

Podstawowa konfiguracja serwera www w 1 dzień

Spis treści

1. Wstęp	3
2. Zapoznanie się z panelem	4
2.1. Panel HyperVM	4
2.2. Panel płatności WHMCS	7
3. Podstawowe zabezpieczenia systemu	8
3.1. SSH	8
3.2. Uniknąć ForkBomb	9
3.3. Prawa dostępu	11
3.4. Dostęp do konta root	13
3.4.1. Nowy użytkownik	13
3.4.2. Użycie sudoers	15
4. Instalacja i konfiguracja LAMP	16
4.1. Konfiguracja Apache2	17
4.1.1. VirtualHost	19
4.2. Konfiguracja PHP5	22
4.3. Konfiguracja MySQL5	24
4.4. phpMyAdmin	26
5. Bind – serwer nazw	27
5.1. Instalacja i zabezpieczenia	27
5.2. Własne NameSeryery	30
5.3. Strefa dla nowej domeny	33
6. Dodatkowe składniki	35
6.1. Postfix	35
6.1.1. Czy działa Postfix?	39
6.1.2. Czy działa mail() ?	40
6.2. ProFTPd	42
6.3. Eaccelerator	43

1 Wstęp

(wersja elektroniczna: <http://www.bluman.pl/internet/id903-hostingi-transfer-i-cena.html>)

Na wstępie chciałem przeprosić wszystkich zawodowych administratorów serwerów, którzy natrafią na mój poradnik. Na pewno odbiorę im przez to część zarobków jakie mieli do tej pory. Ale gdyby popatrzeć na to z drugiej strony to okaże się, że rozpowszechniając taki poradnik w sieci spowoduję, że wiedza na temat serwerów wzrośnie, a co za tym idzie zapotrzebowanie na nie także. W szerszej perspektywie czasu, kiedy ludzie będą mieć już własne serwery dodatkowe ręce do pracy przy ich administracji na pewno będą im potrzebne – mam nadzieję, że wtedy skorzystają z pomocy wykwalifikowanych administratorów. Życzę Wam tego :)

Zapowiadając znajomym napisanie tego tutoriala wiele razy słyszałem słowa „Ale po co?” i w gruncie rzeczy odpowiedź na to trzy wyrazowe pytanie nie jest do końca trywialna. Myślę, że na takie pytanie każdy sobie powinien odpowiedzieć – czy faktycznie jest mu potrzebny VPS, aniżeli ja mam za kogoś odpowiadać. To przecież ty najlepiej znasz swoje potrzeby na miejsce i sposób trzymania stron www. Każdy, kto poważnie myśli i wie co to jest serwer dedykowany/vps zdaje sobie sprawę po co i jak go wykorzystać. Wie, że hosting współdzielony ma swoje granice, i nie są to granice czysto pojemnościowe HDD, czy transferu GB/miesiąc. W pewnym momencie skrypty uruchomione na serwerze przy dużej ilości odwiedzi potrzebują więcej mocy obliczeniowej, które niestety nie jest nam dana w typowym koncie hostingowym.

Dlatego właśnie powstał ten poradnik – aby zminimalizować ból i problemy powstałe przy przejściu z hostingu typu shared na VPS.

W poradniku wykorzystuję system operacyjny Debian®5.0. Na różnych systemach i w innych wersjach Debiana komendy mogą się różnić.

Poradnik ten **chciałem zadedykować mojej ukochanej i jedynej kobiecie na Świecie o pięknym imieniu Anna**. Dzięki niej mogę dalej rozwijać swoje umiejętności i dzięki niej nie poszedłem teraz spać z nią, tylko łaskawą ręką pozwoliła mi spędzić samotnie wieczór przed komputerem ;) Dziękuję ci Aniu za wszystko!

Jeśli ktoś czułby potrzebę skontaktowania się ze mną, to zapraszam: szbluma@gmail.com. Ale nie będę namawiał – mam przecież sporo własnych rzeczy do robienia ;)

2 Zapoznanie się z panelem

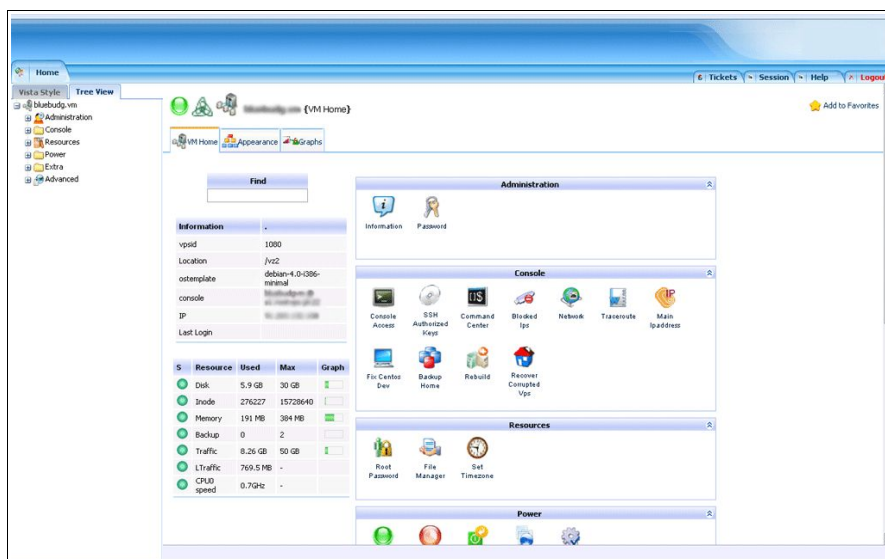
(wersja elektroniczna: <http://www.bluelman.pl/serwer/id928-konfiguracja-serwera-cz1.html>)

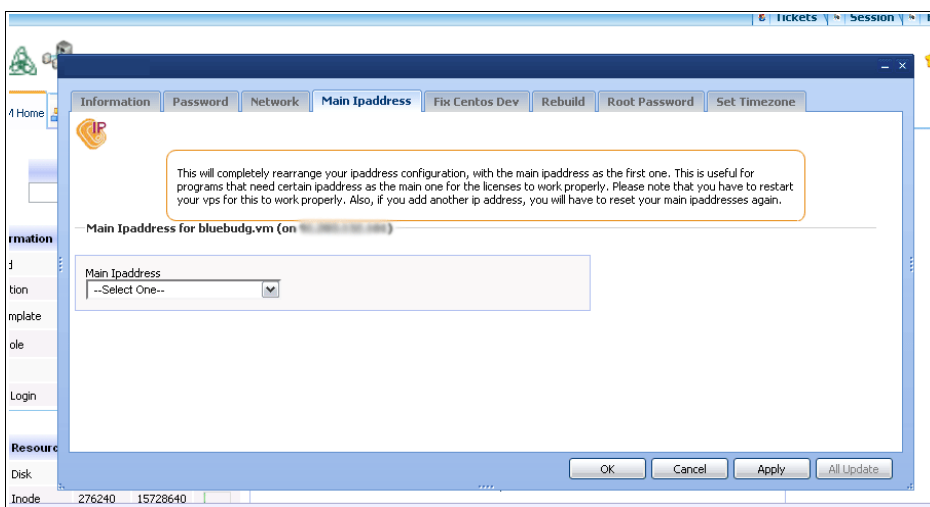
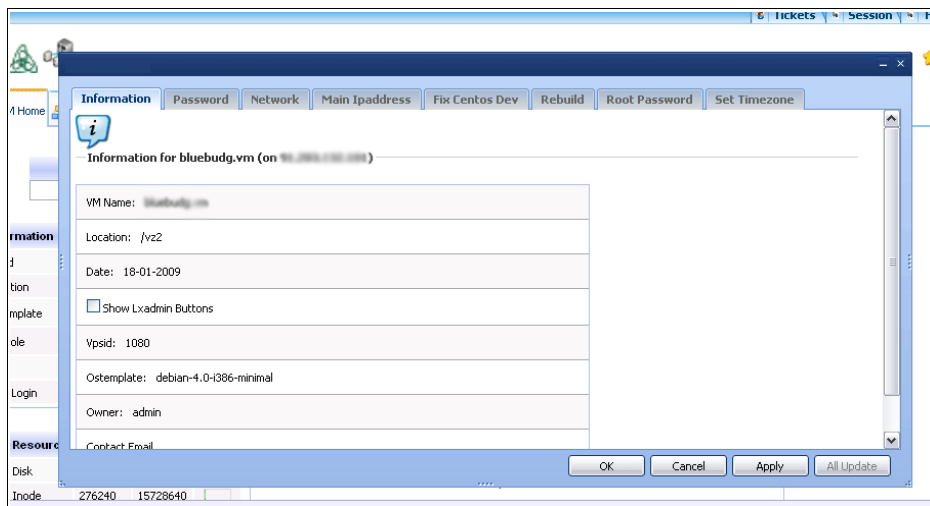
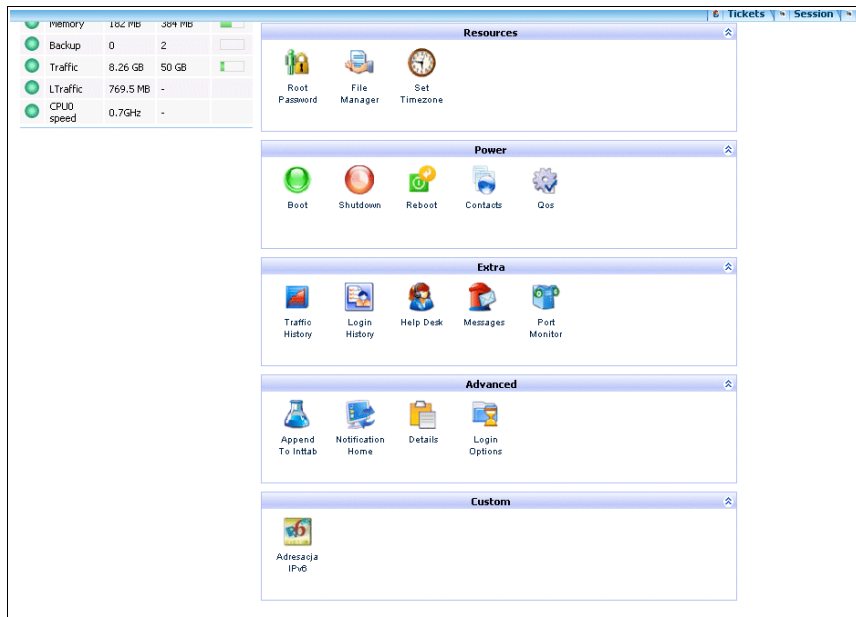
W pierwszej części poradnika skupić chciałbym się na panelu administracyjnym serwera (ale nie usług na nim uruchomionych!).

2.1 Panel HyperVM

Dzięki temu panelowi możemy zarządzać i monitorować podstawowe usługi w serwerze. Dzięki niemu, bez logowania na serwer możemy zrestartować to (trwa to dosłownie kilka sekund), wyłączyć go, czy podejrzeć wykres zużycia RAM, CPU i łącza internetowego. Widzimy także ile transferu zużyliśmy w poprzednim miesiącu i ile zostało już go wykorzystane w tym. I prawdę powiedziawszy przy codziennym użytkowaniu jego funkcje na tym się kończą. Oferuje co prawda jeszcze inne (typu konsola w JAVA, przeglądanie dysku twardego, ustawienie czasu serwera), ale osobiście z tego nie korzystam, ponieważ wolę to robić bezpośrednio na serwerze przez SSH, niż korzystać z przeglądarkowych odpowiedników.

