Debian GNU/Linux 3.1. Biblia

Autorzy: Benjamin Mako Hill, David B. Harris, Jaldhar Vyas
Tłumaczenie: Małgorzata Czart, Grzegorz Kowalczyk, Leszek Sagalara
ISBN: 83-246-0287-9
Tytuł oryginału: Debian GNU/Linux 3.1 Bible
Format: B5, stron: 704

Kompendium wiedzy o systemie Debian GNU/Linux

- System Debian GNU/Linux 3.1 i dodatkowe pakiety oprogramowania
- Zasady korzystania z powłoki tekstowej oraz środowisk graficznych KDE i GNOME
- Linux w pracy i w domu
- Debian jako baza dla wydajnego i stabilnego serwera internetowego

Debian GNU/Linux to jedyna dystrybucja Linuksa niepowiązana w żaden sposób z jakąkolwiek firmą – tworzy ją rzesza ochotników. Rewelacyjny system zarządzania pakietami oprogramowania oraz stabilność i uniwersalność to główne źródła znakomitej reputacji Debiana. Tę właśnie dystrybucję coraz częściej wybierają administratorzy serwerów poszukujący wydajnego systemu operacyjnego. Z kolei twórcy innych dystrybucji Linuksa wykorzystują ją jako bazę dla nowych produktów.

Książka „Debian GNU/Linux 3.1. Biblia" to kompleksowe źródło wiedzy o Debianie i oryginalnych zasadach korzystania z niego. Dzięki zamieszczonym tu informacjom dowiesz się, jak zainstalować i skonfigurować system i jak nim zarządzać. Nauczysz się pracować w trybie tekstowym oraz wykorzystywać środowiska graficzne KDE i GNOME. Poznasz dołączone do tej dystrybucji Linuksa oprogramowanie, skonfigurujesz połączenia sieciowe i zbudujesz serwer internetowy.

- Instalacja Debiana
- Praca z systemem plików
- Korzystanie z powłoki tekstowej
- Instalowanie i usuwanie pakietów oprogramowania
- Zarządzanie kontami użytkowników
- Tworzenie kopii bezpieczeństwa
- Praca w środowisku graficznym
- Korzystanie z internetu i poczty elektronicznej
- Tworzenie i publikowanie dokumentów
- Obróbka cyfrowych fotografii
- Konfigurowanie usług sieciowych
- Serwery WWW, FTP, pocztowy oraz DNS
- Bazy danych

Każdy użytkownik Debiana, niezależnie od doświadczenia, może nauczyć się z tej książki czegoś nowego

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

Rozdział 5.
Podstawy zarządzania systemem Debian GNU/Linux

W rozdziale m.in.:

- Korzystanie z konta użytkownika root
- Dodawanie nowych kont użytkowników oraz grup
- Usuwanie kont użytkowników oraz grup
- Podstawowe informacje o połączeniach sieciowych

Zarządzanie współczesnymi systemami operacyjnymi jest bardzo złożonym zagadnieniem, mimo że idea leżąca u jego podstaw jest trywialnie prosta i spotykana również w wielu innych środowiskach: po pierwsze, nie szkodzić — a diabeł jak zwykle tkwi w szczegółach.

W niniejszym rozdziale omówimy szereg zagadnień związanych z zarządzaniem nowoczesnym systemem operacyjnym, ze szczególnym uwzględnieniem systemów Debian. Zasady, które tutaj poznasz, powinny mieć zastosowanie w każdym innym systemie GNU/Linux, nawet jeżeli będą się różnić pewnymi szczegółami. Jeżeli nie jesteś administratorem swojego systemu ani nie zajmujesz się żadnym innym systemem (np. zdalnym), to w zasadzie możesz spokojnie pominąć ten rozdział, choć i tak serdecznie zachęcamy Cię do jego lektury — posiadając wiedzę przedstawioną w tym rozdziale, będziesz w stanie lepiej zrozumieć niektóre decyzje, jakie podejmował bądź podejmuje administrator Twojego systemu, a być może nawet będziesz w stanie zasugerować mu kilka ciekawych rozwiązań.

