

Administracja Systemami Komputerowymi
Komercyjny serwer FTP oparty o ProFTPD

Leszek Bizoń <*bizonl@student.uci.agh.edu.pl*>
Michał Ciesielski <*pinkey@ds5.agh.edu.pl*>
Mariusz Dziewierz <*aristot@student.uci.agh.edu.pl*>

29 maja 2002

Streszczenie

Referat przedstawia wybrane zagadnienia związane z utrzymanie komercyjnego serwera FTP. Na bazie darmowej implementacji demona FTP, *ProFTPD* w wersji 1.2.4, został opisany sposób konfiguracji usługi umożliwiający tworzenie serwerów wirtualnych FTP z własnymi pulami użytkowników i możliwością uaktywnienia konta anonimowego dostępu. Treść referatu obejmuje również zagadnienia analizy logów pochodzących z demona *ProFTPD*.

Spis treści

1	Wprowadzenie	4
2	Instalacja demona <i>ProFTPD</i>	5
2.1	Określenie wymagań stawianych serwerowi wirtualnemu	5
2.2	Przygotowanie systemu operacyjnego	6
2.3	Kompilacja i uruchomienie demona <i>ProFTPD</i>	8
3	Konfiguracja wirtualnego serwera FTP	9
3.1	Wyłączenie głównego serwera FTP	11
3.2	Ustawienie sposobu bindowania gniazdek	11
3.3	Uaktywnienie i identyfikacja serwera wirtualnego	11
3.4	Ustawienia autentykacji użytkowników	12
3.5	Uwięzienie użytkownika w jego katalogu domowym	13
3.6	Ukrywanie nazw użytkownika i grupy	14
3.7	Opcje dostępu do plików i katalogów	15
3.8	Ustawienie dostępu do komend serwera FTP	16
3.9	Konfiguracja ograniczeń przestrzeni dyskowej	17
3.10	Ograniczenie liczby dostępnych połączeń	18
3.11	Konfiguracja sposobu logowania pracy serwera	18
4	Generowanie statystyk dostępu do serwera	20
4.1	Przygotowanie formatu logowania	20
4.2	Generowanie statystyki	20
5	Podsumowanie	22
A	Plik konfiguracyjny serwera wirtualnego	24
B	Skrypt do generowania statystyk dostępu do serwera	27

1 Wprowadzenie

Jedną z wielu usług świadczonych przez firmy specjalizujące się w „hostingu” jest udostępnianie swym klientom wirtualnych serwerów FTP. Rozwiązanie takie polega na uruchomieniu na jednym komputerze, spełniającym rolę serwera, odpowiednio skonfigurowanego demona FTP, który potrafi obsłużyć sesje transferu plików w sposób zależny od tego, na jaki adres IP są one skierowane. Każdy z adresów, pod którym widoczny jest serwer, jest powiązany z jednym klientem, który w ten sposób otrzymuje do swej dyspozycji w pełni funkcjonalny serwer FTP. Może on służyć do udostępniania plików poprzez konto anonimowego dostępu, bądź też nawet umożliwiać tworzenie kont użytkowników wewnątrz serwera wirtualnego.

Stosowanie rozwiązań opartych o serwery wirtualne niesie wiele korzyści zarówno dla klientów, jak i dla samych dostawców. Dla jednostki, która wykupiła wirtualny serwer FTP, szczególnie atrakcyjne cechy tej usługi to:

- Dostawca dysponuje najczęściej łączami do sieci Internet o przepustowości przewyższającej tę, która jest dostępna klientowi; klient nie musi mieć permanentnego dostępu do sieci.
- Dostawca stosuje odpowiednie polityki mające na celu zabezpieczenie przechowywanych danych na udostępnianych serwerach przed ich utratą; klient może nie dysponować środkami, aby zabezpieczyć swe dane.
- Serwery dostawców zarządzane są przez ekspertów, którzy są uczuleni na zagrożenia bezpieczeństwa; nie każdą firmę stać na to, aby zatrudnić dobrego administratora.

Dla samego dostawcy usług, główną zaletą korzystania z rozwiązań opartych o serwery wirtualne, jest możliwość obsługi wielu klientów przy pomocy zredukowanej ilości zasobów – serwerów, co w połączeniu z uproszczoną konfiguracją znacznie obniża koszty świadczenia usług.

Obsługa wielu klientów przy pomocy jednego serwera wiąże się z nałożeniem pewnych ograniczeń na udostępniane wirtualne serwery FTP. Muszą one współdzielić zasoby na jednej maszynie, przy czym nie może wiązać się to ze znaczącą degradacją jakości świadczonych usług, związaną z przeciążeniem serwera. Przykładowo konieczne jest ograniczenie przestrzeni dyskowej dostępnej dla poszczególnych klientów, czasem niezbędne może być limitowanie dostępnej przepustowości, czy też ograniczenie ilości kont użytkowników, które można założyć w obrębie jednego serwera wirtualnego. Te i inne parametry stanowią podstawę do negocjacji z klientem kosztów utrzymania wirtualnego serwera FTP przez dostawcę.

Jakość usług świadczonych przez dostawcę serwerów wirtualnych FTP, zależy w głównej mierze od użytego demona FTP i od jego umiejętnej konfiguracji. W swoim referacie opieramy się na programie *ProFTPD* w jego ostatniej, stabilnej wersji 1.2.4. Jest to narzędzie o bardzo rozbudowanych możliwościach

konfiguracyjnych, przy czym skrypty konfiguracyjne są bardzo przejrzyste i przypominają swą strukturą te pochodzące z popularnego demona http *Apache*. Choć w przeszłości wykrywano w nim luki w zabezpieczeniach, obecnie uważany jest za bezpieczny i odpowiedni do zastosowań komercyjnych.

W dalszych sekcjach referatu skupimy się na opisie sposobu instalacji i zarządzania demonem *ProFTPD* (sekcja 2), na sposobie jego zaawansowanej konfiguracji (sekcja 3) oraz na pewnych metodach analizy logów generowanych podczas pracy serwera FTP (sekcja 4). Ograniczamy się jedynie do pewnych elementów instalacji rzeczywistego serwera wirtualnego FTP, który może być wykorzystywany w komercyjnych zastosowaniach; w żadnym wypadku naszym zamiarem nie jest pisanie podręcznika, według którego firma mogłaby rozpocząć działalność „hostingową”.

2 Instalacja demona *ProFTPD*

Instalacja i konfiguracja wirtualnego serwera FTP wymaga starannego określenia stawianych przed nim wymagań, gdyż niektóre z nich mogą mieć wpływ np. na sam proces kompilacji demona *ProFTPD*. W swym referacie proponujemy szereg ograniczeń, jakie według nas powinien spełniać typowy wirtualny serwer FTP, aby potem przedstawić kolejne kroki mające na celu sprostanie owym wymaganiom. Mamy tu na myśli konfigurację interfejsów sieciowych, przygotowanie systemu plików, kompilację i konfigurację demona *ProFTPD* oraz samo uruchomienie demona.

2.1 Określenie wymagań stawianych serwerowi wirtualnemu

Zakres usług świadczonych przez firmy zajmujące się utrzymywaniem wirtualnych serwerów FTP może być bardzo zróżnicowany. Przedstawione poniżej wymagania, staraliśmy się dobrać tak, aby istniała możliwość ich łatwej realizacji przy pomocy demona *ProFTPD*:

1. Serwer FTP obsługuje wiele serwerów wirtualnych, pracujących na różnych adresach IP i standardowym porcie 21.
2. Każdy z serwerów wirtualnych posiada własną pulę użytkowników, którzy posiadają własną, ale ograniczoną objętościowo, przestrzeń dyskową.
3. Na każdym z serwerów wirtualnych istnieje możliwość uaktywnienia konta anonimowego dostępu. Do umieszczania plików wewnątrz struktury plików dostępnych publicznie służy osobne konto użytkownika.
4. Użytkownicy korzystający z serwera FTP są ograniczeni dostępem jedynie do swojego katalogu domowego. Konto anonimowego dostępu również uniemożliwia wyjście poza swój katalog bazowy.