Z poziomu tego samego panelu administracyjnego możemy także przeinstalować system. We wszystkich firmach dostępne są takie systemy do wyboru jak: Debian, Ubuntu, CentOS, Gentoo, Fedora, OpenSuSE, Slackware, SuSE w różnych wersjach, ale zawsze w 32-bitowym wydaniu.

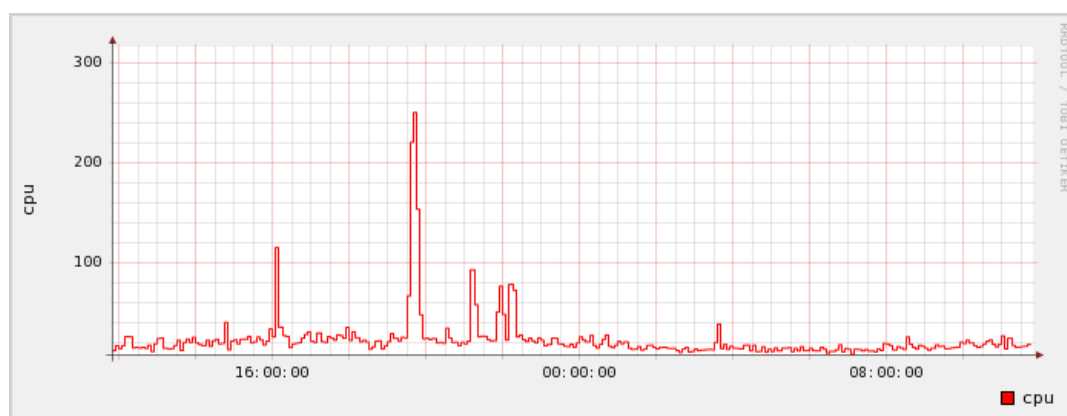
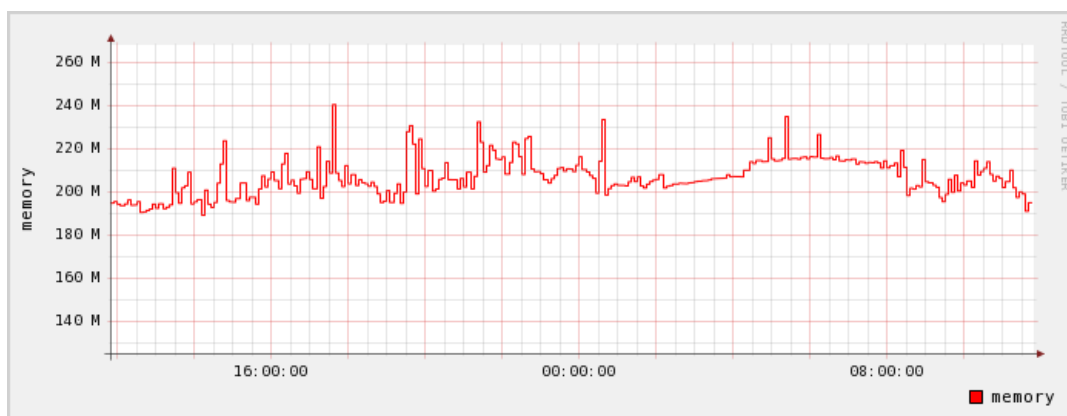
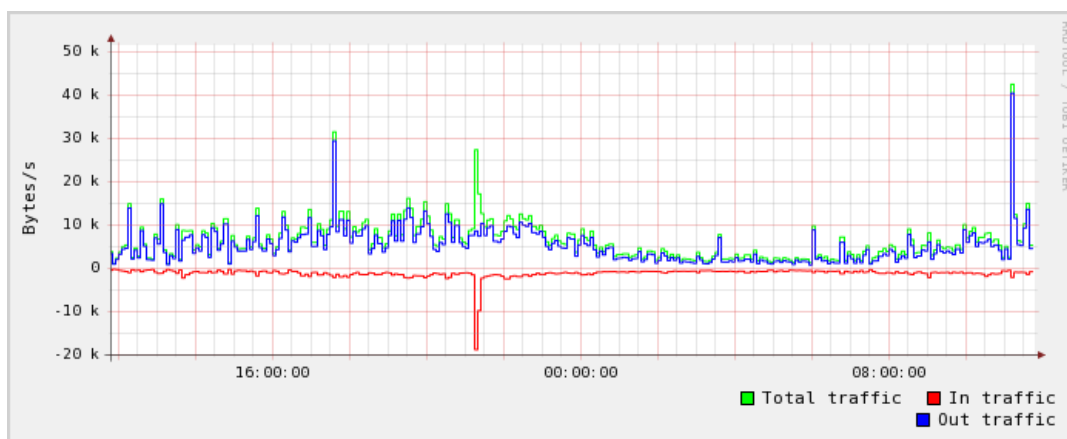




Autor: **Szymon Bluma**, Strona: www.BlueMan.pl, maj 2009
 Poradnik rozpowszechniany na licencji [Creative Commons BY-NC-ND](http://creativecommons.org/licenses/by-nc-nd/3.0/).

Zrzutów ekranowych z tego panelu mógłbym wiele zamieszczać, jednak mija się to z celem, ponieważ w dokumencie PDF, na kartce papieru A4 ilość detali jakie można dopatrzeć jest znikoma. Zapraszam więc do obejrzenia wszystkich screenschootów u mnie na stronie (link pod nagłówkiem działu).

Najciekawszą rzeczą w prezentowanym panelu HyperVM jest możliwość obserwacji zużycia zasobów CPU i RAM w różnych przedziałach czasowych. Dzięki temu chociaż orientacyjnie wiemy, czy serwer jest poprawnie skonfigurowany i czy działa stabilnie.



2.2 Panel płatności WHMCS

Innym panelem jest panel płatności (billingowy). Nikt z niego korzystać nie chce, bo wiąże się to z wydatkami na serwer, no ale trzeba. Dzięki niemu możemy na bieżąco kontrolować zaległe i przyszłe opłaty za usługi wykupione w firmie, a także dokupywać dodatkowe usługi. Panel ten jest na tyle intuicyjny, że nie będę opisywać dokładnie jego możliwości. Myślę, że każdy poradzi sobie samodzielnie z ujarzaniem go.



Tego panelu proszę nie wiązać bezpośrednio z serwerami dedykowanymi, czy VPS. Jest to po prostu dodatkowe oprogramowanie z którego korzystają również firmy, które wynajmują hosting współdzielony.

Online Client Portal

Jesteś tutaj: [Pomoc](#) > [Strefa klienta](#) Witamy ponownie! Szymon! [Wyloguj](#)

Statystyki konta
Liczba Produktów/Usług: 1 (1)
Liczba Domen: 0 (0)
Liczba Support Ticketów: 0
Liczba referowanych zamówień: 1
Saldo kredytów konta: 0.00
Saldo zaległych rachunków: 0.00

0 Otwarte zgłoszenia o problemach ([Wyślij zgłoszenie](#))

Data	Temat	Status	Priorytet
Nie odnaleziono wpisów			

0 Zaległe rachunki

Rachunek #	Data wystawienia	Płatny do	W sumie	Status
Nie odnaleziono wpisów				

Powered by [WHMCompleteSolution](#)

Język: **Polski**

Online Client Portal

Jesteś tutaj: [Pomoc](#) > [Strefa klienta](#) > [Moje rachunki](#) Witamy ponownie! Szymon! [Wyloguj](#)

Moje rachunki

3 Znaleziono wpisy, Strona 1 z 1 « Poprzednia strona Kolejna strona »

Rachunek #	Data wystawienia	Płatny do	W sumie	Status	
847	04/03/2009	18/03/2009		Opłacony	Zobacz rachunek
565	04/02/2009	18/02/2009		Opłacony	Zobacz rachunek
435	18/01/2009	18/01/2009		Opłacony	Zobacz rachunek

Pokaż: [10](#) [25](#) [50](#) [100](#) [Wszystkie](#) « Poprzednia strona Kolejna strona »

Powered by [WHMCompleteSolution](#)

Język: **Polski**

Autor: **Szymon Bluma**, Strona: www.BlueMan.pl, maj 2009
Poradnik rozpowszechniany na licencji [Creative Commons BY-NC-ND](http://creativecommons.org/licenses/by-nc-nd/).

3 Podstawowe zabezpieczenia systemu

(wersja elektroniczna: <http://www.blue-man.pl/serwer/id949-konfiguracja-serwera-cz2.html>)

Mimo że poradnik ten zaadresowany jest do osób, które uruchamiają serwer tylko do swoich prywatnych potrzeb i raczej nie będą współdzielić go ze znajomymi to jednak trzeba dokonać podstawowych zabezpieczeń w zainstalowanym systemie. Głównie chodzi tutaj o odparcie ataków z zewnątrz, aby potencjalny intruz nie narobił szkód w systemie.

Wiele osób instaluje także firewall (iptables), ale ja jakoś nie przekonałem się do niego i nie był mi nigdy potrzebny.

3.1 SSH

W ścisłym znaczeniu SSH to tylko następca protokołu Telnet, służącego do terminalowego łączenia się ze zdalnymi komputerami. SSH różni się od Telnetu tym, że transfer wszelkich danych jest zaszyfrowany oraz możliwe jest rozpoznawanie użytkownika na wiele różnych sposobów. W szerszym znaczeniu SSH to wspólna nazwa dla całej rodziny protokołów, nie tylko terminalowych, lecz także służących do przesyłania plików (SCP, SFTP), zdalnej kontroli zasobów, tunelowania i wielu innych zastosowań. Wspólną cechą wszystkich tych protokołów jest identyczna z SSH technika szyfrowania danych i rozpoznawania użytkownika. Obecnie protokoły z rodziny SSH praktycznie wyparły wszystkie inne "bezpieczne" protokoły, takie, jak np. rlogin czy RSH.

Głównym zagrożeniem są boty, które próbują złamać hasło root w konsoli, aby zainfekować kolejną maszynę do rozsyłania spamu i włamań na inne serwery. Aby tego uniknąć należy zmienić domyślny port SSH i wyłączyć możliwość logowania się głównego użytkownika (root) bezpośrednio na konsole.

W pliku

`/etc/ssh/sshd_config`

musimy zmienić kilka wpisów:

```
Port 5022
Protocol 2
LoginGraceTime 45
MaxAuthTries 5
PermitRootLogin no
PermitEmptyPasswords no
```

- *Port* – numer portu przez który będziemy się logować. Domyślna wartość to 22. Sugeruję ustawiać wysokie numery portów – powyżej 5000.
- *Protocol* – nr 2 protokołu jest znacznie bardziej bezpieczniejszy, niż nr 1. Teraz w Ubuntu i Debianie domyślnie jest już ustawiony nr 2, ale mimo wszystko warto dla pewności zwrócić na to uwagę.
- *LoginGraceTime* – czas w sekundach na zalogowanie. Domyślna wartość to 600. A chyba nie zajmie ci 10 min wpisanie loginu i hasła?
- *MaxAuthTries* – ilość błędnych prób logowania się na konto. Po tych próbach zostaniesz „wyrzucony” z serwera i trzeba łączyć się ponownie. Nie oznacza to, że twoje konto zostanie zablokowane! LoginGraceTime i MaxAuthTries utrudnia życie botom, które metodą brutalforce próbują dostać się do serwera.
- *PermitRootLogin* – zabraniamy logowania się na konto root bezpośrednio przez SSH
- *PermitEmptyPasswords* – Zabrania logowania się na konto bez podania hasła.

