

Damian Daszkiewicz

**Jak walczyć ze SPAMem na forach
internetowych, w księgach gości itp. ?**

www.DamianDaszkiewicz.pl

Większości z nas SPAM kojarzy się tylko z niechcianymi wiadomościami e-mail zawierającymi różne kuszące propozycje. Ostatnio zauważam zwiększoną ilość postów dopisanych na forach, różne wpisy w księgach gości, czy też dopisane komentarze pod artykułami na blogach. W tym artykule przedstawię kilka metod, które zmniejszą ilość SPAMu.

Jak nie ułatwiać życia spammerowi

Na stronach internetowych są miliony adresów e-mail. Spammerzy w tym celu piszą specjalne programy zwane harvesterami (ang. żniwiarki), których celem jest wyszukiwanie na stronach WWW adresów e-mail. Takie programy działają na bardzo prostej zasadzie – wyszukują w źródle strony znaku @ a później pobierają ciąg znaków znajdujący się za małpą, dopóki nie natrafiają na jakiś separator (np. spację). Podobnie wygląda sprawa z pobieraniem znaków „przed małpą”. Następnie jeśli ciągi znaków przed małpą i za małpą spełniają określone warunki (np. odpowiednia długość, brak zakazanych znaków) to pobrane ciągi znaków są dodawane do spammerskiej bazy adresów e-mail.

Dlatego nie powinno się w źródle strony podawać adresów e-mail pisząc go bezpośrednio np. `moj@email.pl`. Powinno się kodować adresy e-mail poprzez używanie wstawek JavaScriptowych. Na stronie <http://www.daszkiewicz.net/kodowanieemaili.php> znajduje się prosty programik, który generuje wstawkę JavaScript, której celem jest wyświetlenie adresu e-mail na stronie WWW.

Jeśli jednak tworzysz np. serwis ogłoszeniowy w którym ogłoszeniodawca podaje swój adres e-mail, który jest zapisywany do bazy danych. To trzeba pomyśleć o jakiejś prostej funkcji, która koduje pobrany z bazy adres e-mail. Taką funkcję prezentuje listing nr 1.

Listing 1, prosta funkcja kodująca adresy e-mail

```
function koduj($mail)
{
$tmp=explode("@",$mail);

$m='<script language="JavaScript">';
$m.='var p1="mai";';
$m.='var m2="'.$tmp[1].'";';
$m.='var p2="lto:";';
$m.='var m1="'.$tmp[0].'";';
$m.='var malpa="@";';
$m.='var adr=m1+malpa+m2;';
$m.='document.write(\'<a hr\'+\`ef="\`';
$m.='+p1+p2+adr+\`>\`+adr+\`</a>\`);';
$m.='</script>';

return $m;
}
```

Blokada .htaccess

Jeśli jakiś użytkownik usilnie spamuje zakładając nowe konta na forum to można zablokować mu dostęp do strony poprzez zablokowanie konkretnego adresu IP. W tym celu należy w pliku *.htaccess* dopisać następującą linijkę: deny from abc.def.ghi.jkl gdzie *abc.def.ghi.jkl* to adres IP danej osoby. Rozwiązanie to ma pewną wadę: sporo osób nie posiada stałego adresu IP (np. użytkownicy neostrady).

Zakazane słowa

Dość sporym problemem są spammerskie posty na forach dyskusyjnych lub w komentarzach na blogach. Analizując takie posty można dojść do wniosku, że są w nich dość często używane pewne ciągi znaków, których nie używają normalne osoby. Można więc napisać prosty skrypt, który przed opublikowaniem posta na forum (lub komentarza na blogu) sprawdza czy w tekście są jakieś zakazane słowa. Gdy są, to dany wpis zostanie zignorowany.