Podobnie jak ma to miejsce w przypadku lekarza, administrator systemu komputerowego jest w zasadzie odpowiedzialny za dobre samopoczucie innych jego użytkowników. Jeżeli dany administrator pracuje w danej firmie, to zapewne jednym z jego najważniejszych zadań będzie zapewnienie spójności, bezpieczeństwa oraz prywatności poufnych danych swojej firmy, a także (a może przede wszystkim) danych klienta. Nietrudno się domyślić, że niewłaściwie funkcjonująca infrastruktura elektronicznych systemów przetwarzania danych, niezbędnych do prawidłowego przebiegu procesów biznesowych danej

firmy, może kosztować pracowników utratę miejsc pracy, ponieważ z pewnością firma taka nie będzie w stanie utrzymać poprawnych kontaktów ze swoimi klientami. Nie trzeba tutaj chyba specjalnie wyjaśniać, dlaczego błędy administratora systemów komputerowych pracujących w środowiskach krytycznych (np. systemy komputerowe szpitali czy też systemy militarne) mogą kosztować czyjeś życie, a z kolei jego prawidłowe decyzje mogą czyjeś życie uratować.

Zdecydowana większość administratorów systemów komputerowych na szczęście nie musi pracować w takich warunkach wysokiego stresu, co jednak w niczym nie zmienia faktu, że każdy administrator powinien profesjonalnie traktować swoje obowiązki, nawet jeżeli podejmowane przez niego kroki na pierwszy rzut oka mogłyby się wydawać nieco przesadzone. Słabe i nieudolne zarządzanie systemem komputerowym może w rezultacie doprowadzić do jego przejęcia przez potencjalnych włamywaczy i hakerów (w złym tego słowa znaczeniu) i wykorzystania go do różnych niecnych celów; taka możliwość „cichego" przejęcia kontroli nad danym systemem wymaga od administratorów czujności i doświadczenia.

Użytkownik, grupa i zarządzanie z poziomu konta użytkownika root

Fundamentalnym założeniem systemów wieloużytkownikowych, takich jak Debian GNU/Linux, jest zasada separacji uprawnień (ang. *privilege separation*). Ponieważ jednak każdy proces działający w danym systemie może mieć dostęp do innych części systemu i prawa do ich modyfikacji, to model separacji uprawnień napotyka tutaj na coraz to nowe, piętrzące się problemy. W takim modelu błąd występujący w nawet najmniejszej, najbardziej niewinnej aplikacji może teoretycznie spowodować usunięcie wszystkich danych przechowywanych w danym systemie czy też umożliwić danemu użytkownikowi nieautoryzowany, pełny dostęp do wszystkich danych innych użytkowników. Domyślnie system kontroli dostępu w systemie Debian GNU/Linux jest oparty na kontach użytkowników i grupach. Dostęp do poszczególnych zasobów jest determinowany przez konto użytkownika oraz grupy użytkowników, których dany użytkownik jest członkiem. Każdy proces uruchomiony przez danego użytkownika posiada dokładnie takie same prawa jak dany użytkownik. Z wyjątkiem kilku specjalnych sytuacji dany użytkownik nie ma możliwości uruchamiania aplikacji na prawach innego użytkownika.

System Linux potrafi obsługiwać różne modele separacji uprawnień; jednym z najbardziej ostatnimi czasy popularnych jest SELinux, czyli w pełnej nazwie Security Enhanced Linux opracowany w Stanach Zjednoczonych przez tamtejszą Narodową Agencję Bezpieczeństwa (ang. *National Security Agency*). SELinux mocno wykracza nawet poza standardowy model separacji uprawnień — oprócz tradycyjnego mechanizmu kont użytkowników oraz grup użytkowników pozwala administratorowi na szczegółowe zdefiniowanie rodzaju operacji, jakie dana aplikacja może wykonywać na określonych zasobach systemowych. Więcej szczegółowych informacji na temat dystrybucji SELinux znajdziesz na stronie *http://www.nsa.gov/selinux/*, z kolei na stronie *http://www.lurking-grue.org/gettingstarted_newselinuxHOWTO.html* znajdziesz sporo ciekawych informacji na temat sposobu instalacji oraz wykorzystywania systemu SELinux opartego na dystrybucji Debian.