Dodatkowym zabezpieczeniem może być umożliwienie logowania się na serwer tylko z określonych adresów IP. W ten sposób jesteśmy niemal w 100% zabezpieczeni przed próbami ataków.

/etc/hosts.allow

```
sshd : 127.0.0.1 : allow
sshd : TwójAdresIP : allow
sshd : TwójInnyAdresIP : allow
sshd : ALL : deny
```

Niestety może to być miecz obusieczny – kiedy będziemy na wakacjach i chcąc „pogrzebać” coś w serwerze nie będziemy mieć takiej możliwości. Chyba, że na czas urlopy przestawimy uprawnienia. No i w dzisiejszych czasach większość z nas nawet w domu ma zmienny adres IP, a więc ta opcja będzie mało przydatna... .

3.2 Uniknąć ForkBomb

Fork-bomba (ang. fork bomb) jest rodzajem ataku Denial of Service na systemy komputerowe.

Atak opiera się na założeniu, że w środowisku wieloprotocowym tylko pewna ilość procesów może być efektywnie wykonywana naraz. Atak polega na bardzo szybkim "rozmnożeniu" kopii programu (fork to nazwa funkcji systemowej służącej do tworzenia nowych procesów) w celu wypełnienia tablicy procesów systemu operacyjnego. W takiej sytuacji wywołanie nowego procesu (mającego na celu np. zabicie procesów bomby) jest wstrzymane do czasu zwolnienia choćby jednego wpisu, co jednak jest mało prawdopodobne,

ponieważ każdy proces bomby jest gotów w tym momencie się rozmnożyć.

Ponieważ każdy z procesów bomby wykonuje jakiś kod (nie usypia się) planista systemowy każdemu z nich przydziela czas procesora, co praktycznie zatrzymuje działanie systemu. **Jedną z technik obrony przed fork-bombami jest ustalenie górnego limitu procesów, jakie może utworzyć dany proces lub użytkownik (dotyczy to również jego dalszych potomków).** I właśnie ten typ obrony zostanie przedstawiony, ponieważ jest najprostszy do zaimplementowania.

```
:(){:|:&};:
```

UWAGA!!

To polecenie w niezabezpieczonym systemie zrobi spustoszenie. Pomoże tylko i wyłącznie restart maszyny! Wykonując je robicie to na własną odpowiedzialność !!

Aby uniknąć tego typu przykrych rzeczy, należy w systemie stworzyć nową grupę użytkowników, którzy będą mieć ograniczoną ilość uruchomionych procesów. Takich grup można stworzyć więcej – w zależności od tego jak bardzo zaufane osoby będziemy chcieli wpuszczać do systemu.

```
# addgroup TwojaNazwaGrupy
```

Dodajemy nową grupę do systemu, po czym w pliku */etc/security/limits.conf*

Dodajemy poniższe linijki

```
@TwojaNazwaGrupy      soft    nproc   100
@TwojaNazwaGrupy      hard    nproc   150
```

W ten sposób ograniczymy ilość maksymalnie uruchomionych procesów dla danej grupy do 150, a przy 100 użytkownik ten zostanie poinformowany o zbliżającej się granicy „wytrzymałości” jego grupy.

3.3 Prawa dostępu

Mimo wszystko – jeśli chcemy udostępnić serwer znajomym, to warto zabezpieczyć się, aby znajomy Kowalski nie patrzył w pliki Nowakowi.

Możemy tego dokonać na kilka sposobów: chmod plików, chroot'owane środowisko, jail.

Te dwa ostatnie sposoby są dość ciężkie do wytłumaczenia i są po prostu trudne, ale dzięki nim mamy większe bezpieczeństwo w systemie. Ja uważam, że jeśli serwer jest głównie jednej osoby, a dodatkowe konto FTP dla znajomego jest udostępniane naprawdę sporadycznie to wystarczy zastosowanie sposobu nr 1.

Najpierw powiem o tym, jak zabronić użytkownikowi korzystać z konsoli. Tworząc przecież nowego użytkownika, jego domyślne wartości umożliwiają mu przecież korzystanie z SSH i wykonywanie skryptów basha. Trzeba to zmienić.

```
/etc/adduser.conf  
/etc/default/useradd
```

W różnych systemach w różnym miejscu domyślne wartości tworzonego użytkownika się znajdują. Sami musicie drogą empiryczną dojść do tego z którego pliku wasz system korzysta.

W każdym bądź razie znajdziecie w tych plikach kilka podstawowych opcji jakie są „doklejane” do polecenia adduser. Możecie sobie poeksperymentować z różnymi ustawieniami – pozwalam ;)

Nas interesuje pierwszy odhashowany zapis. Domyślnie wygląda on:

```
SHELL=/bin/sh  
#lub  
SHELL=/bin/bash
```

A musimy go zmienić na:

```
SHELL=/bin/false
```

Dzięki czemu użytkownik nie będzie mógł logować się na konsole.

Drugim zabezpieczeniem są odpowiednie prawa zapisu/odczytu do plików i folderów.

Chodzi tutaj o to, aby użytkownik *pnowak*, który należy do grupy *pnowak* nie mógł podglądać prywatnych plików użytkownika *mkowalski* (grupa *mkowalski*).

Istnieje kilka sposobów zapisu praw do danego pliku. Najpopularniejszymi są: system numeryczny, oraz literowy. Numerycznie chmod przyjmuje odpowiednią wartość potęgi dwójki dla każdego typu akcji (zapisu, odczytu, uruchomienia).

Typ zapisu	Prawo odczytu	Prawo zapisu	Prawo uruchomienia
Potęga dwójki	2 ²	2 ¹	2 ⁰
Numer dziesiętny	4	2	1
Znak	r (ang. <i>read</i>)	w (ang. <i>write</i>)	x (ang. <i>execute</i>)

Ja od początku byłem wychowywany w nomenklaturze numerycznej i tylko taką stosuję. Inne są dla mnie mało zrozumiałe, no ale to na pewno kwestia przyzwyczajenia... co do tej pory nie było mi potrzebne, więc zostałem przy swoich nawykach :)

Przykłady zastosowań:

Prawa dostępu	Wartość liczbowa	Opis
-rw-----	600	Tylko właściciel ma prawo do odczytu i zapisu.
-rw-r--r--	644	Właściciel ma prawo do zapisu i odczytu, a reszta tylko prawo odczytu.
-rw-rw-rw-	666	Wszyscy mają prawo do odczytu i zapisu.
-rwx-----	700	Tylko właściciel ma prawo do odczytu, zapisu, uruchomienia.
-rwxr-xr-x	755	Właściciel ma wszystkie prawa do pliku, reszta tylko prawo do odczytu i uruchomienia.
-rwxrwxrwx	777	Wszyscy mają pełne prawa (nie zalecane).
-rwx--x--x	711	Wszystkie prawa ma właściciel, reszta tylko prawo uruchomienia.
drwx-----	700	Właściciel katalogu ma pełne prawa do niego (katalogi mają literkę 'd' na początku zamiast '-')
drwx--r--r	744	Właściciel ma pełne prawa do katalogu, reszta ma prawo do odczytu.

Najbezpieczniejsze prawa dostępu do plików to 641, a dla folderów 751. Aby zmienić domyślne ustawienia (które są odpowiednio 644 i 755), należy w konsoli wpisać:

```
# umask 0026
```

3.4 Dostęp do konta root

root to tradycyjna nazwa uniksowego konta, które ma pełną kontrolę nad systemem. Z założenia konto root nie powinno być używane do pracy, do której wystarczyłoby zwykłe konto z ograniczonymi uprawnieniami. Istotną sprawą jest zabezpieczenie tego konta silnym hasłem i zabezpieczenie przed nieautoryzowanym dostępem.

Dlatego zalecam wyłączyć w ogóle możliwość logowania się na konto roota. Nawet jeśli jakiś robak dostanie się do wnętrza systemu poprzez nasze normalne konto użytkownika i będzie próbował zalogować się na konto root'a to nie uda mu się to. Mimo że robactwo jest coraz bardziej inteligentne i na pewno istnieją już robaki, które będą próbować złamać hasła do wszystkich kont systemowych, to pozostaje być przy nadziei, że intruz skup się tylko na koncie root.

3.4.1. Nowy użytkownik

Oczywiście nie możemy zrezygnować z dostępu do super administratora systemu, czy męczyć się każdorazowym włączeniem dostępu do roota. W tym celu stworzymy nowego użytkownika, który będzie mieć wszystkie prawa w systemie.

```
# adduser blueroot
```

Pamiętajcie, aby podać długie i ciężkie hasło do złamania! Najlepiej z mieszanymi dużymi, małymi literami, cyframi oraz znakami specjalnymi.

```
/etc/passwd
```

W tym pliku zmienimy domyślne prawa użytkownika aby był rootem, a także domyślny katalog.

```
blueroot:x:1005:1005:,,,:/home/blueroot:/bin/false
```

- 1005 – jest to UID i GID użytkownika. Zmieńmy te wartości na 0 (czyt. zero)
- /home/blueroot – katalog domowy. Ja podaję tutaj zawsze swój domyślny katalog użytkownika, ponieważ tam trzymam wszystkie swoje pliki.
- /bin/false – jeśli zastosowaliśmy wcześniej praktyki zabezpieczenia systemu, to oznacza to, że użytkownik blueroot nie może korzystać z terminala. Trzeba to zmienić na /bin/bash

Po zmianach linijka ta powinna wyglądać mniej więcej tak:

```
blueroot:x:0:0:,,,:/home/blueman:/bin/bash
```

Plik zapisujemy i wychodzimy z niego.

Możemy także usunąć utworzony katalog domowy użytkownika blueroot (/home/blueroot), skoro korzystać będzie z innego.

/etc/shadow

W tym pliku zapisane są m.in. hasła wszystkich użytkowników w systemie. Aby wyłączyć logowanie na konto root'a wystarczy zmodyfikować ręcznie hasło.

```
root:$1$w7d0W/oV$GeA.WPpb/CqScsqYbPxxTenTekstDodalemSam:13971:0:99999:7:::
```

Można też wpisać jedną literkę/znak na początku/końcu, ale bardziej bezpieczne jest zastosowanie jakiegoś ciągu znaków po którym będziemy mogli się domyśleć, że w razie konieczności włączenia konta root należy ten string usunąć.

I tyle. Teraz nie będzie można już logować się na konto roota, ponieważ jego hasło zostało zmienione na niestandardowy system zapisu hasła.

Aby zalogować się na swojego nowego root'a (blueroot) należy w konsoli wpisać:

```
$ su blueroot
```

Uzupełnić o monitorowane hasło... i cieszyć się nowym kontem root.

