

System portalowy typu „przeciągnij i upuść”

Nie tylko portal

PHPNuke oferuje łatwą instalację i potężne możliwości dla każdego, kto chce szybko stworzyć własny portal. Wielość dodatków i rozszerzeń sprawia, że PHPNuke to świetny wybór dla firm i użytkowników prywatnych. W tym artykule pokażemy, jak dostosować ten system do własnych potrzeb oraz jak rozwiązywać najczęstsze problemy.

JAMES MOHR



Dla niektórych sama obecność w Internecie jest wystarczająca – kilka bardziej lub mniej udanych stron WWW składa się na ich internetowy wizerunek. Dla innych – także dla mnie – ma to inne znaczenie. Chcę posiadać profesjonalnie wyglądający serwis, z ciekawą zawartością – jest to do osiągnięcia, pod warunkiem użycia odpowiednich narzędzi.

W poszukiwaniu Świętego Graala

W moim miejscu pracy jestem uważany za specjalistę od skryptów. Wykorzystuję je do wykonywania prawie każdego zadania. Niezależnie, czy jest to skrypt powłoki czy Perl, zwykle potrafię wymyślić coś użytecznego. Mając jednak wybór pomiędzy kodowaniem a pisanie dokumentacji, preferuję tworzenie dokumentacji. To samo dotyczy prowadzonych

przeze mnie serwisów internetowych.

W moich serwisach internetowych często muszę wybierać pomiędzy chęcią dodania nowych funkcji a dodaniem nowej treści. Od kilku lat pracuję nad Linux Tutorial [1]. Pierwotnie składał się on z zestawu statycznych stron HTML – wiele z nich pochodzi z mojej książki „Linux User's Resource”. Podczas pracy nad witryną zrozumiałem, że prezentowanie statycznych stron to za mało i zacząłem dodawać kolejne funkcje, tworząc w ten sposób bardzo prosty, ale już modularny system do zarządzania treścią. Zawierał on słownik terminów, quiz linuxowsy, składający się z ponad 500 pytań i odpowiedzi oraz pewną liczbę innych opcji, mających ułatwić naukę Linuksa. W pewnym momencie więcej czasu poświęcałem pisaniu kodu i dodawaniu nowych funkcji niż dodawaniu zawartości. Dokładnie odwrotnie niż zamierzałem.

Chciałem zmienić mój serwis. Pewne opcje, które miałem zamiar udostępnić, zwiększyłyby znacząco funkcjonalność serwisu. Stworzenie z tego projektu Open Source i umieszczenie go w SourceForge spowodowało, że do mojego projektu dołączyło się kilku ochotników. Niestety, większość z nich zniknęła w ciągu paru tygodni, ze względu na tzw. „inne zobowiązania”. Nie napisali oni ani jednej linii kodu. Inni nieco pomogli, ale mimo to postęp prac był bardzo powolny. Musiałem samodzielnie napisać większość kodu. Brakowało mi wciąż wielu funkcji, co oznaczało, że muszę znaleźć system portalowy, który umożliwiłby w łatwy sposób integrację istniejącego materiału. Nie chciałem poświęcać zbyt wiele czasu na przeniesienie mojego kodu do nowego portalu. Chciałem zaoszczędzić czas, jaki musiałbym spędzić na programowaniu nowych funkcji.

Rozpocząłem zatem poszukiwania. Poza niezbędnymi funkcjami i łatwością przeniesienia mojego kodu, potrzebowałem odpowiedniego wsparcia technicznego na wypadek kłopotów. Niespodziewanie poszukiwania te nie trwały długo. Szybko natrafiłem na PHPNuke [2].

Dobry, zły i brzydki

Na wstępie zaniepokoił mnie fakt, który implikuje nazwa systemu. PHPNuke jest napisany w PHP. Poza odbytym kilka lat temu krótkim szkoleniem, nie miałem w tym zakresie prawie żadnego doświadczenia. Czy oznaczało to, że muszę poświęcić czas na naukę nowego języka programowania? Na szczęście odpowiedź brzmiała „Nie!” Mimo że PHP nie jest identyczny z Perl, ich składnie są bardzo do siebie podobne. Przeniesienie kodu okazało się zadziwiająco proste. Faktycznie wystarczyło kilka dni i moja witryna działała w nowym środowisku testowym.