5. Sumaryczna objętość plików przechowywanych na jednym serwerze wirtualnym jest z góry ograniczona mechanizmem systemowym *quoty*.
6. Dodanie użytkownika do serwera wirtualnego nie wiąże się z tworzeniem nowego konta użytkownika w systemie. Każdy serwer wirtualny działa w oparciu o jednego użytkownika systemowego, którzy razem tworzą jedną grupę.
7. Każdy z wirtualnych serwerów FTP posiada własne logi dostępu do plików oraz transferów plików. Logi dotyczące autoryzacji są wspólne dla wszystkich serwerów, w celu ułatwienia przygotowania statystyk dostępu do nich.

2.2 Przygotowanie systemu operacyjnego

Maszyna, na której ma zostać uruchomiony serwer FTP obsługujący konta wirtualne wymaga specjalnych przygotowań. Należy ją skonfigurować tak, aby była dostępna poprzez kilka adresów IP, które odpowiadają adresom różnych serwerów wirtualnych. Konieczne jest również dodanie użytkowników systemowych odpowiadających różnym serwerom wirtualnym. Użytkownicy Ci powinni należeć do jednej grupy systemowej. Ostatni etap przygotowań to założenie odpowiedniej struktury katalogów, w której będą składowane pliki.

Publicznie dostępne serwery FTP powinny być osiągalne na standardowym porcie 21. Specyfika protokołu FTP sprawia, iż każdy z serwerów wirtualnych, jeśli ma być widoczny na owym porcie, musi posiadać własny adres IP, który pozwala demonowi FTP rozróżnić poszczególne, zarządzane przez niego serwery wirtualne. Jeden serwer FTP może obsługiwać setki serwerów wirtualnych, jednakże konieczne jest skonfigurowanie systemu tak, aby był on widoczny pod różnymi adresami IP, co w dosyć prosty sposób może być osiągnięte poprzez skonfigurowanie wymaganej ilości *aliasów* dla kart sieciowych. Rozwiązanie tego zagadnienia jest zależne od systemu operacyjnego, dlatego też nie będziemy starali się go dokładnie objaśniać. Ważne jest, aby w wyniku podjętych działań system posiadał kilka adresów IP oraz kilka nazw powiązanych z tymi adresami. W naszej przykładowej konfiguracji, kolejne serwery wirtualne noszą nazwy: *virtual.net*, *virtua2.net*, ..., *virtuan.net*.

Kolejnym krokiem niezbędnym do uruchomienia serwera FTP, jest utworzenie w systemie użytkowników powiązanych z poszczególnymi serwerami wirtualnymi. Wykonanie tej czynności jest uzależnione od systemu operacyjnego. Wynikiem tej operacji są odpowiednie wpisy w plikach */etc/passwd* i */etc/shadow*. Nowi użytkownicy nie muszą posiadać możliwości logowania się do systemu przez *shell*, a wręcz jest to niewskazane (przykładowo powłoka dla tych użytkowników może być ustawiona na */bin/false*, a PAM skonfigurowany tak, aby wymagał do autoryzacji użytkownika poprawnej powłoki, tzn. takiej, która jest wymieniona w pliku *etc/shells*). Katalog domowy przypisany danemu użytkownikowi nie ma żadnego znaczenia (może wskazywać miejsce, w którym znajduje się przestrzeń dyskowa

account	Nazwa konta (<i>login</i>)
password	Zakodowane hasło dostępu do konta
UID	Identyfikator konta
GID	Identyfikator grupy, do której należy konto
GECOS	Opisowa nazwa konta
directory	Katalog domowy użytkownika
shell	Powłoka do sesji interaktywnej

Tablica 1: Znaczenie pól w rekordzie pliku *passwd*.

```
virtual:pEyOMH22E9IZY:504:1008:Anonymous admin:/home/virtual/pub:/bin/false
user1:RxKbkKr3vSlXQ:504:1008:Virtual user1:/home/virtual/user1:/bin/false
user2:tsRIgLaGptD/g:504:1008:Virtual user2:/home/virtual/user2:/bin/false
user3:x5NksOileS.jw:504:1008:Virtual user3:/home/virtual/user3:/bin/false
user4:vhQk8uYIYhmII:504:1008:Virtual user4:/home/virtual/user4:/bin/false
```

Rysunek 1: Przykładowy plik *passwd* serwera wirtualnego FTP.

na dany serwer wirtualny, ale nie musi!). W naszym referacie zakładamy, iż z kolejnymi serwerami wirtualnymi powiązani są użytkownicy o nazwach: *virtual*, *virtua2*, ..., *virtuan*. Owi użytkownicy powiązani są z jedną grupą o nazwie *virtftp*, którą trzeba wcześniej utworzyć.

Istotnym etapem jest przygotowanie struktury katalogów, w której mieścić się będą wirtualne serwery FTP. Każdy serwer wirtualny posiada własny katalog, wewnątrz którego mieszczą się podkatalogi: *etc* – przechowuje plik *passwd* dla danego serwera wirtualnego, *pub* – zawiera strukturę konta anonimowego dostępu do serwera wirtualnego oraz katalogi użytkowników. Właścicielem wszystkich plików i katalogów jest użytkownik powiązany z danym serwerem wirtualnym. Wygodnym rozwiązaniem jest umieszczenie katalogów wszystkich serwerów wirtualnych w jednym miejscu. W tym referacie przyjmujemy, iż katalogi wszystkich serwerów wirtualnych znajdują się w katalogu */home* i mają nazwy: *virtual*, *virtua2*, ..., *virtuan*.

Z każdym z serwerów wirtualnych związany jest plik z hasłami o strukturze, która odpowiada plikowi */etc/passwd*. Zawiera on rekordy o formacie przedstawionej poniżej (znaczenie poszczególnych pól zostało zebrane w tablicy 1).

```
account:password:UID:GID:GECOS:directory:shell
```

Rysunek 1 przedstawia przykładowy plik *passwd* znajdujący się w katalogu *etc* wewnątrz struktury katalogów serwera wirtualnego. Pierwszy rekord w przedstawionym przykładzie ma szczególne znaczenie, gdyż reprezentuje on konto umoż-

liwiający dostęp do struktury konta anonimowego dostępu. Użytkownik *virtual* używa go w celu umieszczenia plików na koncie publicznym. Pozostałe rekordy reprezentują kolejnych użytkowników pojedynczego serwera wirtualnego; pole *directory* wskazuje na ich katalogi domowe. Warto zwrócić uwagę, iż pola *UID* i *GID* są ustawione dla wszystkich użytkowników na jednakowe wartości i odpowiadają odpowiednio identyfikatorowi użytkownika *virtual*, który można znaleźć w pliku */etc/passwd* oraz identyfikatorowi grupy *virtftp*, który jest do odczytania z pliku */etc/group*. Ustawienie pola *shell* nie ma żadnego znaczenia. Zakodowane hasła zostały uzyskane przy pomocy systemowej funkcji *crypt*.