PS.

```
# whoami
```

Dzięki tej komendzie sprawdzimy kim tak naprawdę jesteśmy w systemie. Po zalogowaniu na *blueroot* powinien wyskoczyć napis *root*.

3.4.2. Użycie sudoers

Sposób opisany w punkcie 3.4.1 jest dobry, ale nie idealny, ponieważ często chcemy wykonać tylko 1-2 komendy jako *root*, a resztę czynności obsługiwać już normalnie, tj. z prawami dostępu normalnego użytkownika. W tym wypadku ciągle „przeskakiwanie” między kontem administratora, a użytkownika nie jest funkcjonalne i lepszym wyjściem jest użycie komendy *su*.

sudo (ang. superuser do) – jest to program stosowany w systemach operacyjnych GNU/Linux, Unix i podobnych, w celu umożliwienia użytkownikom uruchomienia aplikacji, normalnie zarezerwowanych dla administratora zwanego *rootem*.

Możliwość korzystania z tego przydatnego, lecz potencjalnie niebezpiecznego narzędzia użytkownik uzyskuje po podaniu swego hasła, co ma utrudnić wykorzystanie programu przez niepowołane osoby w nieczyntych celach. Dzięki *sudo* możemy konkretnym użytkownikom systemu przydzielać komendy, które normalnie wymagają praw *roota*. Możemy także jednemu użytkownikowi przydzielić wszystkie dostępne prawa.

Pakiet *sudoers* nie jest domyślnie dostępny w systemie Debian. Musimy więc go zainstalować:

```
# apt-get install sudo
```

Po instalacji wystarczy dopisać tylko jedną linię w pliku konfiguracyjnym */etc/sudoers*, a najlepiej skorzystać z edytora *visudo*, który ma taką właściwość, że sprawdza poprawność składni pliku. *Visudo* – wywołujemy normalnie z konsoli, po czym otworzy nam się edytor tekstu.

Aby ustawić użytkownika na pełne prawa w systemie należy dodać taką linijkę:

```
uzytkownik ALL=(ALL) ALL
```

gdzie „uzytkownik” to nazwa zwykłego użytkownika w systemie.

Teraz korzystając z przedrostka *sudo* i potwierdzając to hasłem, możemy wykonywać komendy bez notorycznego przechodzenia na konto *roota*.

```
sudo apt-get update
```

4 Instalacja i konfiguracja LAMP

(wersja elektroniczna: <http://www.blue-man.pl/serwer/id961-konfiguracja-serwera-cz3konfiguracja-serwera-cz3.html>)

Teraz przechodzimy do sedna sprawy – po wstępnej konfiguracji, pora uruchomić serwer stron Apache – dzięki temu nareszcie będziemy widzieć efekty naszej pracy (czyt. instalacji) w postaci namacalnych dowodów – wyświetlających się dynamicznych stron PHP.

Instalacja całego tego oprogramowania nie jest trudna. Sprowadza się raptem do jednego polecenia, w którym uwzględnimy wszystkie składniki jakie mają zostać zainstalowane.

```
# apt-get install apache2 libapache2-mod-auth-mysql mysql-server mysql-client  
php5 php5-mysql php5-curl php5-gd php5-memcache php5-xsl
```

Oczywiście przed tą komendą warto zaktualizować swoje repozytorium.

```
# apt-get update
```

Podczas procesu instalacji mogą wyskoczyć monity o wybraniu odpowiedniej opcji, bądź podanie domyślnego hasła root bazy danych. Myślę, że opis tego jest zbędny, ponieważ jeśli pojawi się takie „okienko”, zawsze dobrze opisane są opcje w nim zawarte.

Gdy wszystko zostanie zainstalowane i skonfigurowane, to po wpisaniu w ulubioną przeglądarkę internetową adresu IP serwera na którym został zainstalowany LAMP zobaczymy stronę podobną do tej:



(proszę nie łączyć mnie z przeglądarką Internet Explorer! Jest to najgorsza przeglądarka na Świecie, ale z racji że nie korzystałem z niej (nie miałem otwartych stron w kartach) to nadawała się idealnie do zrobienia screena :P)

Autor: **Szymon Bluma**, Strona: www.BlueMan.pl, maj 2009
Poradnik rozpowszechniany na licencji [Creative Commons BY-NC-ND](http://creativecommons.org/licenses/by-nc-nd/3.0/).

Jeśli dostaniemy informacje, że nie można połączyć się do zdalnego serwera, to znaczy, że coś przy instalacji poszło nie tak, bądź jest uruchomiony firewall, który ogranicza dostęp do serwera przez port 80. Musicie to rozwiązać na własną rękę, ponieważ nie widzę Waszych błędów i plików konfiguracyjnych, a wiadomo – z pustego nawet BlueMan nie należy, a szklaną kulę oddałem do naprawy :)

Co spostrzegawczy obserwatorzy zauważyli, że komenda nie uwzględnia instalacji phpMyAdmin. Bez obaw – powrócę do tego w dalszej części poradnika.

4.1 Konfiguracja Apache2

Podstawową aplikacją, która będzie przyjmować i obsługiwać ruch internetowy po stronie serwera jest Apache. Nazwa Apache pochodzi od indiańskiego plemienia; jego logo to kolorowe piórko.



Apache HTTP Server często nazywany po prostu Apache jest obecnie najpopularniejszym serwerem spotykanym w internecie. Serwer Apache był pierwszą alternatywą dla serwera Netscape Communication Corporations (obecnie znanego jako Sun Java System Web Server). Aplikacja Apache została stworzona przez Roberta McCool, który pracował na zlecenie National Center for Supercomputing Applications.

Domyślna instalacja Apache jest niczego sobie. Właściwie dla przeciętnego Kowalskiego nie ma co tam zmieniać, jedynie takie liftingujące rzeczy.

Plik konfiguracyjny znajduje się w drzewie:

```
/etc/apache2/apache2.conf
```

Otwieramy go ulubionym edytorem tekstu i zmieniamy kilka wartości. Jeśli jakiegoś wpisu nie znajdziesz u siebie, to warto rozważyć dodanie go. Pod spisem wszystkim wprowadzonych zmian wyjaśnię co od czego zależy.


```

ServerName MojaNazwaSerwera
Timeout 30
<IfModule mpm_prefork_module>
    StartServers          20
    MinSpareServers      15
    MaxSpareServers      30
    MaxClients            100
    MaxRequestsPerChild  1000
</IfModule>
<IfModule mpm_worker_module>
    StartServers          10
    MaxClients            100
    MinSpareThreads      20
    MaxSpareThreads      50
    ThreadsPerChild      20
    MaxRequestsPerChild  1000
</IfModule>
ServerSignature Off

```

- *ServerName* – jeśli nie mamy podanej tej wartości, to przy restarcie serwera Apache dostaniemy błąd o tym, choć będzie działać poprawnie.
- *Timeout* – ilość sekund, po których dostaniemy informację, że serwer nie odpowiada.
- *mpm_prefork_module* i *mpm_worker_module* – ilość połączeń, wątków uruchomionych na serwerze zależy przede wszystkim od ilości dostępnego ramu i obciążenia serwera. W/w wartości miałem ustawione na serwerze typu Intel Pentium4 2.66GHz, 1.2GB DDR1 RAM. Przy większych wartościach serwer zachowywał się bardzo niestabilnie – opisywałem swoje problemy na blogu (<http://www.bluelman.pl/informatyka/id417-apache.html>)
- *ServerSignature* – włącza/wyłącza wyświetlanie stopki z informacją o serwerze i zainstalowanym oprogramowaniu, m.in. przy stronie błędu pokazuje się ta informacja. Zalecam wyłączyć wyświetlanie takich informacji, aby potencjalny intruz nie wiedział jaką wersję oprogramowania mamy i do jakiej szukać dziur/exploitów.

To jaką konfigurację przyjmujemy zależy od tego jaki ruch będziemy przyjmować przez serwer. Metodą prób i błędów każdy na pewno dojdzie do swojej idealnej konfiguracji. Pamiętajcie, aby wtedy zrobić kopię bezpieczeństwa plików!

Jeśli jakiejś linijki nie potrzebujemy, to lepszym wyjściem jest jej zakomentowanie (znak hash: #), niż wyrzucenie – może kiedyś będziemy chcieli powrócić do starych ustawień?

Po wprowadzonych wszystkich zmianach nie zapomnijcie zrestartować serwer Apache, aby wczytał nowe ustawienia:

```
# /etc/init.d/apache2 restart
```

4.1.1.VirtualHost


Teraz, oprócz tego, że serwer stron www Apache funkcjonuje prawidłowo musimy posiadać wiedzę jak i gdzie umieszczać strony. Apache umożliwia przecież utrzymywanie kilku stron.

Główny katalog stron to `/var/www/` - to w nim znajduje się plik `index.html` który zawiera napis „It works!” wyświetlany po wejściu na IP serwera.

Mimo wszystko nie zalecam trzymać stron w tamtym miejscu, ponieważ jest z tym kilka problemów – główny to taki, że przez FTP nie dostaniemy się do katalogu `/var/www/`, a drugim powodem jest to, że jednak swoje strony warto trzymać w swoim katalogu domowym – w którym znacznie częściej przebywamy, niż w `/var/www/` (automatycznie po logowaniu na SSH jesteśmy w katalogu domowym – nic nie musimy zmieniać)

Jeśli mamy domenę internetową to u naszego rejestratora należy przekierować ją na własny adres IP. W dzisiejszych czasach wszyscy rejestratorzy taką opcję udostępniają – nazwa.pl, cal.pl, ovh.pl, itd.

nazwa.pl:

 **Przekierowanie na zewnętrzny adres IP**

Przekierowanie domeny na zewnętrzny adres IP umożliwia przekierowanie wpisów w DNS dotyczących WWW oraz poczty tej domeny na wskazany, zewnętrzny adres IP. W formularzu podaj adres IP, na który chcesz przekierować zamówioną domenę (np. adres 85.128.128.36).

UWAGA! - jeżeli dla domeny założone jest darmowe konto pocztowe, po dokonaniu przekierowania zostanie ono usunięte.

Adres IP:

cal.pl:

Przekierowanie domeny na IP

Aby opcja działała domena musi być ustawiona na ns1.cal.pl i ns2.cal.pl. Jeżeli masz wykupiony u nas serwer nie zmieniaj tej opcji.

Adres IP:

Przekierowanie domeny na serwer przez IP nie jest eleganckim rozwiązaniem, ale na razie musi nam ta metoda wystarczyć. W kolejnych częściach poradnika umieszczę konfigurację i ustawiania własnego serwera nazw, więc proszę się nie martwić i na razie nie krytykować tego pomysłu ;)

Kiedy już to zrobimy i domena będzie prawidłowo działać, tzn. wyświetlać przyjazny napis „It works!” pora przystąpić do ustawień VirtualHost, aby nasz adres internetowy wskazywał na inny katalog, niż domyślny `/var/www/`.

`/etc/apache2/sites-available/` - tutaj dostępne są wszystkie VirtualHost jakie są na serwerze. Zalecam tutaj tworzyć kolejne pliki, podzielone względem obsługiwanej domeny, a następnie tworzyć linki do `/etc/apache2/sites-enabled/` w którym są aktywne/działające strony na serwerze.

