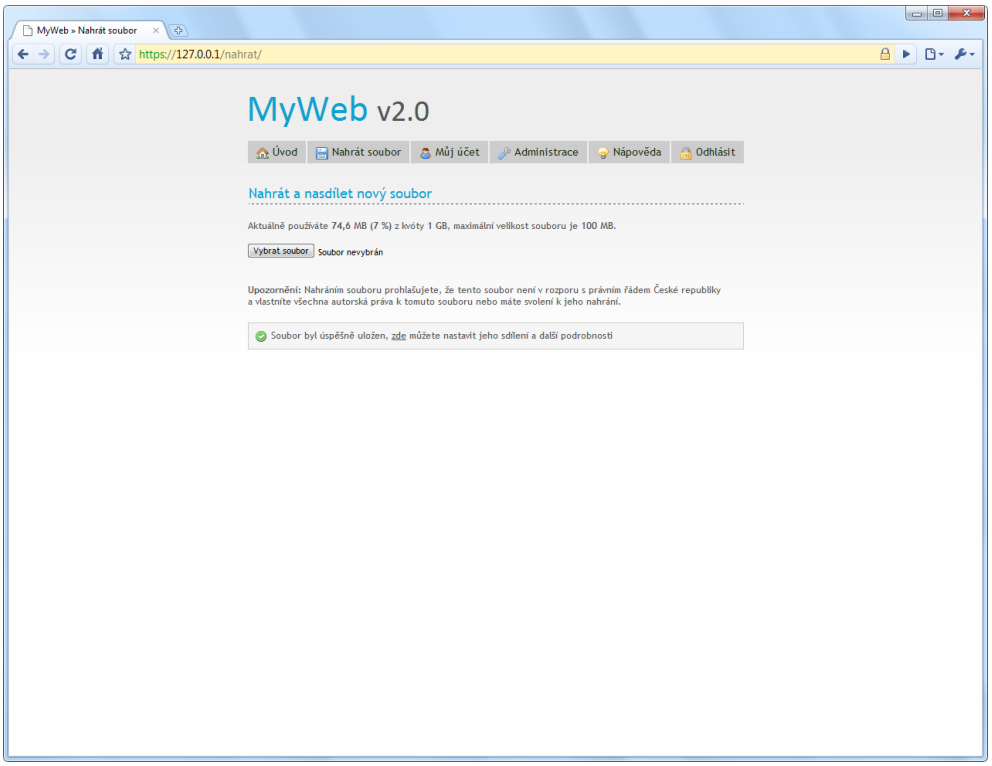
Obr. C.3: Úvodní stránka vytvořené webové aplikace.



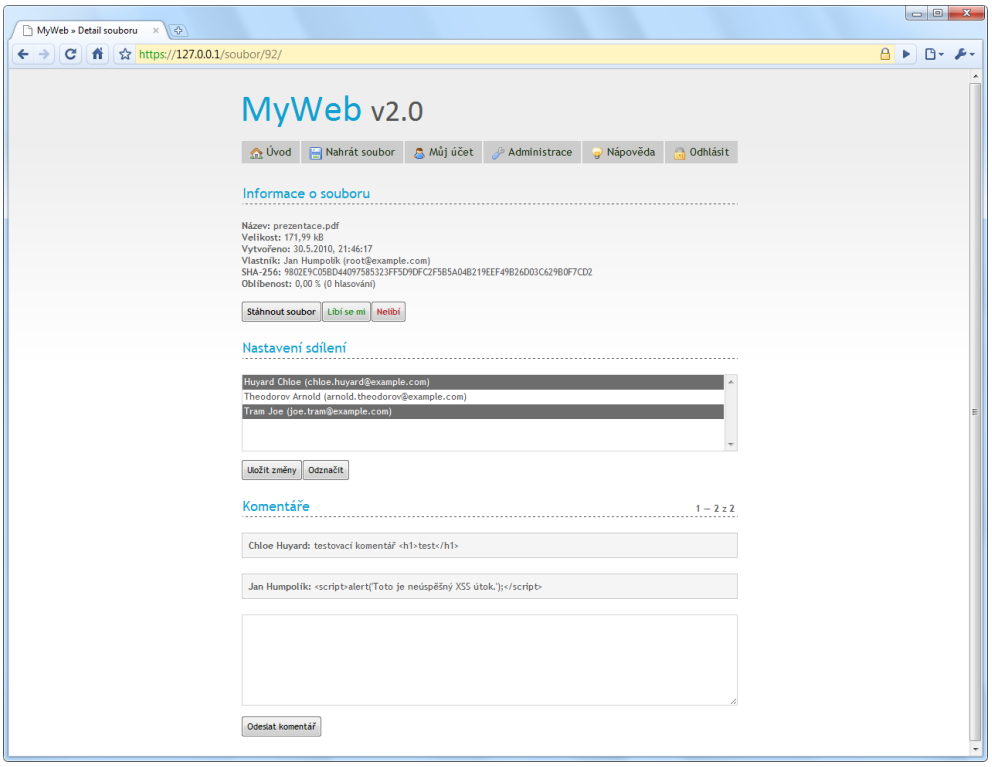
Obr. C.4: Průběh nahrávání souboru.