Ostatnim etapem przygotowań systemu operacyjnego do tego, aby pracował on jako wirtualny serwer FTP, jest nałożenie ograniczeń na objętość plików przechowywanych na poszczególnych serwerach FTP. Mechanizm *quoty* udostępniony w demonie *ProFTPD* przez moduł *mod_quota*, ze względu na swój sposób działania, nie może być zastosowany do bezwzględnego ograniczenia pojemności dyskowej zajmowanej przez serwer wirtualny, dlatego też należy skorzystać z narzędzi udostępnianych w tym celu przez system operacyjny. Nałożenie limitu na serwer wirtualny sprowadza się do nałożenia limitu na objętość plików przechowywanych przez użytkownika, który związany jest z danym serwerem. Moduł *mod_quota* może zostać użyty do limitowania przestrzeni dyskowej zajmowanej przez użytkowników wewnątrz serwera wirtualnego. Konieczne jest umieszczenie w katalogach domowych użytkowników plików *.quota*, które zawierają dwie liczby: *<maksymalna_zajętość>* i *<aktualna_zajętość>*. W przypadku, gdy pliki te nie będą dostępne, użytkownik nie będzie mógł zapisać na swoim koncie żadnych danych.

2.3 Kompilacja i uruchomienie demona *ProFTPD*

Strona WWW projektu *ProFTPD* znajduje się pod adresem <http://www.proftpd.org/>. Pod adresem <http://www.proftpd.org/downloads.html> można znaleźć adresy lustrzanych serwerów FTP, na których umieszczono aktualną dystrybucję demona *ProFTPD* w postaci archiwum (*tarbal*) z kodem źródłowym (w referacie opieramy się na wersji 1.2.4). Sposób kompilacji i instalacji programu jest przedstawiony w wyczerpujący sposób w pliku *INSTALL* znajdującym się w głównym katalogu archiwum, dlatego też ograniczymy się jedynie do kilku wskazówek dotyczących opisywanego przez nas rozwiązania.

Kompilacja i instalacja może zostać przeprowadzona poprzez wydanie ciągu komend:

```
# ./configure --prefix=/usr/local/ --with-modules=mod_quota
# make
# make install
```

O ile kompilacja i instalacja przebiegną na danej architekturze bez żadnych zakłóceń, w katalogu wskazanym przez opcję *-prefix* zostanie utworzona struktura katalogów z wszystkimi plikami niezbędnymi do uruchomienia serwera FTP. Sam

demon FTP znajdzie się w tym konkretnym przypadku w `/usr/local/bin/proftpd`, a jego przykładowy plik konfiguracyjny w `/usr/local/etc/proftpd.conf`. Bazową konfigurację należy zastąpić przez skrypt konfiguracyjny, którego szkic znajduje się w dodatku A. Wszystkie opcje zastosowane w podanym pliku konfiguracyjnym, zostaną dokładnie omówione w sekcji 3.

Uruchomienie demona *ProFTPD* odbywa się poprzez wydanie komendy:

```
# /usr/local/bin/proftpd
```

Podany plik konfiguracyjny uruchamia demon w trybie *standalone*, przez co nie wymaga on konfiguracji *inetd*. Zalecanym rozwiązaniem jest dodanie podanej wcześniej komendy do skryptów startowych systemu. Podczas testowania serwera FTP można skorzystać z dobrodziejstw opcji: *-c*, *-d*, *-t* i *-n*. Komenda:

```
# /usr/local/bin/proftpd -t -c <config>
```

Powoduje sprawdzenie poprawności syntaktycznej pliku konfiguracyjnego *<config>*, natomiast komenda:

```
# /usr/local/bin/proftpd -n -d5
```

uruchamia serwer FTP z rozszerzonym raportem o jego pracy (*debug*), który jest wypisywany na standardowe wyjście, tym samym ułatwiając szybką analizę błędów popełnionych przy konfiguracji serwera.

3 Konfiguracja wirtualnego serwera FTP

Przy starcie demon *ProFTPD* wczytuje plik konfiguracyjny, którego lokalizacja podana jest przy pomocy opcji *-c*, bądź w przypadku niewyspecyfikowania tej opcji, plik ten jest wyszukiwany w katalogu `$PREFIX/etc/proftpd.conf`, gdzie *\$PREFIX* jest katalogiem bazowym podawanym przy kompilacji programu. Skrypt konfiguracyjny jest strukturalizowany o 6 możliwych kontekstach: *server config*, *<Global>*, *<Anonymous>*, *<Directory>*, *<Limit>*, *<VirtualHost>*. Możliwe jest również umieszczanie plików konfiguracyjnych w katalogach, których mają one dotyczyć, w plikach *.ftpassess*; odpowiadają one funkcjonalnie kontekstowi *<Directory>*. Poszczególne konteksty konfiguracyjne mogą się w sobie zagnieżdżać. Ich znaczenie zostało objaśnione w tabeli 2.

Podczas pisania skryptu konfiguracyjnego staraliśmy się, aby mógł on się stać podstawą budowy plików konfiguracyjnych dla rzeczywistych serwerów wirtualnych FTP, poprzez proste zwielokrotnienie jego niektórych sekcji. W kontekście głównego serwera (*server config*) zostały umieszczone dyrektywy, które mają na celu ustawienie pewnych globalnych parametrów serwera FTP, takich jak np. ilość dostępnych połączeń. Istotne jest, iż zdecydowaliśmy się na wyłączenie domyślnego serwera przy pomocy opcji *Port 0*. Postąpiliśmy tak, aby każdy z serwerów wirtualnych był konfigurowany w ten sam sposób w sekcjach *<VirtualHost>*. Oczywiście możliwe jest uaktywnienie tego serwera, przez podanie właściwego

server config	Główny kontekst serwera, który zawiera wszystkie opcje, które nie mają związku z innymi typami kontekstów, bądź dotyczą bazowego serwera FTP.
<Global>	Dyrektywy konfiguracyjne umieszczone w tym kontekście służą do ustawienia globalnych parametrów serwerów FTP, które następnie są uszczegóławiane w sekcjach <i>server config</i> i <i><VirtualHost></i> .
<Anonymous>	Sekcja ta służy do konfiguracji kont dostępu anonimowego do serwera FTP. Ten kontekst nie tworzy nowego serwera FTP, lecz raczej opisuje dodatkową możliwość dostępu przez użytkownika typu <i>anonymous</i> .
<Directory>	Dyrektywy występujące w tym kontekście pozwalają na określenie nowych wymagań w dostępie do określonych katalogów na serwerze. Zastosowanie tego kontekstu do konfiguracji umożliwi bardzo elastyczne ustawienie uprawnień w dostępie do plików, które wykracza znacznie poza możliwości oferowane w tej materii przez zwykły system plików.
<Limit>	Sekcja ta ma zastosowanie w nakładaniu ograniczeń w dostępie do komend protokołu FTP, bądź też całych ich grup.
<VirtualHost>	Ten kontekst stanowi podstawę budowy wirtualnych serwerów FTP. Umożliwia utworzenie dodatkowej możliwości dostępu do serwera FTP, który nasłuchuje żądań połączenia na adresie i/lub porcie odmiennym od serwera bazowego.

Tablica 2: Konteksty konfiguracji serwera *ProFTPD*.

numeru portu, ale należy być ostrożnym, gdyż będą się do niego odnosić również opcje z sekcji *<Global>*. Kontekst *<Global>* zawiera w sobie dyrektywy, które są wspólne dla wszystkich serwerów wirtualnych, natomiast następujące po nim sekcje *<VirtualHost>* zawierają dyrektywy specyficzne dla każdego z serwerów. W sekcji tej skonfigurowane jest również konto anonimowego dostępu, które może zostać zdezaktywowane poprzez skasowanie odpowiedniego fragmentu pliku konfiguracyjnego lub też zmianę praw dostępu do tego konta w sekcji *<Limit>*. Skrypt konfiguracyjny, który podajemy, został tak pomyślany, aby dodanie nowych serwerów wirtualnych wymagało jedynie zwielokrotnienie i edycję sekcji *<VirtualHost>*, a jego prosta konstrukcja umożliwia wygenerowanie go przy pomocy nieskomplikowanego skryptu, napisanego w jednym z wielu języków skryptowych.