Separacja uprawnień jest widoczna w każdej części systemu Debian; procesy działające nieprzerwanie, takie jak np. serwery WWW, zazwyczaj posiadają swoje własne konto użytkownika oraz grupę — w podobny sposób funkcjonuje wiele innych procesów. Innym przykładem może być dostęp do poszczególnych urządzeń sprzętowych (takich jak np. karta dźwiękowa czy karta graficzna), który również jest kontrolowany poprzez odpowiednie grupy, dzięki czemu Ty oraz wybrani użytkownicy macie bezpośredni dostęp do takich urządzeń, podczas gdy aplikacje niezwiązane z danym zadaniem mogą być takiego dostępu pozbawione.

Konto użytkownika root

Konto użytkownika root, czy też inaczej mówiąc konto *superużytkownika*, jest tradycyjnym kontem administratora systemu od samego początku istnienia systemów GNU/Linux. Użytkownik zalogowany na tym koncie ma pełne prawa do sprawdzenia i modyfikacji dowolnie wybranej części systemu. Za takimi uprawnieniami idzie jednak duża odpowiedzialność: kiedy pracujesz na koncie root, znacząco rośnie prawdopodobieństwo, że system zostanie przypadkowo uszkodzony bądź nawet całkowicie zniszczony, włączając w to potencjalne usunięcie danych, których już nigdy nie będziesz w stanie odzyskać. W przypadku pracy na komputerze domowym zazwyczaj Ty sam jesteś administratorem swojego systemu i tym samym to właśnie Ty korzystasz z konta root; z kolei jeżeli pracujesz w jakiejś firmie, to taka rola niekoniecznie musi przypaść Tobie.

Ogólnie mówiąc, konto użytkownika root powinno być używane tylko w sytuacjach, gdy jest to absolutnie konieczne. Redukcja zapotrzebowania na używanie konta root w codziennej pracy jest przedmiotem wielu obecnie działających projektów GNU/Linux. Nie zmienia to jednak w niczym faktu, że używanie konta root do normalnych, codziennych zadań jest praktyką niemalże powszechną. Bezpieczeństwo pracy jest oczywiście niezmiernie istotnym czynnikiem, ale z drugiej strony wygoda i łatwość pracy z systemem jest również czynnikiem nie bez znaczenia. Za każdym razem, kiedy podczas pracy pojawia się na ekranie komunikat, że nie masz odpowiednich uprawnień do wykonania danej operacji, rośnie prawdopodobieństwo, że będziesz chciał skorzystać z uprawnień konta użytkownika root.

Jeśli już musisz użyć konta root, to zalecanym sposobem postępowania jest jednoczesne prowadzenie gdzieś w pliku szczegółowych notatek na temat przeprowadzanych operacji, tak abyś w momencie, kiedy popełnisz jakiś błąd, miał możliwość sprawdzenia, jakie operacje wykonywałeś ostatnio i w jaki sposób przywrócić funkcjonalność systemu. Jeżeli w danym systemie pracuje więcej niż jeden administrator, to prowadzenie takiego dziennika operacji na koncie root staje się tym bardziej istotne — pozwala innym administratorom na sprawdzenie, jakie operacje zostały wykonane przez Ciebie i odwrotnie, pozwala Tobie na zorientowanie się, czego dokonali Twoi zmiennicy.

Szybki dostęp do uprawnień konta użytkownika root

W zasadzie istnieją dwie główne metody uzyskania uprawnień związanych z kontem użytkownika root: pełna sesja związana z zalogowaniem się na koncie użytkownika root (o której opowiemy bardziej szczegółowo w podrozdziale „Pełna sesja użytkownika root" w dalszej części rozdziału) lub też chwilowe podniesienie uprawnień na czas wykonania

pojedynczych poleceń, o czym powiemy za chwilę. Ta ostatnia metoda podnoszenia uprawnień jest bardzo elastyczna i ma liczne bardzo przydatne funkcje, takie jak umożliwienie śledzenia wykonywanych operacji, pliki konfiguracyjne dostosowane do indywidualnych wymagań poszczególnych administratorów, współdzielenie sesji X11 i wiele innych.

