# BEZPEČNOSŤ WEBOVÝCH INFORMAČNÝCH SYSTÉMOV

# SECURITY OF WEB BASED INFORMATION SYSTEMS

**Martin DRLÍK - Peter ŠVEC**

***In the age of information, security is much more important then ever before. Today's information systems contain a great amount of sensitive information. Therefore these systems are open to attacks. This article describes basic steps on how to secure web based information systems. System security is a common task consisting of security servers and their services and strong application development. We are focusing on the most important security faults found in web based information system development.***

## 1 Úvod

Pri analyzovaní úrovne zabezpečenia informačného systému si musíme postaviť otázku, kde sú jeho najcitlivejšie miesta. Je dôležité si uvedomiť, že problém zabezpečenia je komplexný a nesmieme sa zameriavať iba na zabezpečenie jednej časti aplikácie, ale aj na bezpečnosť servera a jeho ďalších služieb a aplikácií.

Musíme si tiež uvedomiť, odkiaľ budú prichádzať útoky na našu aplikáciu. Je všeobecne známe, že najviac útokov prichádza z vlastnej spoločnosti, z vlastnej siete. Druhá veľká skupina útokov sú amatéri, ktorý skúšajú rôzne aplikácie, ktorými sa dá útočiť a treťou najnebezpečnejšou a najmenšou skupinou sú profesionáli, ktorí vedia, čo robia.

Katedra informatiky UKF v Nitre použila pri vývoji svojho informačného systému skriptovací jazyk PHP v kombinácii s webových serverom Apache a databázových serverom MySQL. Za operačný systém servera sme si zvolili unixový systém FreeBSD. Tento článok poukazuje na najzákladnejšie bezpečnostné riziká, ktorým sa treba vyhnúť a neslúži ako návod na prienik do iných informačných systémov.

## 2 Útok priamo na služby servera

### 2.1 Pokus o získanie shell prístupu

Najčastejším útokom je pokus o získanie shell prístupu. Útok je ľahké spozorovať, pretože je signalizovaný hádaním prihlasovacích mien a hesiel. Ďalšími metódami ako získať shell prístup, je využitie chyby v nejakej aplikácií (tzv. diera).

Shell prístup umožňuje získať veľké množstvo informácií o serveri a o službách, ktoré server poskytuje, o jeho hardvérovej konfigurácii, o používateľských účtoch. Všetky tieto poznatky sa zbierajú za účelom získania účtu superpoužívateľa (roota). Po nadobudnutí účtu roota získava útočník plnú kontrolu nad systémom. Je len na ňom, ako s túto moc využije. Jediným príkazom môže zmazať všetky údaje v systéme, čo je však celkom zbytočné, pretože väčšina údajov je pravidelne zálohovaná a vymazaním údajov sa útočník zbytočne odhalí a prinúti správcu zvýšiť bezpečnostné opatrenia a odreže si tým cestu. Musíme si uvedomiť, že prienikom na ľubovoľný server vo vnútornej sieti, sa útočník dostáva do pozície dôveryhodného zamestnanca, teda je súčasťou vnútornej siete.

Spôsob ochrany proti uhádnutiu hesla je v disciplíne používateľov. Je dôležité ich naučiť, že musia používať bezpečné heslá. To znamená, že v hesle sa musia vyskytovať kombinácie veľkých a malých písmen, číslic a špeciálnych znakov. Doporučuje sa používať dostatočne

dlhé heslá a pravidelne ich meniť. Pri zmene hesla sa nové, sa toto heslo nesmie na staré veľmi podobať. Prvých 8 znakov sa vôbec nesmie zhodovať vôbec.

Spôsob ochrany pri útoku cez dieru aplikácie je zasa v disciplíne správcu systému. Pravidelná aktualizácia aplikácií, aplikovanie bezpečnostných záplat a správna konfigurácia rapídne znižuje riziko napadnutia.

### 2.2 Útoky na služby bežiace na servery

Služby, ktoré bežia na serveri, môžeme podľa spôsobu prístupu k nim rozdeliť do troch skupín:

- služby prístupné z Internetu
- služby prístupné z lokálnej siete
- služby prístupné len priamo zo servera samotného

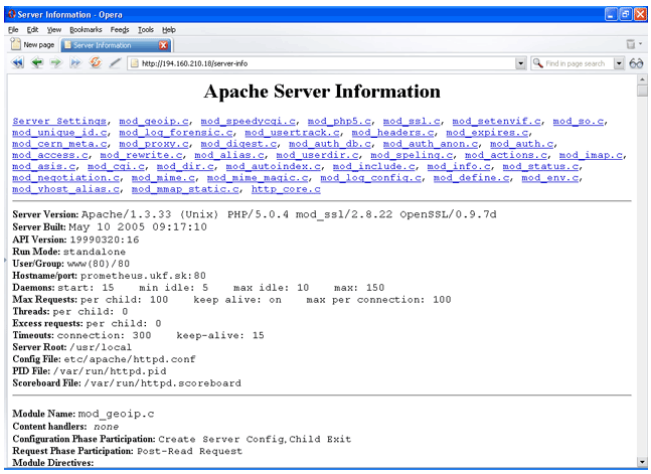
Služieb prístupných z Internetu by nemalo byť veľa a mali by sme povoliť len tie, ktoré naozaj potrebujeme. Keďže sa venujeme webovým systémom, tak jednou z povolených služieb musí byť práve webový server. Ďalšími povolenými službami sú shell prístup a emailová komunikácia. Všetky ostatné služby sú z Internetu zakázané korporátnym firewallom.