Stwórzmy więc VirtualHost, który będzie obsługiwać domenę BlueForum.pl. Pliki znajdują się w `/home/blueman/public_html/blueforum.pl/`

```
$ vim /etc/apache2/sites-available/blueforum
```

```
<VirtualHost *>
    ServerAdmin forum@blueforum.pl
    DocumentRoot /home/blueman/public_html/blueforum.pl/
    ServerName blueforum.pl
    ServerAlias www.blueforum.pl
    ErrorLog /var/log/apache2/blueforum.pl-error_log
    CustomLog /var/log/apache2/blueforum.pl-access_log common
</VirtualHost>
```

- **ServerAdmin** – W razie wystąpienia jakiegoś błędu, często Apache wyświetla email administratora. Tutaj możemy ustawić spersonalizowany dla konkretnej domeny.
- **DocumentRoot** – ścieżka bezwzględna do katalogu na który ma wskazywać domena. Ścieżka ta musi zostać zakończona slashem.
- **ServerName** – (sub)domena na której serwer ma „słuchać”
- **ServerAlias** – jeśli chcemy adres z przedrostkiem www, należy dodać właśnie taką linię do pliku.
- **ErrorLog** – plik w którym będą zapisywane błędy, na jakie Apache zwraca uwagę przy korzystaniu z domeny. Pojawiają się tutaj błędy o braku jakiegoś pliku, jeśli zgłoszenie o taki plik nastąpiło
- **CustomLog** – zapisuje IP i czynności wszystkich odwiedzających stronę

Należy kontrolować co jakiś czas pliki logów, ponieważ przy popularnej witrynie w bardzo szybkim tempie mogą urosnąć do bardzo dużych rozmiarów.

Jeśli plik jest już gotowy, należy go podlinkować w katalogu *sites-enabled*

```
# ln -s /etc/apache2/sites-available/blueforum /etc/apache2/sites-enabled/blueforum
```

Polecenie to jest oczywiście jedną linijką, a nie dwoma.

Przed przeładowanie serwera Apache należy utworzyć katalog który podaliśmy przy konfiguracji VirtualHost dla domeny.

```
$ mkdir /home/blueman/public_html/blueforum.pl
```

Teraz wystarczy restart serwera i gotowe:

```
blueman@bluebudg:/etc/apache2$ sudo /etc/init.d/apache2 restart  
Forcing reload of web server (apache2)... waiting .
```

Wpisując w przeglądarkę adres www.BlueForum.pl zobaczymy pliki umieszczone w */home/blueman/public_html/blueforum.pl/* a nie tak jak poprzednio (domyślnie) */var/www/*.

Subdomeny tworzymy identycznie. Kompletnie niczym to się nie różni.

```
<VirtualHost *>  
  ServerAdmin admin@forumorange.net  
  DocumentRoot /home/blueman/public_html/forumorange.net/artykuly/  
  ServerName artykuly.forumorange.net  
  ServerAlias www.artykuly.forumorange.net  
  ErrorLog /var/log/apache2/forumorange.net.artykuly-error_log  
  CustomLog /var/log/apache2/forumorange.net.artykuly-access_log common  
</VirtualHost>
```

Pamiętajcie przede wszystkim o tym, aby przy każdej zmianie zrestartować serwer.

4.2 Konfiguracja PHP5

Kolejnym składnikiem serwera internetowego jest interpreter PHP, który odpowiada za przetwarzanie plików PHP aplikacji serwisu. Dzięki niemu mamy możliwość tworzenia, dynamicznych stron internetowych.

Nazwa ta jest rekursywnym akronimem słowa *PHP Hypertext Preprocessor*. PHP jest skryptowym językiem programowania stworzony do generowania dynamicznych stron www. Jest językiem, który działa po stronie serwera, ale również może zostać wywołany z linii poleceń systemowych.



Pierwotnie PHP został stworzony w 1995 roku przez Rosmusa Lerdorfa, a obecnie jest rozwijany przez PHP Group.

Lerdorf zaczął tworzyć swój projekt „Personal Home Page Tools”, ponieważ szukał alternatywy do języka Perl, którego używał na swojej domowej stronie internetowej. Potrzebował nowych funkcjonalności do wyświetlania swojego curriculum vitae, oraz analizowania ile ruchu (ściągnięć, itp.) generuje jego strona oraz samo CV. Tworząc PHP zaimplementował dużą ilość bibliotek napisanych w języku C, dzięki którym była możliwość komunikacji z bazami danych dostępnymi na serwerze. Rosmus zdecydował się 8 czerwca 1995 udostępnić swoje dzieło, aby przyspieszyć znajdowanie luk bezpieczeństwa i poprawę kodu, wykorzystując do tego pomoc innych internautów. Wydanie to zostało nazwane wersją PHP2 i posiada swoją podstawową funkcjonalność po dzień dzisiejszy. Składnia nowego języka podobna jest do Perla – posiada zmienne, możliwość umieszczania kodu HTML. Ale mimo podobieństwa do języka Perl, PHP jest bardziej ograniczony i prostszy.

Także i w tym wypadku domyślny plik konfiguracyjny jest niemal bezbłędny. PHP to tylko interpreter, a to jak szybko będzie działać zależy przecież od programisty, który pisze stronę www – na nim spoczywa odpowiedzialność za wydajność witryny.

Plik `php.ini` znajduje się w prostej do zapamiętania lokalizacji:
`/etc/php5/apache2/php.ini`

Tak jak w przypadku Apache (a także każdego innego oprogramowania) – polecam komentowanie, i w ten sposób zostawianie starych linijek konfiguracji, niż ich zastępowanie. Jeśli coś pójdzie nie tak będziemy mogli łatwo wrócić do poprawnej, działającej wersji.

```
max_execution_time = 60
memory_limit = 128M
upload_max_filesize = 50M
```

- *max_execution_time* – ilość czasu w sekundach, przez jaką skrypt maksymalnie może się wykonywać.
- *memory_limit* – wielkość pamięci RAM z jakiej PHP może korzystać. Przy operacji (odczytanie/tworzenie) dużych plików warto ustawić wysoką wartość. Nie zalecam wartości poniżej 10MB
- *upload_max_filesize* – maksymalna wartość jaką możemy przesłać przez formularz. Dzięki temu także w phpmyadmin będziemy mogli uploadować duże bazy danych.

Jak widzicie dużo do zmiany nie było, i wcale nie wpływają one na zwiększenie wydajności aplikacji internetowych.

Jak już wszystkie zmiany wykonamy, to pozostaje tylko zrestartować serwer

```
# /etc/init.d/apache2 restart
```

Jeśli restart serwera nastąpił bez błędów, a mimo wszystko mamy wątpliwości, czy wartości wpisane w PHP są prawidłowo zapisane stwórzmy plik *phpinfo.php*, umieścimy go w */var/www/*, a w nim wpisujemy:

```
<?php
phpinfo();
?>
```

Jeśli w przeglądarce wpisujemy <http://ip.serwera/phpinfo.php> to powinniśmy zobaczyć stronę z wszystkimi opcjami jakie są ustawione w serwerze, bazie danych oraz interpreterze PHP.

PHP Version 5.2.6-1+lenny2	
System	Linux bluepower 2.6.26-1-amd64 #1 SMP Sat Jan 10 17:57:00 UTC 2009 x86_64
Build Date	Jan 26 2009 21:45:09
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File	/etc/php5/apache2

4.3 Konfiguracja MySQL5

Można powiedzieć, że temat rzeka. Istnieje tak dużo opcji konfiguracyjnych serwera baz danych, że nie sposób je wszystkie wymienić, a tym bardziej nauczyć się dobierać odpowiednie ich parametry. Trzeba mieć naprawdę porządne doświadczenie w tej sprawie, aby to ogarnąć.

Początkowo w mojej historii z serwerami www, znajomy optymalizował mi serwer bazy danych. Potem sam próbowałem zmieniać wartości (po dołożeniu pamięci ram), ale niestety zakończyły się one fiaskiem i baza danych działała jeszcze wolniej.

Jeśli chodzi o optymalizację MySQL, to naprawdę warto poświęcić kilka dni/tygodni na to, ponieważ daje to bardzo porządne efekty.

Domyślny plik `/etc/mysql/my.cnf` można śmiało wyrzucić do kosza i zbudować swój własny.

Mój plik `/etc/mysql/my.cnf` wygląda następująco:

```
[mysqld]

tmpdir = /tmp
set-variable = max_connections=30
max_user_connections=24
safe-show-database
#skip-locking
skip-innodb
skip-external-locking
key_buffer = 16M
sort_buffer_size = 64M
record_buffer = 32M
table_cache = 2000
thread_cache_size = 256
tmp_table_size = 190M #128M
read_rnd_buffer_size = 768K
read_buffer_size = 8M
max_allowed_packet = 16M
query_cache_limit = 8M
query_cache_size = 35M
query_cache_type = 1
thread_concurrency = 4
default-character-set = utf8

# dodano po optymalizacji
join_buffer_size = 256K
max_heap_table_size = 50M
log-slow-queries = /var/log/mysql-slow-queries.log
long_query_time = 5
open_files_limit = 5000

!includedir /etc/mysql/conf.d/
```

Oczywiście każdy musi własne wartości dobrać – względem tego jaki sprzęt ma w serwerze, oraz ile odwiedzin mają jego wszystkie strony korzystające z bazy danych.

Jak wspomniałem wcześniej – moje wcześniejsze próby samodzielnego eksperymentowania w dobierania odpowiednich wartości zakańczyły się fiaskiem. Dlatego teraz – w celu poprawnej optymalizacji, korzystam z gotowego oprogramowania, które wspomaga mnie w podejmowaniu decyzji dobór odpowiedniej konfiguracji. Są dwa narzędzia do tego – mysqltunner.com oraz [tuning-primer](#).

Pierwsze uboższe w możliwości, ale podaje bardziej przejrzyste wyniki. Nie sugeruje konkretnych wartości, a tylko przedziały.

Drugie to właściwie kombajn. Właśnie dzięki niemu zawdzięczam to, że serwer bazy danych chodzi stabilnie, a strony wczytują się błyskawicznie.

Po uruchomieniu obydwu z narzędzi będziemy proszeni o podanie loginu i hasła użytkownika bazy danych. Kiedy już to uzupełnimy nastąpi proces analizy użytkowanej bazy danych. Zbieranie informacji nie trwa długo, ale serwer musi być uruchomiony od co najmniej 48 godzin, aby wyniki analizy były sensowne.

Trzeba teraz przeanalizować zasugerowane wartości i wprowadzić odpowiednie poprawki do pliku `/etc/mysql/my.cnf`.

Po wszystkich zmianach należy zrestartować serwer

```
# /etc/init.d/mysql restart
```

Odczekać kolejne ~48 godzin i ponownie sprawdzić wyniki testów uzyskanych przez te programy.

4.4 phpMyAdmin

<http://www.phpmyadmin.net/> Oprogramowanie to jest znane przez każdego programistę PHP. Dzięki niemu w prosty sposób możemy wykonywać wszystkie operacje na bazie danych.