Instalacja i konfiguracja pakietu sudo

Pakiet `sudo` pozwala administratorowi na zapewnienie sobie (podczas codziennej pracy na normalnym, regularnym koncie) i innym, uprawnionym użytkownikom możliwości uruchamiania poszczególnych poleceń i aplikacji na prawach konta użytkownika root. Pakiet ten rejestruje również wszystkie polecenia, które zostały wykonane via polecenie `sudo`, tworząc w ten sposób znakomity dziennik wykonywanych operacji. Wreszcie, korzystanie z polecenia `sudo` jest bardzo proste. Dzięki temu, że `sudo` domyślnie zachowuje zmienne środowiskowe takie jak `$HOME` czy `$XAUTHORITY`, pakiet `sudo` jest obecnie jednym z najlepszych i najwygodniejszych sposobów uruchamiania poszczególnych poleceń i aplikacji z poziomu uprawnień użytkownika root.

Aby zainstalować pakiet `sudo`, powinieneś wykonać polecenie `apt-get install sudo`. Instalacja tego pakietu nie wymaga żadnej interakcji ze strony użytkownika, stąd nie zamieszczaliśmy tutaj żadnych komunikatów, jakie mogą pojawiać się w czasie instalacji. Jedynym komunikatem, o którym warto jednak tutaj wspomnieć, jest komunikat informujący, że właśnie tworzony jest domyślny plik */etc/sudoers*. Plik ten jest tak naprawdę plikiem konfiguracyjnym polecenia `sudo` i odpowiada za to, kto może korzystać z polecenia `sudo` oraz jakie polecenia może uruchamiać. Ponieważ nawet przypadkowy błąd w tym pliku stanowi poważny problem z punktu widzenia bezpieczeństwa systemu, to plik konfiguracyjny polecenia `sudo` nie może być normalnie edytowany.

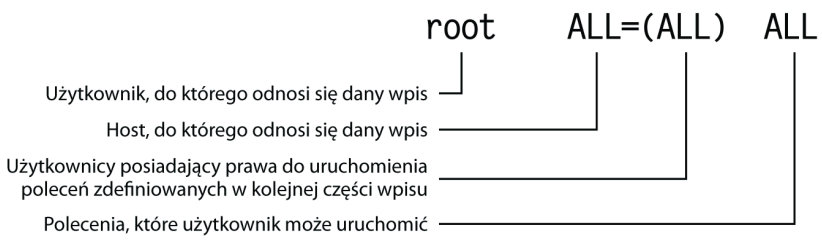
W większości przypadków będziesz jednak chciał dokonać odpowiedniej modyfikacji domyślnej zawartości pliku */etc/sudoers*, tak abyś mógł korzystać z polecenia `sudo` podczas pracy na normalnym, regularnym koncie użytkownika — oczywiście po zainstalowaniu system nie będzie automatycznie tak skonfigurowany. Aby mieć możliwość zmiany zawartości pliku */etc/sudoers*, powinieneś z poziomu uprawnień użytkownika root wykonać polecenie `visudo`. Polecenie to na podstawie zmiennych środowiskowych `$VISUAL` oraz `$EDITOR` wybiera domyślny edytor i otwiera w nim tymczasową kopię pliku */etc/sudoers* do edycji. Jeżeli chcesz zmienić domyślny edytor tekstu, to powinieneś przykładowo wykonać polecenie `export EDITOR=nano`. Po zakończeniu edycji pliku i wyjściu z edytora polecenie `visudo` sprawdzi składnię poszczególnych wpisów, aby była pewność, że nie został popełniony żaden błąd. Jeżeli składnia wpisów jest poprawna, to zostają one przeniesione do właściwego pliku */etc/sudoers* i proces zostaje zakończony; jeżeli składnia nie jest poprawna, to zostaniesz poproszony o anulowanie wprowadzonych zmian bądź poprawienie błędnych wpisów w pliku tymczasowym.