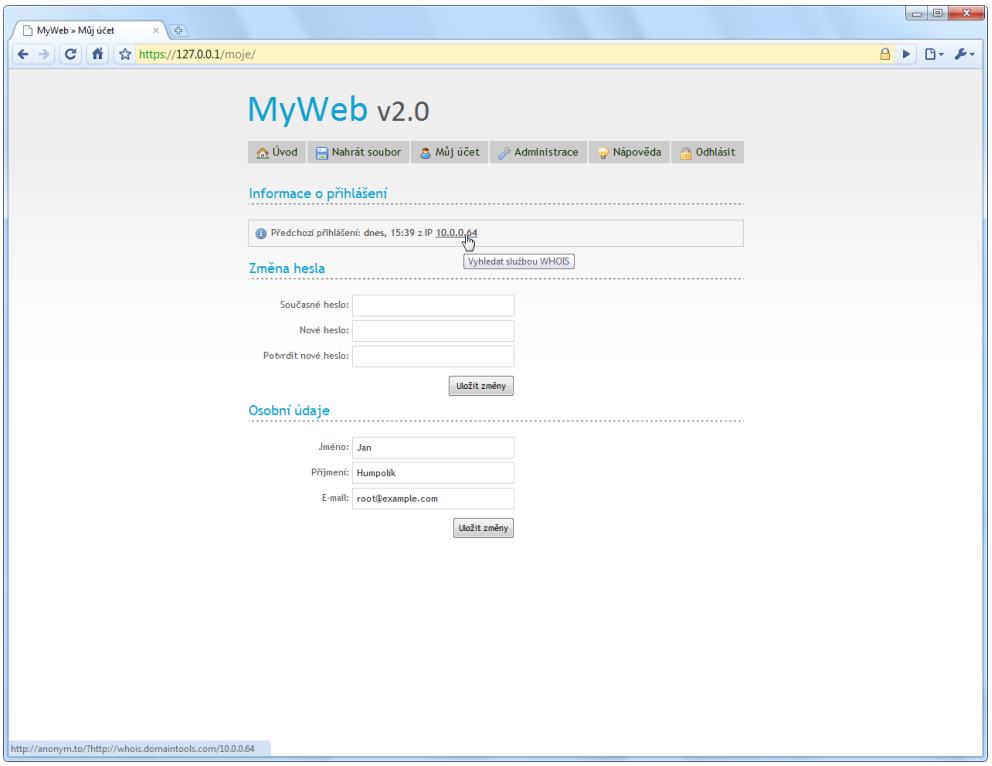
Obr. C.5: Chyba v nahrávání souboru, překročena maximální povolená velikost.



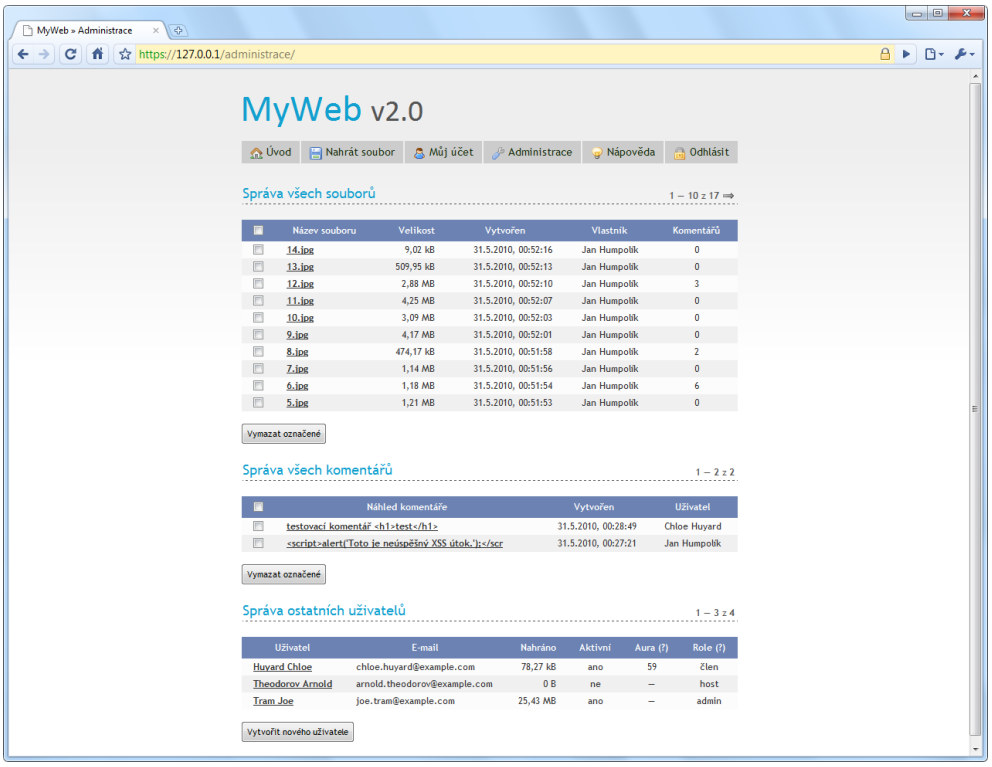
Obr. C.6: Úspešné nahrání souboru.