W kolejnych sekcjach zostaną opisane wszystkie dyrektywy wykorzystane

przy konfiguracji serwerów wirtualnych, ze szczególnym zwróceniem uwagi na te, które nie pojawiają się w domyślnej konfiguracji serwera lub też są wykorzystane w odmienny sposób.

3.1 Wyłączenie głównego serwera FTP

Przy standardowej konfiguracji demona *ProFTPD* uruchamiany jest tzw. serwer główny związany z domyślnym interfejsem sieciowym oraz serwery wirtualne, których adresy określone są jako parametr dyrektywy rozpoczynającej ich bloki konfiguracyjne (*<VirtualHost A.B.C.D>*). Odpowiednie skonfigurowanie serwera głównego umożliwia uruchomienie dodatkowego serwera wirtualnego, lecz wprowadza bałagan do skryptu konfiguracyjnego, dlatego też zdecydowaliśmy się na jego wyłączenie przy pomocy opcji:

```
Port 0
```

Dyrektywa *Port*, o domyślnym parametrze 21, określa port, na którym serwer FTP (główny, bądź wirtualny) nasłuchuje na połączenia od klientów. Nie ma potrzeby umieszczania tej opcji w sekcjach konfiguracji serwerów wirtualnych, gdyż jej domyślne ustawienie gwarantuje ich poprawne działanie. W analogiczny do podanego powyżej sposobu, omawiana opcja może służyć do łatwego blokowania dostępu do wybranego serwera wirtualnego.

3.2 Ustawienie sposobu bindowania gniazdek

Do określenia sposobu, w jaki demon *ProFTPD* binduje gniazdko, na których przyjmuje połączenia od klientów, służy dyrektywa *SocketBindTight*.

```
SocketBindTight on
```

Wariant, który został wybrany w naszym skrypcie konfiguracyjnym powoduje, że dla każdego serwera wirtualnego tworzone jest nowe gniazdko, z własnym deskryptorem pliku, co może stanowić problem przy dużej ilości serwerów wirtualnych. Druga możliwość, gdy parametr opcji *SocketBindTight* ustawiony jest na *off* powoduje otwarcie jednego gniazdko, które nasłuchuje na wszystkich adresach (gniazdko tworzone jest z adresem IP 0.0.0.0). Niestety wariant ten nie chciał działać w przypadku naszego skryptu konfiguracyjnego, a krótka sesja z *strace* wykazała, że problem występuje w momencie wywołania funkcji systemowej *listen(2)*.

3.3 Uaktywnienie i identyfikacja serwera wirtualnego

Każdemu z serwerów wirtualnych obsługiwanych przez demon *ProFTPD*, odpowiada sekcja typu *<VirtualHost>*, w której umieszczone są jego opcje konfiguracji, które nie znalazły dla siebie miejsca w kontekście *<Global>*. Adres serwera wirtualnego jest określony przez jedyny parametr dyrektywy *<VirtualHost>* rozpoczynającej jego blok konfiguracyjny. Przykładowo dla adresu *virtual1.net*, będzie to:

```

<VirtualHost virtual.net>
    ...
    ServerName      "Virtual FTP server"
    ServerAdmin     "ftp@virtual.net"
    DeferWelcome    on
    ServerIdent     on "FTP Server ready"
    ...
</VirtualHost>

```

Adres serwera wirtualnego może być również określony przez podanie jego adresu IP. W przypadku, gdy zaistnieją problemy z zamianą adresu na jego adres IP (np. w wyniku nieprawidłowego skonfigurowania usługi DNS), demon *ProFTPD* nie uruchomi się.

Podany wcześniej przykład udostępnia szereg opcji służących do określenia identyfikacji serwera wirtualnego. Dyrektywa *ServerName* określa jego nazwę, natomiast *ServerAdmin* specyfikuje kontaktowy adres e-mail do administratora danego serwera. Zaproponowane przez nas opcje *DeferWelcome* i *ServerIdent* mają na celu ograniczenie możliwości identyfikacji oprogramowania, w oparciu o które funkcjonuje serwer FTP, przez znudzonych życiem frustratów skanujących komputery. Pierwsza z dyrektyw opóźnia podanie klientowi identyfikacji serwera do momentu zautentykowania się klienta, druga sprawia, iż owa identyfikacja jest bezużyteczna.

3.4 Ustawienia autentykacji użytkowników

Zagadnienia autentykacji użytkowników do serwerów wirtualnych zostały rozwiązane następującymi fragmentami skryptu konfiguracyjnego:

```

<Global>
    ...
    RequireValidShell  off
    ...
</Global>
<VirtualHost virtual.net>
    ...
    AuthUserFile      /home/virtual/etc/passwd
    <Limit LOGIN>
        AllowAll
    </Limit>
    ...
    <Anonymous /home/virtual/pub>
        ...
        UserAlias      anonymous  virtual
        AuthAliasOnly  on

```

```

        <Limit LOGIN>
            AllowAll
        </Limit>
        ...
    </Anonymous>
    ...
</VirtualHost>

```

Dyrektywa *AuthUserFile* określa plik typu *passwd* o formacie opisanym w sekcji 2.2, który używany jest do autentykacji użytkowników. Na każdy serwer wirtualny przypada jeden plik *passwd*, w którym znajduje się lista użytkowników wraz z ich hasłami, identyfikatorami i specyfikacją katalogów domowych. Jeden z użytkowników (w tym przypadku *virtual*) ma dostęp do katalogu anonimowego z pełnymi prawami, ponieważ w pliku *passwd* jako jego katalog domowy podany jest ten z plikami konta anonimowego dostępu.

Opcja *RequireValidShell* z parametrem *off* powoduje, iż użytkownik, który nie posiada dostępu do sprawnej powłoki, może zautentykować się do serwera FTP. Dzięki temu rodzaj powłoki wyspecyfikowany w pliku */home/virtual/etc/passwd* nie ma żadnego wpływu na proces autentykacji i może być to, np. */bin/false* tak, jak to zaproponowaliśmy w przykładowym pliku *passwd*.

Autentykacja do konta anonimowego dostępu nie wymaga podania hasła. Konto to jest konfigurowane w bloku rozpoczynającym się dyrektywą *<Anonymous>*, która jako parametr przyjmuje ścieżkę do katalogu, który ma być udostępniany bez hasła. Podane w przykładzie opcje *UserAlias* i *AuthAliasOnly* sprawiają, iż przy anonimowym logowaniu użytkownik powinien podać login *anonymous*, lecz tak naprawdę zalogowany zostanie jako *virtual* z ograniczonymi prawami dostępu. Przy logowaniu się jako *virtual* hasło będzie nadal wymagane.

Opcja *AllowAll* umieszczona w bloku *<Limit LOGIN>* oznacza, iż możliwe jest zalogowanie się do danego kontekstu. Blokada dostępu do kontekstu serwera odbywa się przez zastąpienie *AllowAll* przez opcję *DenyAll*. W naszym przykładowym skrypcie umieściliśmy oddzielnie blok *<Limit LOGIN>* dla kont użytkowników i konta anonimowego dostępu, aby możliwe było ich niezależne włączanie i wyłączenie. Umieszczenie w obu sekcjach opcji *DenyAll* sprawi, iż nie będzie możliwości zalogowania się do danego serwera wirtualnego.