Domyślna zawartość pliku */etc/sudoers* powinna być identyczna bądź też bardzo zbliżona do tej przedstawionej poniżej:

```
# sudoers file.
#
# This file MUST be edited with the "visudo" command as root.
#
```

```
# See the man page for details on how to write a sudoers file.
#

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL) ALL
```

Jak widać, w zdecydowanej większości zawartość tego pliku stanowią komentarze i puste wiersze. Podobnie jak to ma miejsce w wielu innych plikach konfiguracyjnych systemu Debian, wszystkie wiersze, które rozpoczynają się od znaku #, bądź też ciągi znaków, które występują w dowolnym wierszu po znaku #, są traktowane jako komentarz i ignorowane przez powłokę. Pierwsze trzy wiersze komentarzy powinny być oczywiste[1]. Kolejne trzy wiersze, czyli specyfikacja hosta (ang. *Host*), użytkownika (ang. *User*) oraz polecenia (ang. *Command*) wskazują na sugerowaną strukturę wpisów w tym pliku. Najbardziej interesujący jest ostatni wiersz, stanowiący całą, efektywną, domyślną konfigurację pliku *sudoers*. Wpis ten umożliwia użytkownikowi root uruchamianie dowolnych poleceń, na dowolnych hostach i na prawach dowolnych użytkowników. Taki rodzaj wiersza definiuje uprawnienia danego użytkownika, czyli inaczej mówiąc określa, który użytkownik ma jakie uprawnienia oraz na jakich hostach. Na rysunku 5.1 przedstawiono krótki opis poszczególnych elementów wpisów, jakie możesz znaleźć w pliku *sudoers*.



Rysunek 5.1. *Specyfikacja przywilejów dla polecenia sudo*

Jak zapewne zauważyłeś, składniki wiersza przedstawionego na rysunku 5.1 ujawniają kilka ciekawych sekretów. Pierwszy element, alias użytkownika (ang. *User_Alias*), sam w sobie nie jest zbyt ciekawy, ale za to nie można tego powiedzieć o jego pozostałych trzech towarzyszach. Drugi element, alias hosta (ang. *Host_Alias*), wskazuje, że jeden plik */etc/sudoers* może być wykorzystywany przez wiele różnych komputerów. Nazwy tych komputerów są następnie sprawdzane przez polecenie `sudo` i dzięki takiemu rozwiązaniu możesz w bardzo elastyczny i wygodny sposób ograniczać dostęp na jednych maszynach i jednocześnie odpowiednio poszerzać dostęp na innych maszynach. Trzeci element (ang. *Runas_Alias*) pozwala na zdefiniowanie nazwy konta użytkownika, na którego prawach będą uruchamiane polecenia zdefiniowane w kolejnej części wpisu. Zapis taki wskazuje, że polecenie `sudo` może być wykorzystywane do uruchamiania

[1] `# sudoers file` — plik *sudoers*; `# This file MUST be edited with…` — ten plik MUSI być edytowany przy użyciu polecenia `visudo` wywoływanego przez użytkownika root; `# See the man page for details…` — więcej informacji na temat tworzenia pliku *sudoers* znajdziesz na stronach podręcznika `man` — *przyp. tłum.*

wybranych poleceń przez użytkowników innych niż root. Jest to szczególnie użyteczne podczas rozwiązywania problemów związanych z prawami dostępu innych użytkowników. Czwarty składnik, lista poleceń (ang. *Cmnd_Alias*), pozwala na ograniczenie rozszerzonych praw dostępu tylko do wybranych poleceń. Jest to bardzo bezpieczny sposób na nadanie użytkownikowi dodatkowych, zwiększonych uprawnień; przykładowo, możesz z tego skorzystać w sytuacji, kiedy jedyny komputer dysponujący połączeniem typu *dial-up* z siecią Internet ma kilku użytkowników i każdy z nich musi mieć możliwość nawiązania odpowiedniego połączenia modemowego. Krótko mówiąc, składnia wpisów w pliku *sudoers* jest następująca:

```
Alias_użytkownika   Alias_hosta=(Alias_konta_uruchamiającego_polecenia) Lista_poleceń
```