Povedali sme si, že najviac útokov prichádza z vnútornej siete. Vnútorná sieť nie je chránená korporátnym firewallom. Náš server je preto vystavený priamym útokom. Riešením proti útokom zvnútra je pokladať vnútornú sieť za nedôveryhodnú a správať sa k nej ako k Internetu. To znamená, povoliť prístup len k vybraným službám. Túto funkcionalitu nám zabezpečí firewall na strane servera.

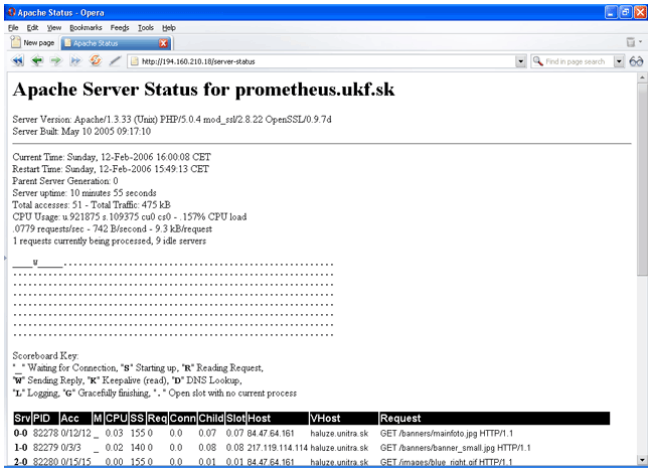
## 3 Útoky na webový server apache

### 3.1 Získanie informácií

Prvým krokom je získanie informácii o webovom serveri. Webový server Apache umožňuje volaním stránok `/server-info` a `/server-status` získať podrobné informácie o konfigurácií servera. Útočník vie potom oveľa ľahšie identifikovať slabé miesta a zamerať sa práve na ne.




*Obr. 1 Zobrazenie informácií o webovom serveri Apache*



*Obr.2 Zobrazenie stavu webového servera Apache*

Spôsob ochrany je jednoduchý. Upravíme konfiguračný súbor Apache servera, tak aby zobrazenie týchto informácií povolil len z lokálneho sieťového rozhrania.

```
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
</Location>

<Location /server-info>
    SetHandler server-info
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
</Location>
```

### 3.2 Pokus o odoprenie služieb (DoS útok)

Odoprenie služieb, čiže denial-of-service je útok na počítačový systém alebo počítačovú sieť, ktorý má za následok znemožnenie prístupu k bežným službám. Táto nedostupnosť je vytvorená znížením šírky prenosového pásma siete obete, čo má za následok stratu sieťovej konektivity (schopnosť nadviazať sieťové spojenie). Druhou metódou je preťaženie systému tak, že nebude schopný včas odpovedať na požiadavky používateľov. Obrana systému Apache spočíva v doinštalovaní a konfigurácii modulu *dosevasive*. Pokiaľ sa v krátkom časovom okamihu zopakujú požiadavky buď na tú istú stránku, alebo z tej istej IP adresy, odmietnu sa na istý čas všetky požiadavky z tohto počítača.

```
<IfModule mod_dosevasive20.c>
   DOSHashTableSize    3097
   DOSPageCount        2
   DOSSiteCount        50
   DOSPageInterval     1
   DOSSiteInterval     1
   DOSBlockingPeriod   10
   DOSEmailNotify      root@prometheus.ukf.sk
   DOSLogDir           "/var/log/dos"
   DOSWhiteList        127.0.0.1
```

```
</IfModule>
```

## 4 Útoky na databázový server

Náš informačný systém beží na databázovom systéme MySQL. Korporátny firewall neumožňuje pripojenie k nemu z Internetu. Z lokálnej siete je pripojenie možné. Podobne ako pri pokusoch o získanie shell prístupu, aj útok na databázový systém spočíva v uhádnutí mena a hesla používateľa. Spôsob ochrany je v zamedzení prístupu k databázam z iného počítača ako z lokálneho. Toto treba mať na pamäti pri vytváraní používateľských účtov v MySQL.

## 5 Útoky na aplikačnú vrstvu

Útok na túto vrstvu je najjednoduchší. Útočník nestojí proti týmu správcov systémov. Nemusí sa prebiť cez sériu firewallov, ktoré mu stoja v ceste. Stačí mu odhaliť chybu, ktorá vznikla nepozornosťou alebo neskúsenosťou programátora, alebo slabinou použitého programovacieho jazyka.

### 5.1 SQL Injection

SQL injection je možnosť vložiť do aplikácie SQL dotazy, ktoré neočakáva, a ktorých výsledkom je získanie citlivých údajov alebo možnosť autorizovať sa voči aplikácii bez znalosti prihlasovacích údajov.