Nie musimy mieć uprawnień administratora systemu, aby móc korzystać z tego skryptu. Jest to najwyczejniejszy w życiu skrypt PHP. Często ściągam go i uploaduje na serwer klienta, jeśli nie udostępnia on mi go, lub po prostu zapomniałem poprosić o dostęp do niego, a akurat na konkretny wieczór zaplanowałem wdrażanie aplikacji.

W systemie Debian można w bardzo prosty sposób zainstalować phpMyAdmin poprzez jedno polecenie.

```
# apt-get install phpmyadmin
```

I dostęp do niego mamy jako strona mieszcząca się pod adresem <http://ip.serwera/phpmyadmin/>

I właściwie nic więcej nie musimy robić, aby działało to dobrze. Jednak bardziej spostrzegawczy programiści zauważają, że udostępniana w repozytorium wersja skryptu nie należy do najnowszych. Dlatego ja wybieram zwykle instalacje na własną rękę, która jest równie banalna jak przez repozytorium.

Przechodzę do katalogu `/var/www`

Ściągam ze strony paczkę z najnowszą stabilną wersją skryptu

```
$ wget http://dfn.dl.sourceforge.net/sourceforge/phpmyadmin/phpMyAdmin-3.1.3.1-all-languages.zip
```

Rozpakowuję ją i zmieniam nazwę folderu.

```
$ unzip phpMyAdmin-3.1.3.1-all-languages.zip
$ mv phpMyAdmin-3.1.3.1-all-languages phpmyadmin
```

I tyle :) Bez żadnej dodatkowej konfiguracji phpMyAdmin działa tak samo jak przy instalacji z pakietów repozytorium. A w ten sposób mogę bardzo łatwo i na własną rękę zaktualizować skrypt do nowszej wersji.

5 Bind – serwer nazw

(Wersja elektroniczna: <http://www.blue-man.pl/serwer/id984-konfiguracja-serwera-cz4.html>)

BIND (Berkeley Internet Name Domain, poprzednio: Berkeley Internet Name Daemon) jest popularnym serwerem (demonem) DNS. Został on stworzony przez Paula Vixie w roku 1988. BIND jest jednym z najpopularniejszych serwerów DNS wykorzystywanym w systemach Linux i Unix. BIND stanowi niezmiernie ważny składnik zapewniający poprawne działanie systemu nazw w Internecie. Wielu użytkowników globalnej sieci bezwiednie korzysta z serwera BIND, kiedy ich przeglądarka WWW odpytuje go o adres IP komputera udostępniającego interesującą ich stronę.

Alternatywnym odpowiednikiem BIND jest PowerDNS, który działa z wykorzystaniem bazy danych (<http://www.powerdns.com/>). Teoretycznie działa szybciej i mniej będzie obciążać serwer dla dużej ilości domen, ale praktycznie nigdy nie miałem okazji tego sprawdzić. Kolejnym plusem jest to, że jednym zapytaniem SQL można zmienić adres IP wszystkich domen w systemie. Dla Binda trzeba pisać/pobrać skrypt Bash do tego.

Skupię się na BINDie, ponieważ jest najpopularniejszy i znacznie łatwiej uzyskać do niego pomoc, która może okazać się niezbędna początkującym administratorom.

5.1 Instalacja i zabezpieczenia

Historia wersji BIND jasno mówi, że od początku zabezpieczenia nie były jego mocną stroną i od początku był dziurawa aplikacją. Poświęćmy więc chwilę czasu na jego załatanie.

```
# apt-get install bind9
```

Po instalacji od razu zatrzymajmy serwer bind, aby wprowadzone zmiany nie odbywały się na uruchomionej aplikacji.

```
# /etc/init.d/bind9 stop
```

```
# /etc/init.d/bind9 status  
bind9 is not running failed!
```

Zmienimy teraz zawartość pliku `/etc/default/bind9` aby daemon był uruchamiany jako nieuprzywilejowany użytkownik `bind`, który będzie chrootowany w `/var/lib/named`.

Zamień odpowiednią linijką na tą:

```
OPTIONS="-u bind -t /var/lib/named"  
#Set RESOLVCONF=no to not run resolvconf  
RESOLVCONF=yes
```

Następnie utwórz niezbędne katalogi:

```
# mkdir -p /var/lib/named/etc  
# mkdir /var/lib/named/dev  
# mkdir -p /var/lib/named/var/cache/bind  
# mkdir -p /var/lib/named/var/run/bind/run
```

Następnie główny katalog konfiguracyjny przesunąć z domyślnej lokalizacji `/etc/bind`

```
# mv /etc/bind /var/lib/named/etc
```

Utwórz także link symboliczny starej lokalizacji i nowej. Unikniemy w ten sposób problemów w przyszłości jeśli aplikacja będzie aktualizowana, bądź w razie potrzeby, kiedy inna aplikacja chciała by skorzystać z binda.

```
# ln -s /var/lib/named/etc/bind /etc/bind
```

Wykonaj następnie poniższe komendy:

```
# mknod /var/lib/named/dev/null c 1 3  
# mknod /var/lib/named/dev/random c 1 8  
# chmod 666 /var/lib/named/dev/null /var/lib/named/dev/random  
# chown -R bind:bind /var/lib/named/var/*  
# chown -R bind:bind /var/lib/named/etc/bind
```

Teraz powinniśmy zmienić plik `/etc/default/syslogd`, aby wciąż niezbędne komunikaty o ewentualnych błędach były zapisywane do logów.

Jeśli w naszym systemie nie mamy ww. pliku to trzeba go „doinstalować”, wraz z odpowiednią aplikacją.

Wpisujemy w konsoli zatem:

```
# apt-get install sysklogd
```

A potem edytujemy plik

```
# vim /etc/default/syslogd
```

```
SYSLOGD="-a /var/lib/named/dev/log"
```

Na koniec restartujemy demon od zarządzania logami, oraz uruchamiamy serwer bind.

```
# /etc/init.d/sysklogd restart
Restarting system log daemon...

# /etc/init.d/bind9 start
Starting domain name service...: bind9.
```

Po tych czynnościach mamy już sprawny, działający deamon BIND. Nie musimy pamiętać nowej lokalizacji plików (*/var/lib/named/etc*), ponieważ utworzyliśmy link do */etc/bind* i z niego będziemy korzystać.

5.2 Własne NameSery

Przedstawiony poniżej sposób przeze mnie jest chyba najgorszym z możliwych – i zdaję sobie z tego sprawę. Ale biorąc pod uwagę to, że nie dysponuję drugim serwerem dedykowanym z możliwością wglądu do binda, nie zagłębiałem się specjalnie w to jakie możliwości daje to lepsze rozwiązanie. Metoda przedstawiona przeze mnie jest sprawnie działającą aplikacją. Przez ponad rok czasu metoda ta nie zawiodła mnie, więc uznaję ją jako „sprawdzoną, działającą”.

Wszystko rozchodzi się o to, aby ustawione ns1.blue-man.pl i ns2.blue-man.pl były na różnych serwerach, oraz aby wzajemnie wymieniały się danymi. Ja to uruchamiam na jednym serwerze – przez rok czasu nie miałem żadnych problemów z BINDem, a więc nie musiał się posiłkować działającym na innej maszynie serwerem secondary.

Można powiedzieć, że jest to temat rzeka. Zainteresowanych odsyłam do www.google.pl po więcej informacji ;)

Przedstawiony poniżej opis konfiguracji będę opisywać na przykładzie mojej domeny ania.li, którą kupiłem tylko dla potrzeb serwerów nazw i ogólnej przydatności serwerowej. Jest to najtańsza domena jeśli chodzi o odnowienie – w www.ovh.pl kosztuje 12.99 zł netto. A zyskałem tym samym prostą 4-literową domenę z imieniem mojej narzeczonej :)

Konfigurację rozpoczynamy od zajrzenia do pliku /etc/bind/named.conf

```
# vim /etc/bind/named.conf
```

Zawiera on (nie wiem do końca jak to określić...) spis wszystkich domen, wraz z parametrami w jakim trybie domena jest „uruchomiona”. Domena może być uruchamiana jako „master”, oraz „slave” (oraz pewnie także w innych o których nie mam pojęcia).

Tuż za ostatnim wpisem:

```
zone "255.in-addr.arpa" {  
    type master;  
    file "/etc/bind/db.255";  
};
```

wstawmy informację o nowej strefie domeny którą będziemy ustawiać.

```
zone "ania.li" {
    type master;
    file "/etc/bind/db.ania.li";
};
```

Musimy także utworzyć plik *db.ania.li* w katalogu */etc/bind*. Nazewnictwo plików i ich położenie nie ma większego znaczenia – ważne, aby aplikacja BIND miała do nich dostęp (uprawnienia).

Plik *db.ania.li* powinien tak wyglądać:

```
$TTL 86400
@      IN      SOA      ns1.ania.li.  root.ania.li. (
        2009041703
        3600
        3600
        3600000
        1209600
        )

;
@      IN      NS       ns1.ania.li.
@      IN      NS       ns2.ania.li.
;
@      IN      MX       10      mail.ania.li.
;
@      IN      A        83.242.78.666
www    IN      CNAME    ania.li.
ns1    IN      A        83.242.78.666
ns2    IN      A        83.242.78.666
```

BIND jest dość skomplikowaną aplikacją i początkujący administrator będzie mieć na pewno problem, aby go ujarzmić. Postaram się zmniejszyć ten ból wyjaśniając problemy z jakimi ja się napotykałem na swojej drodze.

- @ IN SOA ns1.ania.li. root.ania.li.
ns1.ania.li – jest to adres master nameserwer
root.ania.li – można tutaj właściwie wstawić dowolny tekst. Nie wiem od czego to zależy
- 2009041703
Bardzo ważna linijka – tzw. SERIAL! Przy każdej zmianie w pliku binda domeny musimy tutaj podać nową cyfrę (serial). Możemy numerować to dowolnie – 1, 2, 3, 4, 5, 6, 7, 8, a także 1, 2, 1, 2, 1, 2. Jednak przyjęt się pewien standard, który chyba wszyscy używają. Pierwsze 4 cyfry to ROK, kolejne 4 to MIESIĄC oraz DZIEŃ. A następne 2 cyfry to numeracja ilości zmian pliku w danym dniu. W moim przypadku ostatnio w tym pliku „grzebałem” 17 kwietnia 2009, trzykrotnie zmieniając jego zawartość w tym dniu.

Część pliku po zamykającym nawiasie zwykłym zawiera już konkretną zawartość strefy, która jest widoczna i odpytywana przez zewnętrzne serwery.

```
@      IN      NS      ns1.ania.li.  
@      IN      NS      ns2.ania.li.
```

Określa na jakie NameSeryery wskazuje domena, które poniżej zostały zadeklarowane

```
ns1    IN      A      83.242.78.666  
ns2    IN      A      83.242.78.666
```

Możesz użyć tutaj dowolnych nazw – zaproponowane przeze mnie *ns1* i *ns2* są tylko przykładami. Równie dobrze możesz zamiast nich wpisać *reks* i *burek*. Pamiętaj, aby zmiany te zrobić wszędzie, a nie tylko w linijce przy adresie IP!

```
@      IN      A      83.242.78.666  
www    IN      CNAME  ania.li.
```

Linijki te wysyłają przychodzące zgłoszenie z adresu *.ania.li oraz www.ania.li do serwera o wskazanym IP.