Poszczególne elementy składowe są nazywane aliasami, ponieważ mogą się odnosić bezpośrednio do użytkowników, hostów i poleceń, ale równie dobrze mogą być zastąpione specjalnymi słowami kluczowymi, jak np. `ALL`[2], bądź mogą odwoływać się do list. Do list mogą zostać również przypisane odpowiednie słowa kluczowe. Przykładowo, wykonanie polecenia przedstawionego poniżej spowoduje, że użytkownik `franek` otrzyma prawa do uruchamiania poleceń `/bin/ls` oraz `/bin/cp` na prawach użytkowników root lub `daemon` na komputerach (hostach) o nazwach `helios` i `antares`:

```
franek   helios, antares = (root, daemon)  /bin/ls, /bin/cp
```

Pokażemy Ci teraz, w jaki sposób możesz nadać dla swojego normalnego, nieuprzywilejowanego konta użytkownika możliwość uruchamiania poprzez polecenie `sudo` dowolnych innych poleceń na prawach użytkownika root. Ponieważ i tak masz możliwość zalogowania się bezpośrednio na konto użytkownika root, to nie ma żadnego sensu, aby z poziomu Twojego normalnego konta użytkownika ograniczać Ci w jakikolwiek sposób dostęp do konta root. Aby tego dokonać, powinieneś do pliku */etc/sudoers* dodać przedstawiony poniżej wiersz, z tym że ciąg znaków *nazwakonta* powinieneś zastąpić prawdziwą nazwą swojego konta użytkownika. Pamiętaj, że jeżeli chcesz zmodyfikować zawartość pliku */etc/sudoers*, to musisz najpierw na prawach użytkownika root wykonać polecenie `visudo`:

```
nazwakonta   ALL=(ALL) ALL
```

Zapisz zmodyfikowany plik; od tego momentu powinieneś mieć możliwość wykonywania poprzez polecenie `sudo` dowolnych poleceń na prawach użytkownika root z poziomu swojego normalnego, nieuprzywilejowanego konta użytkownika. Jeżeli chcesz się dowiedzieć, jak utworzyć nieco bardziej zaawansowaną zawartość pliku */etc/sudoers*, na przykład aby dać niektórym wybranym użytkownikom ograniczone prawa użytkownika root, powinieneś zapoznać się z treścią podręcznika `man` dla polecenia `sudo`. Możesz ją wyświetlić na ekranie, wykonując polecenie `man 5 sudoers`.

Zastosowanie polecenia sudo

Teraz, kiedy polecenie `sudo` jest już zainstalowane i w pełni skonfigurowane na Twoim komputerze, możesz (a w zasadzie nawet powinieneś) korzystać z niego do wykonywania codziennych zadań związanych z zarządzaniem systemem. Wyloguj się z konta root, a następnie zaloguj się na swoje normalne, nieuprzywilejowane konto. Aby zapoznać się ze sposobem działania polecenia `sudo`, powinieneś teraz wykonać następujące polecenie:

[2] Z ang.: wszyscy — *przyp. tłum.*

```
helion@debian:~$ sudo whoami

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

Password:
```

Po pojawieniu się na ekranie znaku zachęty powinieneś wpisać swoje hasło użytkownika. Jeżeli wszystko podałeś poprawnie, to `sudo` wykona polecenie będące jego argumentem (w naszym przypadku jest to polecenie `whoami`) na prawach użytkownika root.

Od tego momentu zakładamy, że jesteś zalogowany na swoje normalne, nieuprzywilejowane konto użytkownika, a nie konto użytkownika root. Jeżeli użycie konta użytkownika root będzie niezbędne, to każdorazowo będziemy Cię o tym dokładnie informować.