Następnym krokiem było rozpoczęcie implementacji funkcji, na które oczekiwałem od kilku lat. W rzeczywistości, aby stworzyć użyteczny serwis w oparciu o PHPNuke, nie trzeba być dobrym programistą. Tak naprawdę wcale nie trzeba być programistą. PHPNuke dostarcza na tyle dużo funkcji, że każdy może znaleźć coś dla siebie. Nie ma potrzeby dodawania samemu czegokolwiek więcej.

Niektóre z dostępnych opcji to:

- zarządzanie użytkownikami (włączając w to płatne subskrypcje),
- forum,
- newsy użytkowników,
- ankiety,
- odnośniki dodawane przez użytkowników,
- pliki,
- biuletyn,

- encyklopedia,
- raporty,
- recenzje,
- statystyki,
- kontakt z twórcami (opinie użytkowników).

Opcje te są może ciekawe, ale często bezużyteczne, zwłaszcza jeśli nie wiesz, jak je zaimplementować. Na szczęście dokument PHP-Nuke_HOWTO, dostępny w witrynie PHPNuke [2], to kawał dobrej roboty. Opisuje wszystkie ważne opcje i zawiera wiele wskazówek i sztuczek. Miejscami temat jest potraktowany wnikliwie, choć mam wrażenie, że całość jest dość powierzchowna.

Brak szczegółowej dokumentacji wynika z braku komentarzy w kodzie źródłowym. Jedynie komentarze, jakie znalazłem dotychczas, to notki dotyczące praw autorskich na początku każdego pliku. Oznacza to, że jeśli chcesz zmodyfikować lub wykorzystywać istniejący kod, będziesz musiał wykonać dużo mrowczej pracy.

Jedyny element, który mi się nie podoba, to bezpieczeństwo systemu. PHPNuke nie ma opinii bezpiecznego systemu. Prawie każde oprogramowanie zawiera błędy związane z bezpieczeństwem, a PHPNuke przebyło długą drogę naprawiania błędów. Istnieje też szereg rozszerzeń, których zadaniem jest zwiększenie bezpieczeństwa systemu. Odnoszę jednak wrażenie, że twórcy aplikacji są bardziej zainteresowani jej funkcjonalnością niż bezpieczeństwem.

Kolejną sprawą, która mnie niepokoi, jest komentarz Francisco Burziego (dewelopera PHPNuke) w artykule „Clarifications on a possible rewrite of PHP-Nuke” dostępnym w serwisie phpnuke.org:

„Nowy kod będzie zamknięty. Oznacza to, że script kiddies prędzej obronią pracę magi-

sterską lub przeprowadzą przewód doktorski w dziedzinie informatyki, nim złamią nasz produkt. Spowoduje to znaczny spadek ilości problemów związanych z bezpieczeństwem naszej aplikacji.”

Przypomina mi to nieco podejście do bezpieczeństwa prezentowane przez firmę Microsoft. Zabezpieczanie przez ukrywanie (security by obscurity) nigdy nie było dobrym pomysłem. Ponadto „zamknięty” kod, o którym pisze Francisco Burzi wskazuje, że przyszłe wersje PHPNuke mogą nie być oprogramowaniem o otwartym źródle.

Szybkie spojrzenie do wnętrza

Najważniejszym dla mnie aspektem jest zarządzanie użytkownikami. W wielu serwisach część opcji jest dostępna tylko dla zarejestrowanych użytkowników, np wysyłanie postów na forum dyskusyjne jest możliwe dopiero po zarejestrowaniu się. PHPNuke implementuje to w bardzo logiczny i efektywny sposób. Użytkownicy wypełniają formularz z nazwą użytkownika, adresem e-mail oraz hasłem. Następnie PHPNuke natychmiast wysyła potwierdzający e-mail, zawierający odnośnik do strony, na której przed chwilą dokonaliśmy rejestracji. Klikając na ten odnośnik użytkownik aktywuje swoje konto.