Jednak z czarną listą trzeba uważać. Przykładowo możesz dodać do czarnej listy słowo **viagra**. W ten sposób wyeliminujesz pewien procent SPAMu, ale z drugiej strony możesz utrudnić życie normalnym osobom. Załóżmy, że prowadzisz forum medyczne i jakiś lekarz chce porozmawiać na temat efektów ubocznych stosowania „niebieskiej tabletki”. Jego post nie zostanie zapisany, gdyż słowo **viagra** znajduje się na czarnej liście. W ten sposób Twoja walka ze spammerami utrudniła życie niewinnym ludziom. Osobiście do pliku z zakazanymi słowami wpisuję bardzo specyficzne słowa, których na pewno zwykły człowiek nie użyje w swoich wypowiedziach (np. nazwy spammerskich domen, które były reklamowane na forum lub numery GG/ICQ jakiś natrętów).

Listing nr 2 zawiera fragment kodu sprawdzający, czy w treści napisanej wypowiedzi znajduje się jakiś niedozwolony ciąg znaków. Jeśli chcesz ten kod wykorzystać na swoim forum opartym o skrypt PHPBB2 to dodaj go na samym początku pliku *posting.php*. Skrypt wymaga utworzenia pliku *cenzura.txt* w którym będziesz dodawał niedozwolone ciągi znaków (np. adresy spammerskich domen). Plik *cenzura.txt* ma prostą budowę: w jednej linijce powinien się znajdować jeden ciąg znaków.

Listing 2, fragment kodu sprawdzający czy post zawiera niedozwolone słowa

```
$docenzury=strtoupper($_REQUEST['message']);

$PlikCenzura=file("cenzura.txt");
for ($licznik=0; $licznik<count($PlikCenzura); $licznik++)
{
$czurowany=chop(strtoupper($PlikCenzura[$licznik]));
$posx = strpos($docenzury, $czurowany);

if ($posx !== false)
{
echo("Post zawiera niedozwolone znaki");
exit;
}
}
```

```
unset($PlikCenzura);  
unset($docenzury);  
unset($cenzurowany);
```

Zbyt wiele adresów URL

Niektórzy spammerzy nie są zbyt inteligentni. Wala w komentarzu np. 50 adresów do różnych stron WWW. Można to wykorzystać podczas pisania prostych filtrów sprawdzających komentarz przed dodaniem, czy nie jest SPAMem. Załóżmy, że w komentarzu mogą znajdować się maksymalnie 2 adresy URL. Prosty przykład funkcji prezentuje listing nr 3

Listing 3, funkcja zliczająca ilość linków w danym tekście

```
function czyspam($str)  
{  
$ile=substr_count($str,"http://");  
$ile+=substr_count($str,"https://");  
  
if ($ile>2) return true;  
else return false;  
}
```

Jak ma na imię Małysz?

Prowadząc jakiś serwis skupiający pewną społeczność warto postawić forum. Większość osób wybiera najpopularniejsze forum czyli PHPBB2. To rozwiązanie ma wiele zalet: po pierwsze forum jest darmowe. Po drugie „interfejs” forum jest znany większości internautów. Po trzecie: w sieci istnieje masa dodatków rozszerzających możliwości forum. Jeśli chodzi o jakość kodu, to pomnę to milczeniem.

Problem jest taki, że wszystkie fora mają tę samą budowę: takie same nazwy plików, tak samo nazywają się pola formularza rejestracyjnego. Spammerzy w tym celu korzystają z gotowych programów, które zakładają nowe konta na podanych forach i tworzą nowe wątki (w których zachęcają do kupna niebieskiej tabletki). Aby znaleźć fora, na których chce się spamować nie trzeba się namęczyć, wystarczy wpisać w

dowolnej wyszukiwarce *Powered by phpBB*.

Najprostsze rozwiązanie eliminujące ten problem polega na dodaniu niestandardowego pola w formularzu. Niech to będzie jakieś pytanie, na które odpowiedź zna każdy (np. „jak ma na imię Małysz”, „napisz słownie cyfrę 1”, „Jaka jest stolica Polski”). Tutaj mała uwaga: pytanie musi być naprawdę banalne.