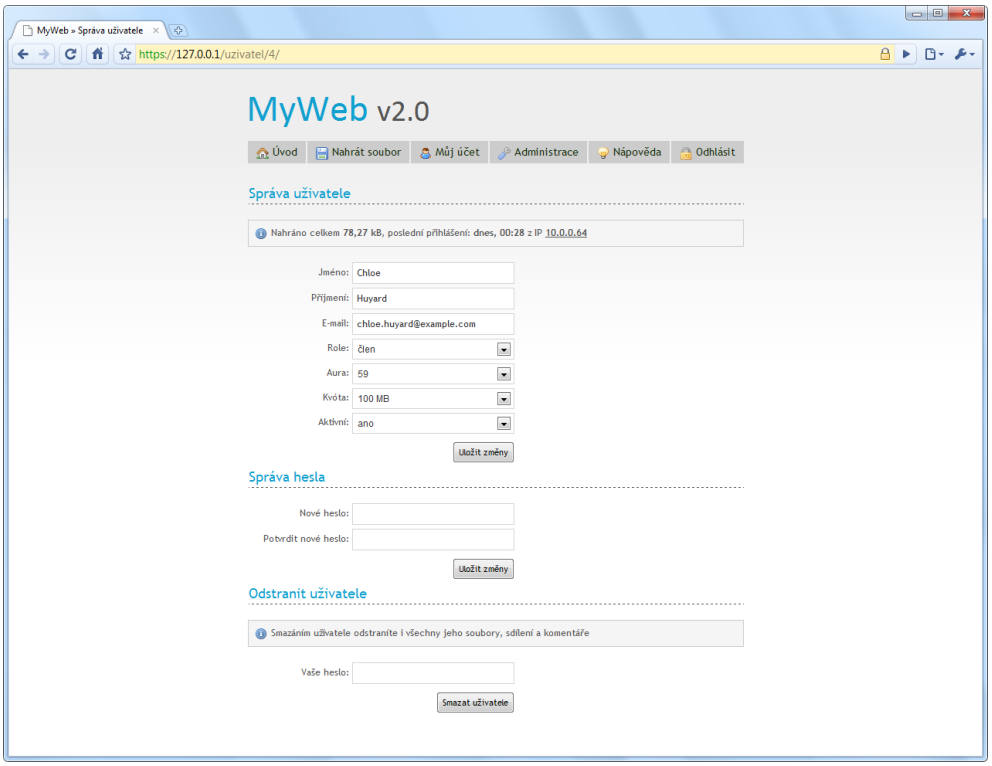
Obr. C.7: Detail souboru pro nastavení sdílení a přidání komentáře.