Polecenie `sudo` może być wykorzystywane bardzo często i w zasadzie może działać jak alternatywne rozwiązanie w stosunku do ciągłej pracy na koncie użytkownika root — z tym że teraz bezpieczeństwo całego systemu nie zostało poświęcone na ołtarzu wygody bądź niefrasobliwości jego administratora…

Gdy uruchamiasz polecenie `sudo` po raz pierwszy oraz za każdym razem, kiedy uruchamiasz je po przerwie (domyślnie jest to 10 minut), polecenie `sudo` prosi o podanie hasła. Do pewnego stopnia polecenie to zachowuje się więc jak swego rodzaju wygaszacz ekranu; zostało zaprojektowane tak, aby zapobiegać sytuacji, że nieautoryzowana osoba trzecia siądzie przed klawiaturą Twojego komputera i uzyska dostęp do konta root w momencie, kiedy akurat na chwilkę wyszedłeś z pokoju (ale pozostawiłeś zalogowaną sesję na konsoli). Zwróć uwagę, że prosząc o wpisanie hasła, polecenie `sudo` oczekuje, że wpiszesz hasło do swojego własnego, nieuprzywilejowanego konta użytkownika — a nie konta użytkownika root. Dzięki takiemu rozwiązaniu możesz bezpiecznie udostępniać innym użytkownikom uprawnienia użytkownika root bez konieczności udostępniania im hasła do tego uprzywilejowanego konta (co przecież nie miałoby żadnego sensu i pozwalało na pełny dostęp).

Polecenie `sudo` posiada cały szereg różnych opcji, które zostały dokładnie opisane na stronach podręcznika `man` dla tego polecenia — jeżeli chcesz dowiedzieć się, jak uruchamiać wybrane polecenia na prawach użytkowników innych niż root, lub zapoznać się z innymi dodatkowymi możliwościami tego polecenia, powinieneś szczegółowo zapoznać się ze stroną podręcznika `man` poświęconą poleceniu `sudo`. Oprócz tych opcji i przełączników dla polecenia `sudo` jako argument powinieneś podać pełny wiersz polecenia, które chcesz wykonać. Przykładowo, jeżeli wpiszesz polecenie `sudo ls -lh --color`, to oznacza to, że polecenie `ls -lh --color` zostanie uruchomione na prawach użytkownika root. W środowisku, w którym większość zadań związanych z zarządzaniem systemem operacyjnym wykonywana jest z poziomu wiersza poleceń powłoki systemu, taka funkcjonalność polecenia `sudo` jest naprawdę trudna do przecenienia.

Polecenie sudo domyślnie zachowuje wszystkie zmienne środowiskowe. Oznacza to, że wartość Twojej zmiennej $HOME oraz wszystkich innych zmiennych pozostaje nienaruszona. Kiedy uruchamiasz dowolną aplikację, poszukuje ona swoich plików konfiguracyjnych w miejscu wskazywanym przez wartość zmiennej $HOME (czyli w Twoim katalogu domowym) — czyli w efekcie końcowym aplikacja skorzysta z plików konfiguracyjnych zlokalizowanych w Twoim katalogu domowym, a nie w katalogu */root/* (który jest standardowym katalogiem domowym użytkownika root). Jest to ogromna zaleta zwłaszcza w przypadku systemów, na których pracuje kilku administratorów. Dzięki takiemu rozwiązaniu każdy z indywidualnych administratorów może mieć swój własny zestaw plików konfiguracyjnych dla poszczególnych poleceń i aplikacji, stąd nie muszą oni już dłużej polegać na jednym jedynym pliku konfiguracyjnym, który kiedyś dawno temu utworzył jakiś „najważniejszy" administrator. Co więcej, również wartości zmiennych $XAUTHORITY oraz $DISPLAY zostają zachowane. Oznacza to w praktyce, że jeżeli korzystasz ze środowiska graficznego, to będziesz mógł uruchamiać aplikacje graficzne na prawach użytkownika root bez żadnych dodatkowych wysiłków — i to w całkowicie bezpieczny sposób, bez konieczności zezwalania poszczególnym użytkownikom na podłączanie się do Twojej sesji środowiska graficznego (co potencjalnie mogłoby umożliwić takim użytkownikom przechwycenie używanych przez Ciebie haseł).