Znak małpy @ oznacza po prostu „dla wszystkich adresów” w sensie (subdomen).

Typy wpisów:

- A podajemy adresy IP, na które ma wskazywać dany wpis,
- AAAA jak wyżej, tyle, że adresy typu Ipv6
- NS podajemy nazwy (nie adresy IP!) serwerów, które utrzymują daną strefę,
- MX podajemy nazwy serwerów obsługujących pocztę,
- CNAME podajemy aliasy na inne domeny.

Pamiętajcie!

Po słownych nazwach serwerów zawsze trzeba wstawić kropkę. Po IP nie wolno tego robić !

Tym samym mamy już skonfigurowane i działające własne dwa serwery nazw :] Możemy ich używać podpinając kolejną domenę do serwera.

Przy każdej zmianie wpisu domeny trzeba zmieniać SERIAL na nowy, aby aplikacja pobrała nowe wartości.

Musimy także zrestartować cały bind jeśli wprowadzamy jakieś zmiany

```
# /etc/init.d/bind restart
```

5.3 Strefa dla nowej domeny

Nim jednak to zrobimy w panelu zarządzania domeną u naszego domenowego providera musimy najpierw na naszym serwerze utworzyć jej strefę.

Edytujemy więc plik `/etc/bind/named.conf`
I tuż przed (który jest na końcu pliku)

```
include "/etc/bind/named.conf.local";
```

dodajemy

```
zone "moja-domena.pl" { type master; file "/etc/bind/db.moja-domena.pl"; };
```

Wychodzimy z pliku `named.conf` uprzednio go zapisując. Oraz tworzymy plik `/etc/bind/db.moja-domena.pl` z zawartością:

```
$TTL 86400
@      IN      SOA      ns1.ania.li.    root.moja-domena.pl. (
      2009050301
      3600
      3600
      3600000
      1209600
      )

;
@      IN      NS       ns1.ania.li.
@      IN      NS       ns2.ania.li.
;
@      IN      MX       10    mail.moja-domena.pl.
;
@      IN      A        83.242.78.666
www    IN      A        83.242.78.666
```

Zwróćcie proszę uwagę na to, aby w rekordzie *SOA* było odwołanie do naszego *NS*, oraz także *@ IN NS* wskazywało na nasze główne *NS*. A także na to, aby *SERIAL* był nowy (przy kolejnych zmianach tego pliku).

Jeśli wszystko wprowadziliśmy dobrze i po restarcie binda

```
# /etc/init.d/bind restart
```

Serwer nie zakomunikował żadnych błędów możemy się cieszyć :)

Aby sprawdzić, czy strefa nowej domeny została prawidłowo załadowana, wykonujemy polecenie:

```
# named -g 2>&1 | grep loaded
```

Gdy już nic nowego nie będzie pojawiać się na wydruku ekranu naciskamy Ctrl+C oraz szukamy, czy nasza nazwa domeny jest na ekranie.

Wydruk powinien wyglądać mniej więcej tak:

```
03-May-2009 01:23:01.574 zone 0.in-addr.arpa/IN: loaded serial 1
03-May-2009 01:23:01.575 zone 127.in-addr.arpa/IN: loaded serial 1
03-May-2009 01:23:01.576 zone 255.in-addr.arpa/IN: loaded serial 1
03-May-2009 01:23:01.577 zone ania.li/IN: loaded serial 2009041703
03-May-2009 01:23:01.578 zone localhost/IN: loaded serial 2
03-May-2009 01:23:01.578 zone moja-domena.pl/IN: loaded serial 2009050301
```

Jeśli wszystko jest OK, to możemy zmienić NS domeny na własne w firmie w której wykupiliśmy naszą nazwę. Powinno zadziałać za pierwszym razem.

6 Dodatkowe składniki

(Wersja elektroniczna: <http://www.bluelman.pl/nieprzypisany/id994-konfiguracja-serwera-cz5.html>)

Serwer bez tych dodatkowych składników może i będzie funkcjonować bez żadnego zająknięcia. Jednak jeśli chcemy mieć serwer taki, jak w innych firmach hostingowych to musimy doinstalować jeszcze trochę oprogramowania do pełnego wykorzystania naszych aplikacji internetowych.

W tym dziale opiszę więc instalację/konfigurację: postfix, proftpd, eaccelerator, gzip.

6.1 Postfix

Postfix jest niezbędnym pakietem do podstawowej obsługi poczty na serwerze. Dzięki niemu będziemy mogli wysyłać email z poziomu PHP poprzez wbudowaną funkcję mail().

```
# apt-get install postfix libsasl2-2 sasl2-bin libsasl2-modules procmail
```

UWAGA !!

Niektóre pakiety mogą nie występować w starszych wersjach Debiana. Przypominam, że poradnik oparty jest na systemie Debian 5.0.

Przy instalacji wyskoczy monit o podanie kilku wartości, które zostaną zapisane do konfiguracji.

```
General type of mail configuration:  
System mail name:
```

W pierwszym zapytaniu proszę wybrać „Internet Site”, a w drugim wpisać nazwę głównej domeny na serwerze – w moim przypadku jest to po prostu „ania.li”.

Kiedy instalacja definitywnie się zakończy, należy oprócz tych podstawowych dwóch opcji ustawić także inne.

Wpisujemy więc normalnie w konsoli:

```
# dpkg-reconfigure postfix
```

I ponownie zostaniemy zapytani o kilka pytań. Większość pól powinna być już poprawnie uzupełniona. Odpowiadamy wg wzorca podanego przeze mnie (z prawej strony umieściłem sugerowaną odpowiedź).

```

General type of mail configuration:                [Internet Site]
System mail name:                                [ania.li]
Root and postmaster mail recipient:              [zostaw puste]
Other destinations to accept mail for (blank for none): [ania.li,
                                                    localhost.ania.li, localhost.localdomain, localhost]
Force synchronous updates on mail queue?        [No]
Local networks:                                  [127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128]
Use procmail for local delivery?                 [Yes]
Mailbox size limit (bytes):                      [0]
Local address extension character:               [+ ]
Internet protocols to use:                       [all]

```

Zwróćcie uwagę na:

- „*Other destinations to accept mail for (blank for none)*” - w dokumencie nie zmieściła mi się odpowiedź w jednej linijce, ale Wy zapiszcie to razem.
- „*Local networks*” - dodatkowe kwadratowe nawiasy wewnątrz odpowiedzi oznaczają adresację IPv6. Nie pomijajcie ich – zapiszcie dokładnie tak jak zostało pokazane.

Następnie wykonaj takie polecenia:

```

# postconf -e 'smtpd_sasl_local_domain ='
# postconf -e 'smtpd_sasl_auth_enable = yes'
# postconf -e 'smtpd_sasl_security_options = noanonymous'
# postconf -e 'broken_sasl_auth_clients = yes'
# postconf -e 'smtpd_sasl_authenticated_header = yes'
# postconf -e 'smtpd_recipient_restrictions =
permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination'
# postconf -e 'inet_interfaces = all'
# echo 'pwcheck_method: saslauthd' >> /etc/postfix/sasl/smtpd.conf
# echo 'mech_list: plain login' >> /etc/postfix/sasl/smtpd.conf

```

6 (słownie: szóste) polecenia od góry także nie zmieściło się w całości w linijce w dokumencie. Zwróćcie na nie proszę uwagę, aby nie wystąpiły żadne błędy przy wykonaniu jego. Można na przykład skopiować całość do swojego ulubionego edytora tekstu (typu: Notatnik, Gedit, Kate), usunąć zbędne przejścia do nowej i wtedy całość wkleić do konsoli.

Następnie musimy stworzyć certyfikat dla TLS:

```

# mkdir /etc/postfix/ssl
# cd /etc/postfix/ssl/
# openssl genrsa -des3 -rand /etc/hosts -out smtpd.key 1024

```

W tym i każdym innym wywołaniu polecenia „openssl” zostaniemy zapytani o

„hasło”. Ja przywykłem podawać tam dowolny ciąg znaków, ale nie moje hasło systemowe. W każdym poleceniu identyczny klucz podaję, ponieważ sam do końca nie wiem od czego zależą te polecenia.

```
# chmod 600 smtpd.key
# openssl req -new -key smtpd.key -out smtpd.csr
```

```
# openssl x509 -req -days 3650 -in smtpd.csr -signkey smtpd.key -out smtpd.crt
```

```
# openssl rsa -in smtpd.key -out smtpd.key.unencrypted
```

```
# mv -f smtpd.key.unencrypted smtpd.key
# openssl req -new -x509 -extensions v3_ca -keyout cakey.pem -out cacert.pem
-days 3650
```

Następnie musimy skonfigurować Postfix, aby korzystał z TLS.

```
# postconf -e 'myhostname = ania.li'
```

UWAGA!!

Upewnij się, że zamiast ania.li wprowadziłeś swoją nazwę domeny.

```
# postconf -e 'smtpd_tls_auth_only = no'
# postconf -e 'smtp_use_tls = yes'
# postconf -e 'smtpd_use_tls = yes'
# postconf -e 'smtp_tls_note_starttls_offer = yes'
# postconf -e 'smtpd_tls_key_file = /etc/postfix/ssl/smtpd.key'
# postconf -e 'smtpd_tls_cert_file = /etc/postfix/ssl/smtpd.crt'
# postconf -e 'smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem'
# postconf -e 'smtpd_tls_loglevel = 1'
# postconf -e 'smtpd_tls_received_header = yes'
# postconf -e 'smtpd_tls_session_cache_timeout = 3600s'
# postconf -e 'tls_random_source = dev:/dev/urandom'
```

Konfiguracja już skończona. Aby zobaczyć jak wygląda plik konfiguracyjny postfixa, to można go podejrzeć w lokalizacji: `/etc/postfix/main.cf`
I radzę to zrobić – choćby dlatego, aby przejrzeć jego wartości dla jakiej domeny został ustawiony.

Autoryzacja serwera, który wysyła email jest poprzez *saslauthd*. Musimy zatem zmienić kilka rzeczy, aby *Postfix* działał poprawnie, ponieważ obecnie jest w chrootowanym środowisku w */var/spool/postfix*.

```
# mkdir -p /var/spool/postfix/var/run/saslauthd
```

Następnie musimy włączyć *saslauthd* w jego pliku konfiguracyjnym.

```
# vim /etc/default/saslauthd
```

UWAGA!!

Jeśli otwarty plik jest pusty:

Na parę przypadków instalowania postfix zauważyłem, że na niektórych serwerach był, a na niektórych nie był zainstalowany plik *saslauthd*. Dlatego otwierając ten plik w celu zmiany *START* i *OPTIONS* uzyskiwałem pusty plik – wynik braku domyślnego pliku pakietu.

Aby naprawić te braki należy zainstalować:

```
# apt-get install sasl2-bin libsasl2-2 libsasl2-modules
```

I ponownie otworzyć plik w celu jego edycji.