Metoda ta jest istotna z dwóch powodów. Po pierwsze, zapobiega rejestracji bez wiedzy i zgody kogoś innego. Po drugie, jeżeli użytkownik naprawdę chce się zarejestrować, to musi tego dokonać w dwóch krokach. Postępowanie takie jest użyteczne dla właściciela witryny i jest określane jako podwójne upoważnienie. Jest to szczególnie ważne przy korzystaniu z funkcji biuletynu. Dzięki temu nie zostaniemy oskarżeni o spamowanie swoich użytkowników.

Oficjalnie PHPNuke może być instalowany w systemach Windows i Linux. Działa również w FreeBSD, OS/2 oraz nawet w Mac OS X. Podstawowe wymagania to posiadanie serwera WWW z PHP w wersji 4.2.x lub wyższej z modułem Apache o nazwie *mod_php*. W PHP-Nuke 5.3 została dodana warstwa abstrakcji SQL. Oznacza to, że PHPNuke powinno działać bez problemu z serwerami MySQL, mSQL, PostgreSQL, ODBC, ODBC_Adabas, Sybase oraz Interbase.

Wykorzystanie warstwy abstrakcji SQL powoduje, że wszystkie zapytania w tym języku zostają poddane standaryzacji przez warstwę abstrakcji SQL. Następnie zostaje wywoływana odpowiednia funkcja dla kon-

Ramka 1: Parametry pliku *config.php*

\$dbhost	Nazwa hosta, na którym uruchomiona jest baza danych. Pozostaw wpis „localhost”, jeśli znajduje się ona na tej samej maszynie co instalacja PHPNuke.
\$dbname	Nazwa użytkownika mającego dostęp do bazy danych. Jeśli konfigurujesz aplikację u swojego dostawcy Internetu, prawdopodobnie będziesz miał innego użytkownika niż root. Skonsultuj to ze swoim usługodawcą.
\$dbpass	Hasło użytkownika \$dbname.
\$dbname	Domyślnie jest to „nuke”, ale wielu dostawców Internetu nakazuje dodanie prefiksu podobnego do „NAZWAZUZYTKOWNIKA_nuke”. Ze względów bezpieczeństwa zawsze ją zmieniam.
\$prefix	Prefiks tabel bazy danych. Jego też zmieniam ze względu na bezpieczeństwo.
\$user_prefix	Powinien być taki sam jak \$prefix. Możesz mieć wiele serwerów PHPNuke korzystających z wspólnego użytkownika bazy danych. W tym przypadku \$user_prefix będzie taki sam dla każdego serwera.
\$dbtype	Ustala typ bazy danych. Dla MySQL musi to być „MySQL”.
\$sitekey	Jest to klucz witryny, wykorzystywany do generowania obrazków służących jako zabezpieczenie podczas logowania. Jego długość możesz dobrać do swoich wymagań.
\$gfx_chk	Określa, kiedy zabezpieczający obrazek z kluczem powinien być wyświetlany. Wartość 0 oznacza, że nigdy, 7 – zawsze. Kod ten jest podstawowym zabezpieczeniem przed automatycznym logowaniem, dlatego zawsze ustawiam 7 w systemach produkcyjnych.

kretniej bazy danych, czyli tej, z której korzystamy.

Strona PHPNuke jest podzielona na pięć obszarów: nagłówek, lewą kolumnę, zawartość, prawą kolumnę i stopkę. To, co pokazuje się w każdym z nich, jest zdefiniowane w module administracji systemu albo w poszczególnych modułach lub motywach. W lewej i prawej kolumnie widać różne bloki, które administrator może definiować jako osobne pliki PHP albo kod HTML. Zawartość nagłówka znajduje się zwykle wewnątrz motywu, a stopka jest zdefiniowana w Preferencjach.

Zawartość jest tworzona przez same moduły, będące głównymi komponentami PHPNuke. Każda wspomniana tutaj opcja jest zarządzana przez moduł. Dla każdego modułu istnieje podkatalog w katalogu module. Rzeczywista praca jest wykonywana przez plik index.php w odpowiednim podkatalogu modułu. Dzięki takiej zestandaryzowanej strukturze, aby dodać swój własny moduł, musisz stworzyć podkatalog i plik index.php.