3.5 Uwwięzienie użytkownika w jego katalogu domowym

Domyślna konfiguracja demona *ProFTPD* umożliwia zalogowanemu użytkownikowi wędrówki po całej strukturze katalogów, co nie jest korzystne mając na względzie bezpieczeństwo serwera FTP. Zastosowanie dyrektywy *DefaultRoot* z parametrem *~*, jak w przykładzie poniżej, rozwiązuje ten problem.

```

<VirtualHost virtual.net>
    ...
    DefaultRoot    ~

```

```
...
</VirtualHost>
```

Zastosowanie omawianej opcji w sekcji konfiguracyjnej serwera wirtualnego, powoduje, że użytkownik nie może wyjść powyżej swojego katalogu domowego, który został określony w pliku *passwd* dla zadanego serwera wirtualnego.

Podobne zabiegi nie są wymagane w przypadku kont anonimowego dostępu do serwera FTP, ponieważ domyślnym zachowaniem demona *ProFTPD* jest ograniczenie swobody użytkownika anonimowego do struktury katalogów o korzeniu określonym przez paramter dyrektywy rozpoczynającej blok konfiguracyjny typu *<Anonymous>*, jak w przykładzie:

```
<VirtualHost virtual.net>
...
  <Anonymous /home/virtual/pub>
...
</Anonymous>
...
</VirtualHost>
```

Podany fragment skryptu konfiguracyjnego sprawia, że użytkownik, który loguje się anonimowo na serwerze *virtual.net*, nie może wyjść w strukturze katalogów powyżej katalogu: */home/virtual/pub*.

3.6 Ukrywanie nazw użytkownika i grupy

Po procesie autentykacji, w celach bezpieczeństwa, demon *ProFTPD* przełącza swojego efektywnego użytkownika oraz efektywną grupę na takie, które zostały określona w pliku konfiguracyjnym. W naszym przypadku wszystkie serwery wirtualne obsługiwane są przez użytkowników z jednej grupy *virtftp*. Każdemu serwerowi wirtualnemu odpowiada jeden użytkownik w systemie. Ustawienie grupy wykonuje się za pomocą dyrektywy *Group*, natomiast użytkownik wybierany jest opcją *User*. W podanym poniżej przykładzie serwerowi *virtual.net* odpowiada użytkownik systemowy *virtual*.¹

```
<Global>
...
  Group          virtftp
...
</Global>
<VirtualHost virtual.net>
...

```

¹Użytkownika i grupę określoną dyrektywami *User* i *Group* należy kojarzyć z użytkownikiem i grupą pochodzącymi z systemowych plików */etc/passwd* i */etc/group*, a nie z tymi, które pochodzą z pliku określonego przez *AuthUserFile*.

```

    User          virtual
    ...
</VirtualHost>

```

Funkcjonowanie serwera wirtualnego z jednym tylko identyfikatorem użytkownika sprawia, że klienci wykonujący komendę „LIST” otrzymywaliby niewłaściwe informacje o właścicielu i grupie, do której należą pliki. Dlatego też, w celach „kosmetycznych” warto zastosować dyrektywy *DirFakeUser* oraz *DirFakeGroup*. Parametrami tych opcji są odpowiednio nazwa użytkownika oraz nazwa grupy, które mają być wyświetlane przy podawaniu listy plików i katalogów. Zamiast nazwy można użyć znaku ~, za który podstawiana jest nazwa zalogowanego użytkownika, bądź grupy, do której on należy. W naszym skrypcie konfiguracyjnym zastosowanie podanych dyrektyw jest następujące:

```

<Global>
    ...
    DirFakeUser    on  ~
    DirFakeGroup   on  ~
    ...
</Global>
<VirtualHost virtual.net>
    ...
    <Anonymous /home/virtual/pub>
        ...
        DirFakeUser    on  ftp
        DirFakeGroup   on  ftp
        ...
    </Anonymous>
    ...
</VirtualHost>

```

Przy listowaniu plików, jako ich właściciel będzie podawany aktualnie zalogowany użytkownik serwera wirtualnego, a w przypadku dostępu anonimowego, jako nazwa użytkownika będzie podawane *ftp*.

3.7 Opcje dostępu do plików i katalogów

Podane niżej fragment skryptu konfiguracyjnego determinują pewne aspekty związane z dostępem do plików i katalogów wewnątrz serwera FTP.

```

<Global>
    ...
    PathDenyFilter    "(\\.quota)|(\.ftpaccess)$"
    Umask              022
    AllowOverwrite    on
    ...

```

```

</Global>
<VirtualHost virtual.net>
    ...
    <Anonymous /home/virtual/pub>
        ...
        HideNoAccess    on
        ...
    </Anonymous>
    ...
</VirtualHost>

```

Dyrektywa *PathDenyFilter*, wraz z parametrem będącym wyrażeniem regularnym, określa, jakie pliki nie mogą zostać utworzone na serwerze FTP w wyniku operacji *upload*. Zdecydowaliśmy się odrzucić możliwość utworzenia plików o nazwie *.quota* oraz *.ftpaccess*. Pierwszy z nich służy do sterowania ograniczeniem dostępnej przestrzeni dyskowej dla użytkownika i umożliwienie jego modyfikacji otwierałoby użytkownikowi okno do łatwego zwiększenia sobie dostępnej przestrzeni dyskowej. Pliki *.ftpaccess* służą do konfigurowania dostępu do plików, które znajdują się w katalogu, w którym umieszczono ten plik. Właściwie można by było udostępnić użytkownikom taką opcję, ale istnieje zbyt duże prawdopodobieństwo, że zrobią coś źle, więc lepiej nie dawać im zbyt wiele swobody.

Przy pomocy opcji *Umask* można określić oktalną maskę praw dostępu, które zostaną wyłączone w momencie tworzenia nowego pliku/katalogu. Opcja ta działa analogicznie do systemowego *umask(2)*. W naszym przykładowym skrypcie konfiguracyjnym wyłączamy możliwość zapisu dla grupy i dla pozostałych użytkowników.

Domyślna konfiguracja demona *ProFTPD* uniemożliwia nadpisywania plików. Ponieważ jest to bardzo pożądane w przypadku prywatnych kont użytkowników, możliwość nadpisywania plików jest uaktywniana przez dyrektywę *AllowOverwrite* z opcją *on*.

W sekcji konfiguracji konta anonimowego dostępu umieściliśmy komendę *HideNoAccess* z parametrem *on*, aby schować przed użytkownikiem pliki, do których nie ma on dostępu. Ostateczne ograniczenie dostępu do tych plików nastąpi w kolejnym podrozdziale, gdzie pojawi się dyrektywa *IgnoreHidden*.

3.8 Ustawienie dostępu do komend serwera FTP

Przy pomocy bloków *<Limit>* istnieje możliwość określenia praw dostępu do różnych komend serwera FTP w zależności od kontekstu, do którego w danej chwili zalogowany jest użytkownik. Konfiguracja dostępu może odbywać się dla każdej z komend z osobna, bądź też dla całych grup komend poprzez podanie symbolicznej nazwy tej grupy, np.: *ALL* – wszystkie komendy serwera FTP, *WRITE* – komendy, które dokonują zapis danych na serwerze, *READ* – komendy dotyczące odczytu plików z serwera, *DIRS* – komendy służące do poruszania się w strukturze katalogów. W przykładowym skrypcie konfiguracyjnym, którego interesujące

nas fragmenty znajdują się poniżej, włączyliśmy pełen dostęp do komend FTP dla użytkowników serwerów wirtualnych. W przypadku anonimowego dostępu do serwera wirtualnego, zalogowany użytkownik może czytać pliki i przemieszczać się w skrajności katalogów, natomiast nie ma on możliwości zapisu i tworzenia nowych plików i katalogów.

```
<VirtualHost virtual.net>
...
<Limit ALL>
    AllowAll
</Limit>
...
<Anonymous /home/virtual/pub>
...
<Limit WRITE>
    DenyAll
    IgnoreHidden    on
</Limit>

<Limit READ DIRS>
    AllowAll
    IgnoreHidden    on
</Limit>
...
</Anonymous>
...
</VirtualHost>
```

W sekcji dotyczącej anonimowego dostępu do serwera wirtualnego znajdują się dyrektywy *IgnoreHidden* z parametrem *on*, które sprawiają, że nie możliwe jest wykonanie komendy FTP, na pliku, bądź katalogu, który został wcześniej ukryty dyrektywą *HideNoAccess*.