Zmień jego domyślne wartości *START* i *OPTIONS* na następujące:

```
#Should saslauthd run automatically on startup? (default: no)
START=yes

#OPTIONS="-c -m /var/run/saslauthd"
OPTIONS="-c -m /var/spool/postfix/var/run/saslauthd -r"
```

Zamykamy, zapisując plik i wykonujemy kolejną komendę:

```
# adduser postfix sasl
```

I po tym restartujemy oba deamony

```
# /etc/init.d/postfix restart
# /etc/init.d/saslauthd restart
```

6.1.1. Czy działa Postfix?

Sprawdzimy teraz, czy wysyłanie email działa poprawnie. Najpierw rzucimy okiem w systemie, czy wszystko jest uruchomione, a potem wykonamy prosty test z użyciem mail() w PHP.

```
# telnet localhost 25
```

UWAGA!!

Tutaj także możemy nie mieć odpowiedniego pakietu do telnetu. Instalacja jego jest banalnie prosta:

```
# apt-get install telnet
```

Kiedy połączenie będzie już ustanowione z Twoim Postfix'em, to w konsoli wpisz:

```
ehlo localhost
```

Jeśli zobaczysz:

```
250-STARTTLS
250-AUTH LOGIN PLAIN
```

To znaczy, że wszystko działa. W moim przypadku na ekranie zobaczyłem:

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 ania.li ESMTP Postfix (Debian/GNU)
ehlo localhost
250-ania.li
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
```

Aby wyjść z linii komend telnetu wpisz „quit”.

6.1.2. Czy działa mail() ?

Teraz pora na test, czy email dochodzą na naszą prawdziwą skrzynkę email. Proponuję skorzystać ze skrzynki <http://poczta.o2.pl/> - nigdy nie miałem z nią problemów przy takim testowaniu. Niestety, ale Gmail nie zawsze ufa nowym serwerom pocztowym i wiadomości mogą do niego nie dochodzić.

Stwórz więc plik PHP do którego będziesz mógł się dostać z poziomu przeglądarki www, wpisując odpowiedni adres.

Zakładam, że nie masz jeszcze podczepionej żadnej domeny, więc dostęp do strony będzie odbywać się poprzez adres IP serwera. W związku z tym testowy plik musisz umieścić w `/var/www`.

```
$ vim /var/www/mail-test.php
```

UWAGA!!

Jeśli nasz użytkownik systemowy nie ma praw zapisu do tego katalogu, to wykonajcie to polecenie jako root.

W pliku umieśćcie taki kod:

```
<?php
$name = "Moj serwer"; //nazwa adresata
$email = "email@adres.com"; //email adresata
$recipient = "twoj_adres_email@o2.pl"; //email odbiorcy
$mail_body = "Działa mi wysyłanie email"; //Tresc wiadomosci
$subject = "BlueMan.pl - test funkcji email"; //Temat wiadomosci
$header = "From: ".$name." <".$email.">\r\n"; //naglowki

if ( mail($recipient, $subject, $mail_body, $header) )
    echo 'Email wysłano';
else
    echo 'Bład - nie można wysłać wiadomości';
?>
```

A następnie uruchomcie go wpisując w przeglądarce:

<http://twoj-adres-IP/mail-test.php>

Niemalże od razu na skrzynkę powinniście dostać wiadomość. Jeśli tak się nie dzieje, i przez najbliższe max 3 godz także nic nie dostaliście to może to oznaczać dwa problemy:

1. Źle zainstalowałeś/skonfigurowałeś swojego deamona
2. Twój adres IP jest na światowej czarnej spam-liście.

Jeśli test uda się dla o2.pl sprawdźcie także dla innych skrzynek. Jak zawsze najbardziej problematyczny jest Gmail.com ;)

Może się też tak zdarzyć, że chcąc uruchomić serwer w domu, który ma zewnętrzne IP, ale korzystacie z internetu dostarczanego przez jakiegoś lokalnego providera (kablówka, etc.), to cała pula adresów tego usługodawcy jest dodana do czarnej spam-listy.



<http://www.spamhaus.org/pbl/> - tutaj możecie sprawdzić, czy Wasz IP znajduje się na niej. Można także wypełnić wniosek o odblokowanie adresu IP.

Zdarzyło mi się, że dwa razy moje adresy IP były uznane jako SPAM, ale po wypełnieniu wniosku, na następnny dzień blokada ta została usunięta.

Gmail i inne skrzynki zaczęły odbierać email odemnie i już nie miałem z tym problemów. Jednak systematycznie sprawdzam, czy oby nie powrócił mój adres na ich listę – może się tak zdarzyć, więc miejcie to na uwadze. Tym bardziej, że przez pierwszy okres po odbanowaniu serwis sprawdza, czy nie robimy jakiś niekoszernych rzeczy na tym IP w związku z wysyłaniem email – miejcie się na baczności :)

6.2 ProFTPd

Najbardziej popularny (o ile nie monopolista) serwer FTP który można zainstalować na naszym serwerze. Można, ale nie trzeba – wystarczy ściągnąć program [WinSCP](#) i dzięki niemu przeglądać i uploadować pliki na serwer. Jednakże, niektóre aplikacje PHP chcą, aby podać im także dane dostępowe do FTPa (np. WordPress, SMF) w momencie instalowania nowych dodatków.

Instalacja tego programu sprowadza się tylko i wyłącznie do dwóch kroków:

```
# apt-get install proftpd
```

W zależności jak często będziemy korzystać z serwera FTP wybierzmy opcję, która bardziej nam odpowiada – *standalone* lub *inted*.

Następnie dodajmy 3 wpisy do pliku konfiguracyjnego:

```
DefaultRoot ~  
IdentLookups off  
ServerIdent on "FTP Server ready."
```

Niektórzy jeszcze zmieniają port na którym FTP będzie działać. Jest to dodatkowy ukłon w stronę bezpieczeństwa, ale ja się do niego nie stosuję.

Pozostaje więc tylko zrestartować FTP, aby działał z nowymi ustawieniami

```
# /etc/init.d/proftpd restart
```

6.3 Eaccelerator

Dzięki temu dodatkowi do PHP, część skryptów będzie w jakiś automagiczny sposób cachowana. Uzyskamy w ten sposób bardzo duże przyśpieszenie wczytywania się stron, poprzez skrócenie czasu generowania wynikowego kodu z PHP.

<http://eaccelerator.net/> - strona główna projektu.

Jak widzicie ostatnia wersja (0.9.5.3) wyszła w maju 2008 roku. Ale nie martwcie się – działa z wszystkimi wersjami PHP (obecnie najnowsza 5.2.9).

Instalacja jest bardzo prosta i została wystarczająco wyjaśniona na stronie projektu (<http://eaccelerator.net/wiki/InstallFromSource>). Ja napiszę to jeszcze raz, ale po polsku i na końcu wyjaśnię jedną ważną rzecz.

Na początku stwórzmy w `/var/www` plik w którym sprawdzimy czy mamy i czy poprawnie zainstalowaliśmy.


```
# vim /var/www/phpinfo.php
```

```
<?php
phpinfo();
?>
```

Po wywołaniu tego pliku w przeglądarce uzyskamy znany wszystkim ekran wartości jakimi posługuje się nasza instalacja PHP5.

Nasz interesuje „stopka” tuż pod pierwszą tabelką.

Stream Socket	
Transports	
Registered Stream Filters	string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, convert.iconv.*, bzip2.*, zlib.*

This program makes use of the Zend Scripting Language Engine: Zend Engine v2.2.0, Copyright (c) 1998-2006 Zend Technologies	Powered By 
--	---

Ale ani słowa w niej (ani całym pliku), że korzystamy z eAccelerator. Pora więc go zainstalować. Niestety, ale nie ma gotowej paczki dla systemu Debian i będziemy musieli ręcznie to wykonać. Zanim jednak zainstalujemy główny pakiet, musimy uzupełnić nasz system o inne niezbędne składniki.

```
# apt-get install build-essential php5-dev
```

Po tym możemy już zainstalować główną paczkę eAccelerator:

```
# cd /tmp
# wget http://bart.eaccelerator.net/source/0.9.5.3/eaccelerator-0.9.5.3.tar.bz2
# tar xvfj eaccelerator-0.9.5.3.tar.bz2
# cd eaccelerator-0.9.5.3
# phpize
# ./configure
# make
# make install
```

Cacher został już zainstalowany. Teraz pora powiedzieć o tym PHP, aby zaczął go używać.

Pliki konfiguracyjne PHP5 w systemie Debian Lenny są w `/etc/php5/conf.d/` i właśnie tam powinniśmy utworzyć dodatkowy plik z opcjami eAccelerator'a.

```
# vim /etc/php5/conf.d/eaccelerator.ini
```

```
extension="eaccelerator.so"
eaccelerator.shm_size="16"
eaccelerator.cache_dir="/var/log/eaccelerator"
eaccelerator.enable="1"
eaccelerator.optimizer="1"
eaccelerator.check_mtime="1"
eaccelerator.debug="0"
eaccelerator.filter=""
eaccelerator.shm_max="0"
eaccelerator.shm_ttl="0"
eaccelerator.shm_prune_period="0"
eaccelerator.shm_only="0"
eaccelerator.compress="1"
eaccelerator.compress_level="9"
```

Opis poszczególnych opcji konfiguracyjnych znajduje się na stronie producenta: <http://www.eaccelerator.net/wiki/Settings>

Zapisujemy i wychodzimy z pliku.

W 3 linijce podałem ścieżkę, gdzie skompresowane pliki PHP będą umieszczone. Możesz podać tam dowolną ścieżkę na dysku – ważne, aby lokalizacja ta istniała i miała pełne prawa (`chmod 777`), ponieważ eAccelerator nie stworzy jej sobie sam.

Aby utworzyć odpowiedni katalog dla cachera wykonaj polecenia:


```
# mkdir -p /var/log/eaccelerator
# chmod 0777 /var/log/eaccelerator
```

Po tym wszystkim zrestartuj serwer Apache, aby nowa konfiguracja PHP została wczytana.

```
# /etc/init.d/apache2 restart
```

I odśwież poprzednio oglądaną stronę w przeglądarce pod adresem *phpinfo.php*.

Powinieneś zobaczyć różnice w stopce, oraz dodatkowe opcje konfiguracyjne dla cachera.

Stream Socket	
Transports	
Registered Stream Filters	string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, convert.iconv.*, bzip2.*, zlib.*
This program makes use of the Zend Scripting Language Engine: Zend Engine v2.2.0, Copyright (c) 1998-2006 Zend Technologies with eAccelerator v0.9.5.3, Copyright (c) 2004-2006 eAccelerator, by eAccelerator	
Powered By 	

Teraz możemy być pewni, że działa nasz eAccelerator.

Ciekawscy mogą jeszcze sprawdzić jakie katalogi stworzył sobie cacher w */var/log/eaccelerator/*. Powiniście ujrzeć 16 katalogów (0-9 oraz a-f).

Z ciekawostek napiszę, że eaccelerator zainstalowany u mnie na serwerze 5 lutego 2009 do dzisiaj wygenerował 1.4GB skompresowanych plików PHP.

UWAGA!!

Jeśli przyjdzie nam aktualizować PHP do nowszej wersji, to musicie się także przygotować do ponownej kompilacji eAcceleratora. Najlepiej wtedy wykonać wszystkie kroki w tym rozdziale od początku – usuwając stare pliki w */tmp*.