Aktywowane moduły stają się widoczne domyślnie w bloku „Modules” po lewej stronie. Ich aktywność niekoniecznie oznacza, że są widoczne. W mojej witrynie znajduje się pewna liczba aktywnych modułów niewidocznych w bloku Modules. Mam do nich dostęp w następujący sposób:

```
modules.php?name=NAZWA_MODUŁU
```

gdzie NAZWA_MODUŁU jest nazwą modułu oraz nazwą katalogu, w którym moduł jest przechowywany.

Na mojej stronie wykorzystuję tę właściwość dla wielu opcji, których nie chcę dołączać do menu, np. znajduje się tam pewna liczba „specjalnych stron”, generowanych au-

tomatycznie. Strony te są tworzone przez osobne moduły. Muszą one pozostać dostępne, chociaż są niewidoczne dla użytkownika.

Nieaktywne moduły są wciąż dostępne dla administratora. Umożliwia to ich testowanie przed udostępnieniem online.

Dzięki temu, że moduły są głównymi komponentami PHPNuke, jak można się domyśleć, skrypt modules.php jest kluczem do tego wszystkiego. Pomimo to, module.php nie tworzy zwykle szkieletu strony i nie stanowi rzeczywistej zawartości. Jest natomiast odpowiedzialne za wywołanie właściwego modułu. Jeśli moduł nie może być znaleziony, jest nieaktywny lub niedostępny z innej przyczyny, zadaniem pliku module.php jest wyświetlenie odpowiedniego komunikatu.

Przygotowanie do uruchomienia

Pierwszym krokiem w instalacji jest oczywiście pobranie PHPNuke. W chwili pisania tego artykułu aktualną wersją jest 7.3, ale nie jest ona jeszcze dostępna publicznie. Pobranie jej wiąże się z koniecznością uiszczenia niewielkiej opłaty. Na potrzeby tego artykułu dokonaliśmy zakupu tej wersji.

Przed publicznym udostępnieniem (czytaj: darmowym) autorzy aplikacji zwykle wydają wersję, która jest dostępna wyłącznie po uiszczeniu niewielkiej opłaty. Jej wartość to około 10 dolarów lub 9,9 euro. Ponieważ uważam, że jest to bardzo dobry produkt, jestem skłonny wydać te pieniądze dla wspierania jego rozwoju – zwłaszcza, że w zamian otrzymuję przed innymi najnowszą wersję. Realizacja zamówienia trwa około godziny, po tym czasie można już pobrać odpowiedni plik.

Jeśli nie chcesz ponosić żadnych wydatków, możesz pobrać poprzednią wersję bezpośrednio ze strony: www.phpnuke.org, klika-

jąc opcję Download w głównym menu.

Pomimo swojej złożoności, PHPNuke jest łatwy w instalacji, zwłaszcza jeśli uruchamiasz go w systemie, w którym masz uprawnienia root-a, nie obawiasz się zbytnio o bezpieczeństwo i chcesz wykorzystać domyślną konfigurację. Możesz zainstalować i uruchomić PHPNuke dosłownie w 5 minut. Ponieważ nie odpowiada mi domyślna konfiguracja, instalacja systemu zabrała mi nieco więcej czasu.

W zależności od wersji pakietu PHPNuke, plik Install.txt może zawierać błędny opis instalacji. Jest tam napisane, aby rozpakować pakiet do głównego katalogu serwera WWW. Niestety tak nie jest w przypadku wersji 7.0, którą wcześniej pobrałem z SourceForge – pakiet w rzeczywistości został rozpakowany w podkatalogu o nazwie PHP-Nuke-7.0 i należało samodzielnie przenieść jego zawartość do głównego katalogu serwera WWW.