3.9 Konfiguracja ograniczeń przestrzeni dyskowej

W zaproponowanej przez nas konfiguracji serwerów wirtualnych opieramy się na dwóch mechanizmach służących do ograniczenia zajmowanej przez użytkowników przestrzeni dyskowej (*quota*). Do kontroli zajmowanej przestrzeni dyskowej przypadającej na jeden serwer wirtualny korzystamy z możliwości udostępnianych przez system operacyjny w zakresie realizacji *quoty*. Wewnątrz danego serwera wirtualnego można nałożyć ograniczenie na każdego z użytkowników poprzez skorzystanie z modułu *mod_quota*. Konfiguracja tego modułu została przedstawiona poniżej:

```
<VirtualHost virtual.net>
```

```

...
Quotas          on
QuotaCalc       on
QuotaType       hard
QuotaBlockSize 1024
QuotaBlockName  kB
...
<Anonymous /home/virtual/pub>
    ...
    Quotas          off
    ...
</Anonymous>
...
</VirtualHost>

```

W podanym przykładzie *quota* jest włączona dyrektywą *Quotas* dla standardowych użytkowników serwera wirtualnego, natomiast jest wyłączona w przypadku konta anonimowego dostępu, ponieważ jego użytkownicy nie mają możliwości zapisu danych. Opcja *QuotaCalc on* powoduje, że zajętość dysku jest przeliczana przez serwer po każdej operacji, która może modyfikować zajętość miejsca na dysku oraz w przypadku, gdyby zajętość okazała się ujemna. Opcja *QuotaType hard* sprawia, iż w przypadku, gdyby ostatnio zapisywany na serwerze plik naruszał ilość dostępnego miejsca na dysku, jest on automatycznie kasowany. Pozostałe dyrektywy: *QuotaBlockSize* i *QuotaBlockName* mają jedynie charakter kosmetyczny i określają odpowiednio: wielkość bloku w bajtach i jego nazwę. Użytkownik otrzymuje informacje o dostępnym miejscu na serwerze przeliczoną według zadanej wielkości bloku.

3.10 Ograniczenie liczby dostępnych połączeń

W domyślnej konfiguracji demona *ProFTPD* nie narzucono ograniczeń na ilość możliwych połączeń z serwerem, co stanowi pewne zagrożenie ze względu na możliwość przeprowadzenia ataków typu *Denial of Service*. Ograniczenie można wprowadzić dyrektywą *MaxInstances* jak w poniższym przykładzie:

```
MaxInstances      300
```

Zaproponowana liczba (300) została wymyślona po uprzednim długim wpatrywaniu się w sufit i nie ma żadnego związku z rzeczywistością. W przypadku rzeczywistego serwera FTP musi ona stanowić kompromis między obciążeniem serwera oraz ilością klientów, którzy mogą zostać równocześnie obsłużeni przez serwer.

3.11 Konfiguracja sposobu logowania pracy serwera

Istotnym zagadnieniem przy konserwacji serwera FTP jest monitorowanie jego pracy, co można osiągnąć przez analizę logów generowanych przez demon

ProFTPD. Logi te dotyczą samego procesu autoryzacji użytkowników do serwera FTP oraz przebiegu sesji klienta. Ponieważ w rozpatrywanej przez nas konfiguracji mamy do czynienia z wieloma serwerami wirtualnymi, rozsądnym jest rozdzielanie logów pochodzących z poszczególnych serwerów wirtualnych. W naszym przykładowym skrypcie konfiguracyjnym, każdy z serwerów posiada własne pliki, do których loguje swoją pracę (*ExtendedLog*) oraz informacje o przesyłanych plikach (*TransferLog*). Zbiorcze logi, dotyczące procesu autoryzacji na wszystkich serwerach wirtualnych, zbierane są w osobnym pliku i mają one nieco inny format, określony przez dyrektywę *LogFormat*. Za konfigurację procesu logowania odpowiedzialne są następujące fragmenty skryptu konfiguracyjnego:

```
LogFormat    auth    "%s %u [\"%v\"] [%P] %l@%h[%a] %t \"%r\""
<Global>
...
ExtendedLog  /root/usr/var/proftpdauth.log AUTH auth
...
</Global>
<VirtualHost virtual.net>
...
ExtendedLog  /home/virtual/access.log WRITE,READ default
TransferLog  /home/virtual/xfer.log
...
</VirtualHost>
```

Dyrektywa *LogFormat* umożliwia ustalenie własnego formatu logów z pracy serwera FTP. Zaproponowany przez nas format o etykiecie *auth* jest ukierunkowany na logowanie przebiegu autoryzacji i jego zastosowanie ułatwia nam znacząco późniejsze generowanie statystyk o najczęściej odwiedzanych serwerach wirtualnych. Standardowy ciąg formatujący, którego nie trzeba umieszczać w skrypcie konfiguracyjnym, ma postać:

```
LogFormat    default "%h %l %u %t \"%r\" %s %b"
```

Znaczenie symboli, występujących w ciągach formatujących *default* i *auth*, jest następujące:

- %h* – nazwa maszyny zdalnej, z której następuje połączenie,
- %l* – nazwa użytkownika maszyny zdalnej (wg. identd),
- %u* – identyfikator zautentykowanego użytkownika lokalnego,
- %t* – data i godzina wywołania danej komendy,
- %r* – komenda otrzymana od klienta,
- %s* – status z wykonania komendy,
- %b* – ilość bajtów przesłanych jako rezultat wykonania komendy,
- %v* – nazwa serwera wirtualnego, którego dotyczy wpis w logach,
- %P* – pid procesu obsługującego dane połączenie,
- %a* – adres IP maszyny zdalnej, z której następuje połączenie.

Więcej informacji na temat logów demona *ProFTPD* znajduje się w sekcji 4, gdzie opisujemy sposób ich analizy.

4 Generowanie statystyk dostępu do serwera

Niniejsza sekcja prezentuje możliwości analizy logów demona *ProFTPD* do generowania statystyk z pracy serwera. Jako przykład zostanie przedstawiony skrypt napisany w języku *Perl*, który na podstawie zbiorczych logów dotyczących autentykacji użytkowników, wypisuje 10 najczęściej odwiedzanych wirtualnych serwerów FTP.

4.1 Przygotowanie formatu logowania

Logi zapisywane w standardowej konfiguracji demona *ProFTPD*, nie dają możliwości realizacji postawionego przez nas zadania, ponieważ nie zawierają nazwy serwera wirtualnego, do którego odnosi się dany rekord. Brak ten stał się powodem, dla którego zdefiniowaliśmy nowy format logowania o nazwie „auth” (sekcja 3.11). Najważniejsze zmiany dotyczyły wprowadzenia pola *%v*, które zawiera nazwę serwera wirtualnego oraz przeniesienie statusu z wykonania komendy protokołu FTP na początek wiersza (pole *%s*), aby wychwycenie interesujących nas zapisów stało się prostsze. Poza tymi dwoma, najistotniejszymi dla nas z punktu widzenia generowania statystyk, modyfikacjami, format logowania „auth” zawiera również informacje o zautentykowanym użytkowniku, identyfikatorze procesu, który obsługuje daną sesję FTP, informację o zdalnym użytkowniku i wydaną przez niego komendę wraz z datą i godzinę jej wykonania. Pole z nazwą serwera wirtualnego (*%v*) zostało dodatkowo otoczone parami znaków: [" i "], aby umożliwić jego ekstrakcję w przypadku, gdy zawiera białe znaki. Dodatkowo przyjęliśmy założenie, że nazwa serwera wirtualnego nie może zawierać znaku: ", co jest dla nas istotne przy konstrukcji wyrażenia regularnego, które dopasowuje się do tej nazwy.