Bránou pre SQL injection je ľubovoľný html formulár, ktorý odosiela údaje databázovému serveru. S veľmi veľkou pravdepodobnosťou je takouto vstupnou bránou prihlasovací dialóg.

Normálny SQL dotaz vyzerá nasledovne

```
SELECT * FROM users WHERE name = '" + $userName + "';
```

SQL umožňuje spájanie dotazov. Po bodkočiarke v dotaze môže nasledovať dotaz ďalší. Do poľa pre zadanie mena sme umiestnili dotaz, ktorý je zobrazený tučne. Tento dotaz spôsobí zmazanie tabuľky používateľov.

```
SELECT * FROM users WHERE name = 'a'; DROP TABLE users; SELECT * FROM
data WHERE name LIKE '%';
```

To, čo sa do SQL dotazu vloží, je len na tvorivosti útočníka. Ochranou proti takémuto počínaniu útočníka je kontrola vstupov a ošetrenie premenných predtým, ako ich odošleme v rámci dotazu SQL serveru. Nebezpečnými znakmi v dotazoch sú `\x00, \n, \r, \, ', "` a `\x1a`. Ochranou proti SQL injection je v jazyku php použitie funkcie `mysql_real_escape_string()`

Po použití tejto funkcie vyzerá náš dotaz nasledovne

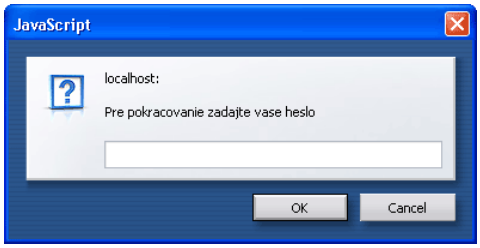
```
SELECT * FROM users WHERE name = \"".
  mysql_real_escape_string($user_name)."\""
```

### 5.2 XSS (Cross site scripting)

Cross site scripting je forma útoku na skripty spúšťané na strane webového klienta. Najčastejšou formou je podstrčenie JavaScriptu. Najviac používané sú skripty, ktoré spôsobia presmerovanie na inú webovú stránku alebo vypísanie nejakého chybového hlásenia. Veľmi častým použitím XSS je sociálne inžinierstvo. Táto metóda má za cieľ prinútiť používateľov

prezradiť svoje identifikačné údaje bez toho aby o tom vôbec vedeli. Príkladom takéhoto XSS je napríklad takého volanie:

```
<script>var heslo= prompt('Pre pokracovanie zadajte vase heslo', '');
 location.href="https://10.1.1.1/pass.cgi?passwd=heslo";</script>
```




*Obr. 3 Ukážka sociálneho inžinierstva*

Takýto skript vyprodukuje žiadosť zobrazenú na obrázku. Veľké percento používateľov takémuto hláseniu dôveruje a s radosťou napíše svoje heslo, ktoré sa ako vidíme odošle na server 10.1.1.1, ktorý je pod kontrolou útočníka. Keďže javascript sa spúšťa na strane webového klienta, na jeho správnu funkčnosť je nutná nevedomosť používateľa.

XSS je možné použiť pri aplikáciách typu *návštevná kniha*, kde má hocikto možnosť vložiť do stránky uvedený kód. Pokiaľ sa takáto aplikácia na stránke nenachádza, tak XSS nie je pre nás hrozbou.

Ak takúto aplikáciu potrebujeme, tak môžeme použiť php funkciu htmlspecialchars(), ktorá skonvertuje špeciálne znaky na ich ekvivalentné HTML entity. Za špeciálne znaky pritom považujeme &, <, >, apostrof a úvodzovky. Nahradením znakov <, > znefunkčníme vložené javascripty.

## 6 Záver

Pokúsili sme sa poukázať na najzákladnejšie bezpečnostné chyby v návrhu webového informačného systému. Je dôležité si uvedomiť, že základom bezpečného systému je dobre chránený server a jeho služby a správny návrh aplikácie. Je len na programátorovi, ako bude zaobchádzať so vstupnými údajmi od používateľa. Od aplikácie sa očakáva dôsledná kontrola týchto vstupných údajov a ich zmena pred tým ako sa odošlú na spracovanie databázovým systémom.

## 7 Literatúra

1. GARDIAN, M. – VYSKOČ, J. 2004. „Malá“ informačná bezpečnosť. In *Data Security Management*, roč. 8, 2004, č. 3, str. 14-17.
2. MCCLURE, S. – SCAMBRAY, J. – KURTZ, G. 2003. *Hacking bez tajemství*. Brno : Computer Press, 2003. 612 s. ISBN 80-722-6948-8.
3. http://en.wikipedia.org/wiki/Cross_site_scripting
4. http://en.wikipedia.org/wiki/SQL_Injection

## Adresa

Mgr. Martin Drlík, Mgr. Peter Švec,
Katedra informatiky, FPV UKF, Tr. A. Hlinku 1, 949 74 Nitra
mdrlik@ukf.sk, psvec@ukf.sk

Recenzent: Ing. Eva Oláhová, CIT FEM SPU v Nitre