W przypadku wersji 7.3, pliki są rozpakowywane bezpośrednio do właściwego katalogu, bez tworzenia podkatalogu PHP-Nuke-7.3. Oprócz różnych plików.txt, są w nim trzy podkatalogi: sql, który zawiera skrypt SQL służący do stworzenia potrzebnych tabel (nuke.sql); upgrades, zawierających skrypty potrzebne do aktualizacji między wersjami; oraz html, który jest głównym katalogiem PHPNuke.

Po rozpakowaniu plików należy otworzyć w edytorze tekstowym plik config.php, znajdujący się w głównym katalogu pakietu. W pliku pliku tym należy dokonać szeregu zmian, aby dopasować go do lokalnych wymagań. Jako dodatkowe zabezpieczenie polecam przeniesienie w bezpieczne miejsce skryptu config.php z głównego katalogu. Zamiast tego można utworzyć plik config.php w głównym katalogu serwera WWW, wpisując do niego lokalizację prawdziwego pliku config.php, np.:

Prenumerata Linux Magazine Nie przegap takiej okazji

- Zamawiając prenumeratę oszczędzasz!
- Płacisz jak za 9 numerów, a otrzymujesz 12!
- Z każdym numerem DVD lub płyta CD-ROM.

Najszybszy sposób zamówienia prenumeraty:

<http://www.linux-magazine.pl>

Infolinia: 0801-800-105



```
<?php include('../config.php');?>
```

Sugeruję stworzenie specjalnego użytkownika wraz z jego hasłem, którego jedynym celem jest dostęp do bazy danych. Domyślnie zmiana \$dbname jest ustawiona na root. Nie jest to dobry pomysł, gdyż praca z uprawnieniami root-a jest zawsze groźna.

Na moim komputerze mam zainstalowanych kilka serwerów WWW, włącznie z serwerem testowym Linux Tutorial. Ponieważ chcę mieć odseparowane instalacje PHPNuke działające na wielu serwerach, musiałem stworzyć wiele baz danych. Oznaczało to zmianę wielu plików config.php i nuke.sql, wybrałem następujące wartości:

```
$dbname = „jimmonuke“;
$prefix = „jimmonuke“;
$user_prefix = „jimmonuke“;
```

Krótkie objaśnienie każdego z pól zawiera Ramka 1.

Następnie w pliku nuke.sql muszę zmienić wszystkie wyrażenia zawierające nuke_ na jimmonuke_ (uwaga na znak podkreślenia), gdyż wybrałem niestandardowe nazwy tabel.

Teraz należy stworzyć bazę danych. Ponieważ zdefiniowałem swoją bazę danych o nazwie innej od domyślnej, nie mogę wykorzystać przykładu zwartego w pliku Install.txt. Bazę tworzy się w następujący sposób:

```
mysqladmin create jimmonuke
```

W zależności od konfiguracji systemu, być może trzeba będzie użyć opcji -p, która wyświetli monit o hasło. Oczywiście użytkownik ten musi mieć uprawnienia do tworzenia nowych baz danych. W moim przypadku stworzyłem bazę danych z poziomu konta root i nadałem mojemu użytkownikowi uprawnienia tylko do bazy jimmonuke. W tym celu użyłem następującego zapytania SQL:

```
GRANT ALL ON jimmonuke.* to
jimhome identified
by „HASŁO“
```

Następnie tworzymy tabele w bazie danych, wykorzystując skrypt nuke.sql:

```
mysql jimmonuke < nuke.sql
```

Kiedy łączymy się z serwisem opartym na PHPNuke, zwykle otwieramy plik index.php w głównym katalogu – w tym celu należy skonfigurować opcję DirectoryIndex swojego

serwera WWW. Powinna ona wskazywać na index.php, tak by ten plik był ładowany w miejsce standardowego index.html. Wiele serwerów WWW (np. Apache) ma skonfigurowane oba pliki jako domyślne, dlatego zazwyczaj nie jest to problem.

Start!