Analiza logów o formacie „auth” polega na wyszukiwaniu wierszy rozpoczynających się liczbą 230, co oznacza wykonanie komendy zakończone pomyślną autentykacją użytkownika. Z tak wyznaczonych rekordów wyciągana jest nazwa serwera wirtualnego. Z każdą nazwą związany jest licznik, który zlicza ilość logowań do danego serwera zakończonych sukcesem. Po przeglądnięciu całego pliku z logami, uzyskane liczniki wraz ze związanymi z nimi nazwami można posortować malejąco względem liczby odwiedzin, otrzymując w ten sposób listę najczęściej odwiedzanych serwerów wirtualnych.

4.2 Generowanie statystyki

W dodatku B znajduje się kod źródłowy skryptu *stat.pl*. Przyjmuje on na standardowym wejściu logi demona *ProFTPD* (kilka wierszy pochodzących z takiego

```
331 virtual ["Virtual-1 FTP server"] [3036] UNKNOWN@virtual.net[192.168.100.1]\
[27/May/2002:15:59:22 +0200] ["USER anonymous"]
230 virtual ["Virtual-1 FTP server"] [3036] UNKNOWN@virtual.net[192.168.100.1]\
[27/May/2002:15:59:24 +0200] ["PASS hello"]
331 virtua3 ["Virtual-3 FTP server"] [3557] UNKNOWN@virtua3.net[192.168.100.3]\
[27/May/2002:16:55:47 +0200] ["USER user3"]
230 user3 ["Virtual-3 FTP server"] [3557] UNKNOWN@virtua3.net[192.168.100.3]\
[27/May/2002:16:55:50 +0200] ["PASS (hidden)"]
331 virtua4 ["Virtual-4 FTP server"] [3592] UNKNOWN@virtua4.net[192.168.100.4]\
[27/May/2002:17:02:03 +0200] ["USER anonymous"]
230 virtua4 ["Virtual-4 FTP server"] [3592] UNKNOWN@virtua4.net[192.168.100.4]\
[27/May/2002:17:02:07 +0200] ["PASS sdfg"]
331 virtua2 ["Virtual-2 FTP server"] [3722] UNKNOWN@virtua2.net[192.168.100.2]\
[27/May/2002:17:31:58 +0200] ["USER virtual"]
530 virtua2 ["Virtual-2 FTP server"] [3722] UNKNOWN@virtua2.net[192.168.100.2]\
[27/May/2002:17:31:58 +0200] ["PASS (hidden)"]
331 virtual ["Virtual-1 FTP server"] [3724] UNKNOWN@virtual.net[192.168.100.1]\
[27/May/2002:17:31:58 +0200] ["USER virtual"]
230 virtual ["Virtual-1 FTP server"] [3724] UNKNOWN@virtual.net[192.168.100.1]\
[27/May/2002:17:31:58 +0200] ["PASS (hidden)"]
```

Rysunek 2: Przykładowe logi demona *ProFTPD* (wiersze zostały podzielone ze względu na ich duży rozmiar).

pliku przedstawia rysunek 2) i wypisuje na standardowym wyjściu do dziesięciu najczęściej odwiedzanych serwerów wirtualnych (rysunek 3).

Pierwsza pętla *while* w skrypcie *stat.pl*:

```
while (<>) {
    ...
}
```

powoduje czytanie kolejnych wierszy logów ze standardowego wejścia i zapamiętywanie ich w zmiennej domyślnej *\$_*. Warunek:

```
if (/^230/) {
```

Najczęściej odwiedzane serwery wirtualne:

1. Virtual-1 FTP server (193)
2. Virtual-3 FTP server (184)
3. Virtual-4 FTP server (180)
4. Virtual-2 FTP server (161)

Rysunek 3: Przykładowy rezultat działania skryptu *stat.pl*.

```
    ...  
}
```

sprawdza, czy statusu wykonania komendy wynosi 230, co oznacza udaną autentycację do serwera wirtualnego. Weryfikacja tego warunku polega na próbie dopasowania do ostatnio wczytanego rekordu wyrażenia regularnego `^230` (liczba 230 na początku wiersza). Jeśli znaleziono interesujący wiersz, wykonywana jest nieco złożona na pierwszy rzut oka konstrukcja:

```
$count{$1}++ if (/\[\"([^\"]*)\"\\]/);
```

Powoduje ona zwiększenie licznika odwiedzin serwera wirtualnego o nazwie zawierającej się między parami znaków [" i "]. Cel ten realizowany jest przez próbę dopasowania wzorca `\[\"([^\"]*)\"\\]` do ostatnio wczytanego wiersza. Zawarta w wyrażeniu regularnym para nawiasów powoduje przechwycenie nazwy serwera do zmiennej `$1`. Jeśli dopasowanie powiodło się, zwiększany jest odpowiedni licznik, który przechowywany jest w tablicy asocjacyjnej `%count`, adresowanej nazwami serwerów wirtualnych.

Po wykonaniu pętli `while`, w tablicy asocjacyjnej `%count` znajdują się nazwy serwerów wirtualnych wraz ze zliczoną ilością odwiedzin. Kolejnym krokiem jest posortowanie nazw serwerów według liczby odwiedzin w porządku malejącym, co można osiągnąć sekwencją instrukcji:

```
@sorted = sort { $count{$b} <=> $count{$a} } keys %count;
```

W wyniku ich wykonania w tablicy `sorted` znajduje się lista kluczy tablicy asocjacyjnej `%count` posortowana według odpowiadających im wartości liczników. Za wypisanie listy najczęściej odwiedzanych serwerów wirtualnych odpowiada pętla `foreach` przebiegająca po wszystkich nazwach zapisanych w tablicy `sorted`:

```
$index = 0;  
foreach $host (@sorted) {  
    last if (++$index > $LIMIT);  
    printf "%3d. %s (%d)\n", $index, $host, $count{$host};  
}
```

Zawarta wewnątrz pętli instrukcja `last` powoduje przerwanie procesu wypisywania nazw serwerów, po wysłaniu na standardowe wyjście `$LIMIT` pierwszych nazw.

Przykładowe wywołanie skryptu `stat.pl` może wyglądać następująco:

```
# ./stat.pl < /var/log/proftpdauth.log
```

5 Podsumowanie

Demon `ProFTPD` może być z powodzeniem używany w zastosowaniach komercyjnych do budowy wirtualnych serwerów FTP. Po przygotowaniu przemyślanych

założeń świadczenia usług tego typu, jego konfiguracja nie powinna nastroczać większych trudności, ze względu na bardzo przejrzystą strukturę skryptów konfiguracyjnych. Na czytelność wpływa m.in. wydzielenie konfiguracji każdego z serwerów wirtualnych, a równocześnie możliwość zgrupowania wspólnych dyrektyw konfiguracyjnych. Modułowa budowa demona *ProFTPD* z wbudowanym standardowo modułem do listowania katalogów sprawia, iż nie wymaga on specjalnego przygotowania struktury katalogów w przypadku pracy w środowisku ze zmienionym korzeniem struktury katalogów (*chrooted environment*), w przeciwieństwie, np. do demona *wu-ftp*.