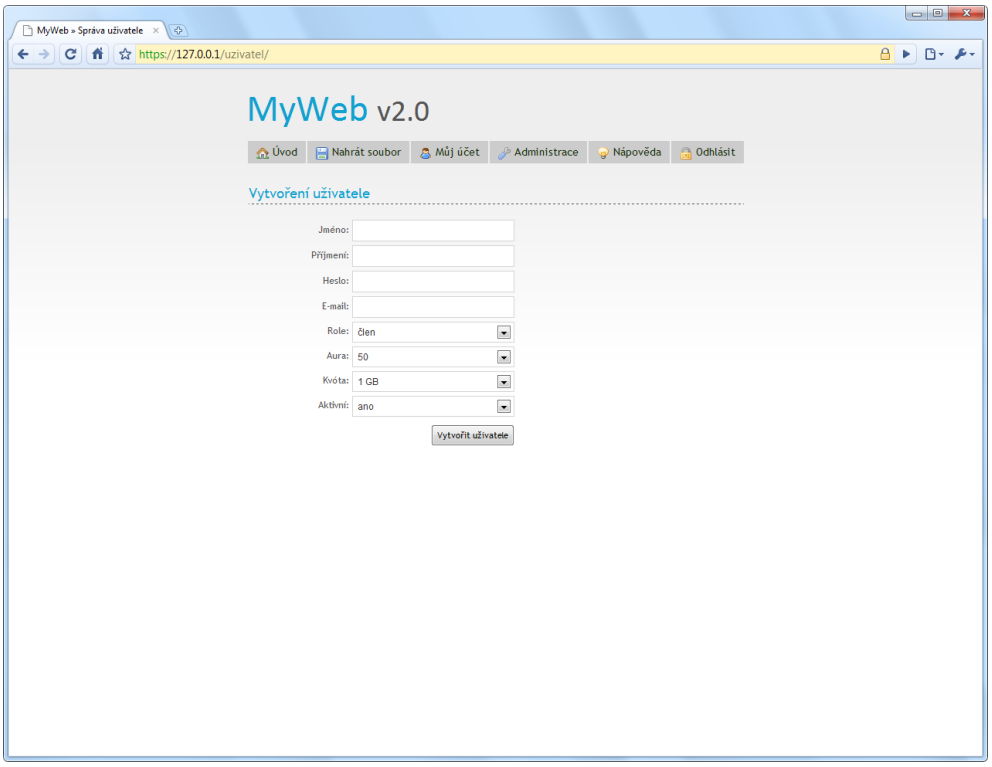
Obr. C.8: Nastavení pro přihlášeného uživatele.



Obr. C.9: Administrace, přístupná jen pro roli „admin“.

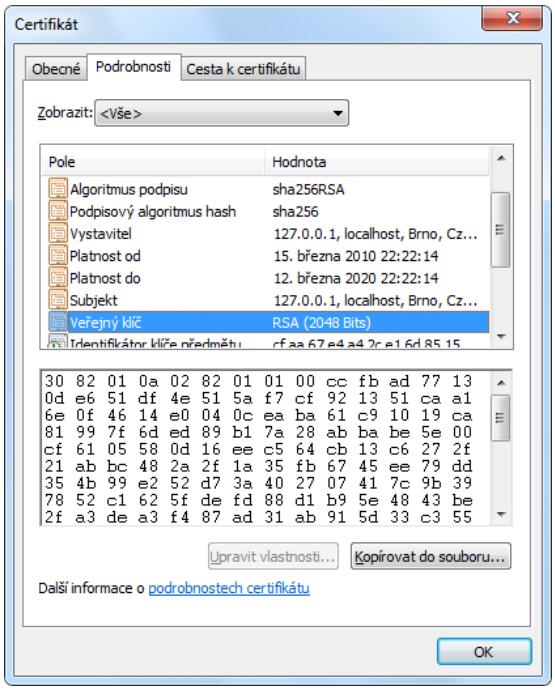


Obr. C.10: Správa uživatele, přístupná jen pro roli „admin“.

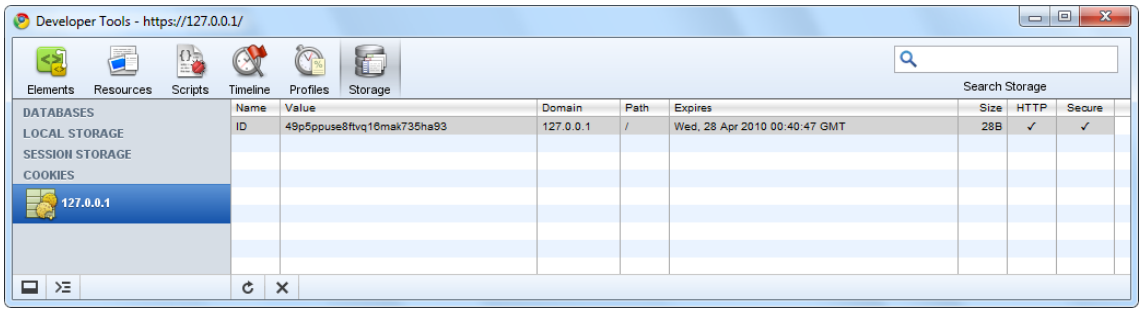


Obr. C.11: Vytvoření uživatele, přístupné jen pro roli „admin“.

Obr. C.12: Nápověda, přístupná pro všechny uživatele.



Obr. C.13: Použitý *self-signed* serverový certifikát RSA-2048/SHA-256.



Obr. C.14: Zabezpečení cookies (HttpOnly, přenos pouze přes šifrované spojení).

```
HTTP/1.1 200 OK
Date: Mon, 22 Mar 2010 21:01:18 GMT
Server: Apache
X-Frame-Options: deny
Set-Cookie: ID=efq32297u0j5j1s2rtptj91ci7; path=/; domain=127.0.0.1; secure; HttpOnly
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 847
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

Obr. C.15: HTTP hlavičky serveru, obrana proti ClickJackingu a skryté verze SW.