Jeżeli konfiguracja została wykonana prawidłowo, system jest już gotowy do uruchomienia. Administrację systemem PHPNuke rozpoczynamy od adresu:

```
http://nazwa.twojej.witryny/
admin.php
```

Zazwyczaj nie umieszczam PHPNuke w głównym katalogu serwera WWW, zamiast tego zmieniłem nazwę domyślnego katalogu z html na nuke i skopiowałem jego zawartość do głównego katalogu serwera. Dzięki temu mam dostęp do strony administracyjnej w następujący sposób:

```
http://nazwa.twojej.witryny/
/nuke/admin.php
```

Kiedy otwieramy stronę administracyjną po raz pierwszy, jesteśmy proszeni o stworzenie konta administratora. Można też utworzyć zwykłego użytkownika z takimi samymi prawami jak administrator. Osobiście preferuję takie rozwiązanie z kilku powodów.

Po pierwsze, mogę zalogować się do systemu na to konto, by stwierdzić, jak zachowuje się i wygląda system wykorzystywany przez zwykłych użytkowników. Po drugie, jest wielce prawdopodobne, że będę chciał wysłać wiadomości, informacje lub cokolwiek innego, a wówczas nazwa użytkownika mnie nie zdradzi. Gdybym użył konta o nazwie „admin”, użytkownicy natychmiast znalazłyby nazwę konta administratora. Pozornie nie jest to problemem, ale im mniej inni wiedzą o strukturze serwisu, tym lepiej.

Po tym, jak utworzymy konto administratora, zostaniemy przekierowani na stronę administracyjną. Tutaj można skonfigurować podstawowe zachowanie systemu PHPNuke, a także określić, które moduły i bloki mają być wyświetlane.

Konfiguracji podstawowego systemu dokonujemy poprzez kliknięcie przycisku „Preferences”. Tutaj można zdefiniować nazwę serwisu, stopkę, różne wartości domyślne itd. Warto to zrobić na początku.

Dla zapisania zmian należy wcisnąć znajdujący się na dole strony przycisk „Save

Changes”. Teraz można przejść do zarządzania (przycisk „Modules”), które zostało wstępnie skonfigurowane przez PHPNuke. Można tutaj na przykład określić, które moduły mają być aktywne, kto może je widzieć itp. Zauważmy, że prawie połowa modułów jest domyślnie skonfigurowana jako „nieaktywne”, co oznacza, że taki moduł nie pokaże się po lewej stronie w menu portalu.

W systemach produkcyjnych najlepiej usunąć wszystkie nieużywane moduły. Mogą one bowiem zawierać luki w bezpieczeństwie, o których nie wiemy, dając w ten sposób możliwość włamania do systemu. Teoretycznie domyślne moduły nie mogą być uruchamiane bezpośrednio, a skrypt modules.php zapobiega uruchamianiu nieaktywnych modułów, ale lepiej być nieufnym i usunąć je z systemu. Podobnie nowe moduły należy najpierw gruntownie przetestować w środowisku testowym.

Konfigurując system należy pamiętać, że jeden z modułów jest zawsze podświetlony, domyślnie jest to moduł News. Podświetlenie określa, który z modułów jest widoczny na stronie głównej.

Blokujemy

Jak wspominałem na początku, PHPNuke nie ma dobrej reputacji, jeśli chodzi o bezpieczeństwo. Nie oznacza to jednak, że jest to system dziurawy. Tym niemniej należy włożyć nieco wysiłku, aby system był dobrze zabezpieczony.

W tym celu najpierw należy sprawdzić listy dyskusyjne związane z bezpieczeństwem, dostępne w serwisie NukeCops [3]. Znajdują się tam dyskusje poświęcone określonym problemom PHPNuke (i modułów) i sposobom ich usuwania. Warto też sprawdzić serwis NukeFixes [4], znajdują się tam specjalnie przygotowane wersje (począwszy od wersji 6.0) PHPNuke z nałożonymi łatkami zwiększającymi bezpieczeństwo. ■

INFO

[1] System autora oparty o PHPNuke:

<http://www.linux-tutorial.info>

[2] PHPNuke: <http://www.phpnuke.org>

[3] NukeCops: <http://www.nukecops.com>

[4] NukeFixes: <http://www.nukefixes.com>

[5] PHPNuke forum:

<http://www.nukeforums.com>

[6] Zasoby PHPNuke:

<http://www.nukeresources.com>