W swoim referacie oparliśmy autentykację o pliki typu *passwd*, osobne dla każdego z serwerów wirtualnych. Jest to rozwiązanie proste, ale w przypadku większych serwerów wydaje się, iż odpowiedniejsze byłoby skorzystanie z modułów autentykacji opartych o SQL, bądź o LDAP. Zastosowanie bazy danych, bądź usług katalogowych z użytkownikami i ich hasłami ułatwiłoby znacząco zarządzanie serwerem. Nie rozwijaliśmy jednak swojego referatu w tym kierunku, ponieważ wprowadziłoby to do niego zbyt wiele treści niekoniecznie związanych z demonem *ProFTPD*.

Literatura

- [1] ProFTPD User's Guide.
(<http://proftpd.linux.co.uk/localsite/Userguide/other/userguide.pdf>)
- [2] Mark Lowes. Professional FTP Daemon FAQ.
(<http://proftpd.org/docs/faq/faq.pdf>)
- [3] RFC-959 File Transfer Protocol (FTP).
(<http://www.rfc-editor.org/rfc/rfc959.txt>)

A Plik konfiguracyjny serwera wirtualnego

```
1 # Plik konfiguracyjny serwera ProFTPD. Uruchamia serwery
2 # wirtualne z wieloma kontami uzytkownikow i kontem goscia.
3
4 # Ustalenie limitu na ilosc polaczen ze strony klientow w
5 # celu ochrony przed atakami DOS
6 MaxInstances      300
7
8 # Wylaczenie glownego serwera FTP; aktywne sa jedynie
9 # serwery konfigurowane w sekcjach <VirtualHost>
10 Port              0
11
12 # Ustawienie "dokladnego" sposobu bindowania gniazdek
13 SocketBindTight  on
14
15 # Zdefiniowanie formatu logow dotyczacych autoryzacji
16 LogFormat         auth "%s %u [%v]" [%P] %l@h[%a] %t \"%r\""
17
18 # Opcje wspolne dla wszystkich serwerow wirtualnych
19 <Global>
20     # Nazwa grupy, pod ktora bedzie dzialal serwer FTP
21     Group          virtftp
22
23     # Oszukiwanie w nazwie wlasciela pliku przy listowaniu
24     # katalogu
25     DirFakeUser    on ~
26     DirFakeGroup   on ~
27
28     # Wylaczenie wymuszenia posiadania sprawnej powloki
29     # przez uzytkownika
30     RequireValidShell off
31
32     # Uniemozliwienie zapisu plikow o nazwach .quota i
33     # .ftpaccess
34     PathDenyFilter "(\\.quota)|\\.ftpaccess)$"
35
36     # Ustawienie maski praw dostepu dla nowo tworzonych
37     # plikow i katalogow
38     Umask          022
39
40     # Wlaczanie mozliwosci nadpisywania plikow
41     AllowOverwrite on
42
43     # Logowanie procesu autoryzacji do pliku
44     # /var/log/proftpdauth.log
45     ExtendedLog    /var/log/proftpdauth.log AUTH auth
46 </Global>
47
```

```

48 # Konfiguracja serwera wirtualnego o adresie virtual.net
49 <VirtualHost virtual.net>
50     # Nazwa serwera FTP
51     ServerName      "Virtual-1 FTP server"
52     # Adres e-mail do administratora serwera FTP
53     ServerAdmin     "ftp@virtual.net"
54
55     # Ustawienie nic nie znaczej identyfikacji serwera FTP
56     DeferWelcome    on
57     ServerIdent     on "FTP Server ready."
58
59     # Ustawienie lokalizacji pliku z identyfikatorami
60     # uzytkownikow i ich haslami
61     AuthUserFile    /home/virtual/etc/passwd
62
63     # Nazwa uzytkownika, pod ktora bedzie dzialal serwer FTP
64     User            virtual
65
66     # Uwiezienie uzytkownika w jego katalogu domowym
67     DefaultRoot     ~
68
69     # Uaktywnienie modulu mod_quota dla kont uzytkownikow
70     # wewnatrz serwera wirtualnego
71     Quotas          on
72     QuotaCalc       on
73     QuotaType       hard
74     # Informacje o quocie sa wyswietlane w kilobajtach
75     QuotaBlockSize 1024
76     QuotaBlockName  kB
77
78     # Uzytkownik moze wydawac wszystkie komendy protokolu
79     # FTP
80     <Limit ALL>
81         AllowAll
82     </Limit>
83
84     # Zezwolenie uzytkownikom na logowanie sie na serwerze
85     # wirtualnym
86     <Limit LOGIN>
87         AllowAll
88     </Limit>
89
90     # Ustawienie lokalizacji plikow z logami sesji FTP
91     ExtendedLog     /home/virtual/access.log WRITE,READ default
92     TransferLog     /home/virtual/xfer.log
93
94     # Konfiguracja konta goscia wewnatrz serwera wirtualnego
95     <Anonymous /home/virtual/pub>
96         # Na konto goscia mozna logowac sie tylko z loginem

```

```

97     # 'anonymous'
98     UserAlias      anonymous  virtual
99     AuthAliasOnly  on
100
101     # Wszystkie pliki i katalogi na listingach wystepuja
102     # jako nalezace do uzytkownika ftp i grupy ftp
103     DirFakeUser    on  ftp
104     DirFakeGroup   on  ftp
105
106     # Ukrycie w listingach plikow, do ktorych nie ma
107     # dostepu
108     HideNoAccess   on
109
110     # Wylaczenie quoty dla konta goscia
111     Quotas          off
112
113     # Umozliwienie logowania sie na koncie anonimowego
114     # dostepu
115     <Limit LOGIN>
116         AllowAll
117     </Limit>
118
119     # Wylaczenie praw do komend powodujacych zaspis
120     <Limit WRITE>
121         DenyAll
122         IgnoreHidden  on
123     </Limit>
124
125     # Umozliwienie odczytu plikow z konta goscia
126     <Limit READ DIRS>
127         AllowAll
128         IgnoreHidden  on
129     </Limit>
130 </Anonymous>
131 </VirtualHost>
132
133 # Konfiguracja serwera wirtualnego o adresie virtua2.net
134 <VirtualHost virtua2.net>
135     ...
136
137
138     # Sposob konfiguracji kolejnych serwerow wirtualnych
139     # jest analogiczny do konfiguracji serwera virtual.net
140
141     ...
142
143 </VirtualHost>

```

B Skrypt do generowania statystyk dostępu do serwera

```
1  #!/usr/bin/perl -w
2
3  # Okreslenie ilosci najczesciej odwiedzanych serwerow
4  # wirtualnych do wypisania
5  $LIMIT = 10;
6
7  while (<>) {          # Czytaj kolejne wiersze wejścia
8      if (/^230/) {    # Czy status z wykonania wynosi 230?
9          # Zwiększenie licznika odwiedzin serwera o nazwie
10         # zawartej między parami znaków: [" i "]
11         $count{$1}++ if (/\[\"([^\"]*)\"\\\]/);
12     }
13 }
14
15 print "Najczęściej odwiedzane serwery wirtualne:\n";
16
17 # Sortowanie nazw serwerow wirtualnych wedlug liczby
18 # odwiedzin (wynik w tablicy @sorted)
19 @sorted = sort { $count{$b} <=> $count{$a} } keys %count;
20 $index = 0;          # Zeruj licznik wypisanych nazw
21 foreach $host (@sorted) {
22     # Przerwanie petli, jesli wypisano juz $LIMIT nazw
23     last if (++$index > $LIMIT);
24     # Wypisanie nazwy serwera wraz z liczba odwiedzin
25     printf "%3d. %s (%d)\n", $index, $host, $count{$host};
26 }
```