Widziałem pewne forum, na którym aby się zarejestrować należało policzyć całą oznaczoną ;-). Druga ważna rzecz: niech odpowiedzią będzie jedno słowo najlepiej nie zawierające polskich znaków (niektórzy użytkownicy forum mogą być za granicą, gdzie komputery nie mają zainstalowanego polskiego układu klawiatury, a dla walki ze spammerami nie powinniśmy im utrudniać życia).

Czasami widziałem modyfikacje polegające na wpisaniu wyniku obliczeń (np. $1+1=?$). Nie jest to dobre rozwiązanie, gdyż czasami się zdarza, że na forum nie rejestruje się „automat”, tylko jakiś „hindus”. O ile „hindus” może się domyśleć, że w polu obok należy wpisać wynik dodawania, to po przeczytaniu jakiegoś zdania w obcym dla niego języku raczej nie będzie wiedział co trzeba wpisać.

Wprowadzenie owego zabezpieczenia polega na modyfikacji dwóch plików. Po pierwsze: należy w pliku *usercp_register.php* znaleźć ciąg znaków: [// Get current date](#) i za nim dopisać kod przedstawiony na listingu 4. Drugi plik do edycji to pliku szablonu (zazwyczaj jest to plik *templates/subSilver/profile_add_body.tpl*). Należy znaleźć w nim ciąg znaków przedstawiony na listingu nr 5 i za tym ciągiem znaków dodać kod przedstawiony na listingu nr 6.

Listing 4, kod weryfikujący czy podczas rejestracji na forum udzielono prawidłową odpowiedź na zadane pytanie

```
if (trim(strtolower($_POST['pmaalysz']))!="adam"){  
message_die(GENERAL_ERROR, 'Nie wpisałeś imienia Małysza (lub wpisałeś  
błędne imię). Prawdopodobnie jesteś spammerskim robotem', '', __LINE__,  
__FILE__, '');  
}
```

Listing 5, w pliku szablonu należy znaleźć następujący ciąg znaków

```
<tr>
<td class="row1"><span class="gen">{L_EMAIL_ADDRESS}: *</span></td>
<td class="row2"><input type="text" class="post" style="width:200px"
name="email" size="25" maxlength="255" value="{EMAIL}" /></td>
</tr>
```

Listing 6, i dodać poniższy kod

```
<tr>
<td class="row1"><span class="gen">Wpisz imię Małysza *</span></td>
<td class="row2"><input type="text" class="post" style="width:200px"
name="pmaalysz" size="10" maxlength="255" value="" /></td>
</tr>
```

Krótkie wyjaśnienie: plik *profile_add_body.tpl* zawiera szablon formularza rejestracyjnego (w tym pliku dodaliśmy dodatkowe pole w którym należy wpisać odpowiedź na zadane pytanie). Natomiast plik *usercp_register.php* odpowiada między innymi za rejestrowanie się użytkowników na forum (dodaliśmy w tym pliku kod, który sprawdza, czy udzielona odpowiedź na zadane pytanie jest poprawna).

Zliczanie wpisów

Założmy, że posiadasz prosty system komentarzy. Pod każdym artykułem można dodać swój komentarz. Dość sporym problemem są osoby, które pod każdym artykułem dodają ten sam komentarz (np. zachęcający do odwiedzenia pewnej strony WWW). Można to zjawisko wyeliminować w bardzo prosty sposób: wystarczy zliczać ilość dodanych komentarzy z danego adresu IP w pewnej jednostce czasu. Gdy określony limit zostanie przekroczony, to komentarz nie zostanie dodany. Przykładowe limity to: maksymalnie 3 wpisy w ciągu 5 minut i maksymalnie 10 wpisów w ciągu godziny. Aby wdrożyć to rozwiązanie należy w bazie danych utworzyć tabelę o nazwie **logiip** zawierającą dwie kolumny: **czas** i **ip**. Listing nr 7 przedstawia kod sprawdzający, czy dany użytkownik nie przekroczył któregoś z tych dwóch limitów (funkcja zwraca wartość false, jeśli użytkownik przekroczył limit).

Oprócz sprawdzania limitów funkcja dodaje do bazy danych informację, że dany użytkownik właśnie dodał komentarz (więc nie należy wywoływać tej funkcji kilka razy).

Listing nr 7, funkcja sprawdzająca, czy dana osoba nie dodaje zbyt wiele komentarzy

```
function dodackomentarz()
{
//pobierz adres IP
$aIP=$_SERVER["REMOTE_ADDR"];

//czy dodano >10 komentarzy w ciągu ostatniej godziny?
$czas=time()-3600;
$result = mysql_query("SELECT id FROM logiip WHERE ip='$aIP' and
czas>$czas");
$file1=mysql_num_rows($result);
mysql_free_result($result);

//czy dodano >3 komentarze w ciągu 5 minut?
$czas=time()-300;
$result = mysql_query("SELECT id FROM logiip WHERE ip='$aIP' and
czas>$czas");
$file2=mysql_num_rows($result);
mysql_free_result($result);

$wynik=true;
if ($file1>10 || $file2>3) $wynik=false;

//dodaj do bazy informację o dodaniu nowego komentarza z tego adresu IP
if ($wynik==true) $x=mysql_query("INSERT INTO logiip VALUES
('$ip', '$czas')");

return $wynik;
}
```

Ponieważ rozmiar tabeli *logiip* będzie rósł, więc dobrym pomysłem będzie okresowe usuwanie niepotrzebnych (starych) informacji. Funkcja nie zlicza informacji o komentarzach dodanych wcześniej niż godzinę temu, więc można takie

wpisy usuwać. Po usunięciu wpisów warto jest zoptymalizować tabelę, co przyspieszy działanie skryptu (optymalizację tabeli można porównać do defragmentacji dysku). Najlepiej jest utworzyć krótki skrypt, który przedstawiam na listingu nr 8 i dodać go do crona, aby był uruchamiany co godzinę.

Listing 8, skrypt usuwający niepotrzebne wpisy

```
$a=time()-3600;  
$x=mysql_query("DELETE FROM logiip WHERE czas<$a");  
$x=mysql_query("OPTIMIZE TABLE logiip");
```

Wtyczki oparte o filtr Bayes'a

Jeśli powyżej opisane proste techniki nie przynoszą odpowiednich rezultatów, to możesz skorzystać z odpowiedniej wtyczki. Wtyczki te działają na dość prostej zasadzie: wysyłamy na zewnętrzny serwer treść komentarz i otrzymujemy odpowiedź, czy dany komentarz jest czysty, czy nosi znamiona SPAMu. Wtyczki mają dodatkową zaletę: ponieważ wykorzystują mechanizmy algorytmu Bayes'a więc się uczą. Zauważ, że takie wtyczki są „karmione” olbrzymią ilością danych (gdyż korzystają z nich tysiące osób) więc są bardzo skuteczne. Polecam zapoznanie się z projektem na stronie www.sblam.com (strona jest w języku polskim i są na niej gotowe pliki dla popularnych for i blogów).

Jeśli używasz bloga opartego na WordPressie, to warto jest się zapoznać z darmową wtyczką Akismet: www.akismet.com

Przydatne linki:

[Wtyczka sblam](http://www.sblam.com)

[Wtyczka Akismet](http://www.akismet.com)

[Program do kodowania adresów e-mail](#)

[Ebook Profilaktyka antyspamowa](#)

Artykuł: [CAPTCHA – jak odróżnić złe od gorszych](#)

[Definicja terminu Harvester](#)

[Skarbnica wiedzy o SPAMie](#)

[Moduł dla CMS'a drupal utrudniający automatyczne zakładanie kont:](#)