Przeglądanie i analiza dzienników polecenia sudo

Polecenie sudo domyślnie zapisuje wszystkie polecenia wykonane przez danego użytkownika, korzystając z demona syslog; w systemie Debian wszystkie informacje zostają zapisane w pliku */var/log/auth.log*. Warto tutaj jednak zwrócić uwagę na jeden dość istotny fakt: sudo zapisuje w dzienniku tylko polecenia, które zostały wydane bezpośrednio. Przykładowo, jeżeli uruchomisz powłokę użytkownika root, korzystając z polecenia sudo su -, to takie wydarzenie zostanie zarejestrowane, niemniej jednak żadne z poleceń uruchomionych później, podczas sesji powłoki root, nie zostanie już zarejestrowane. Poniżej zamieszczamy przykładowy wiersz dziennika zdarzeń sudo (plik */var/log/auth.log*):

```
Dec  3 21:58:16 debian sudo:   helion : TTY=pts/1 ; PWD=/home/helion ; USER=root ;
COMMAND=/bin/su -
```

Pierwsza część wiersza aż do słowa sudo to, krótko mówiąc, znacznik czasowy danego wpisu dziennika, obejmujący datę, czas oraz nazwę hosta, na którym zostało wykonane polecenie sudo (w naszym przypadku debian). Kolejne pola to odpowiednio (w kolejności występowania): nazwa użytkownika, który wykonał polecenie sudo (w naszym przypadku helion), nazwa terminalu, na którym dany użytkownik był zalogowany w czasie wykonywania polecenia (w naszym przypadku jest to pts/1), bieżący katalog roboczy użytkownika (w naszym przypadku /home/helion), nazwa użytkownika, na którego prawach zostało wykonane dane polecenie (w naszym przypadku root) i wreszcie nazwa uruchamianego polecenia (w naszym przypadku /bin/su -).

Pełna sesja użytkownika root

Pomimo iż polecenie sudo jest bezpieczne, szybkie i wygodne, to jednak nie jest doskonałe. Od czasu do czasu nie będziesz miał innego wyjścia, jak tylko zalogować się do powłoki bezpośrednio na konto użytkownika root. Jednym z najczęstszych powodów

takiego postępowania jest konieczność przeglądania katalogów, które normalnie nie są dostępne dla żadnego innego użytkownika poza root. Jako dobry przykład możemy przytoczyć zapis następującej sesji:

```
helion@debian:~$ cd /var/spool/exim4/
bash: cd: /var/spool/exim4/: Brak dostępu
helion@debian:~$ sudo cd /var/spool/exim4/
sudo: cd: command not found
helion@debian:~$
```

Jak widać, katalog */var/spool/exim4/* nie jest dostępny dla normalnego użytkownika, a ponieważ polecenie `cd` jest wbudowanym poleceniem powłoki i nie można uruchomić go poprzez polecenie `sudo`, to nie ma innej możliwości dostania się do tego katalogu, jak tylko poprzez zalogowanie się do powłoki na prawach użytkownika root. Jednym ze sposobów na rozwiązanie takiego problemu jest po prostu zalogowanie się na konto użytkownika root na innej, wirtualnej konsoli. Innym rozwiązaniem jest uruchomienie nowej powłoki użytkownika root przy użyciu polecenia `sudo -s`. Warto jeszcze tutaj wspomnieć, że nową powłokę możesz uruchomić również w bardziej tradycyjny sposób, korzystając z polecenia `su`. Wykonanie polecenia `su` bez żadnych opcji ani argumentów powoduje uruchomienie nowej powłoki na prawach użytkownika root, ale bez przetwarzania wszystkich normalnie wykonywanych podczas procesu logowania plików startowych i konfiguracyjnych powłoki. Jeżeli jednak do polecenia `su` dołączysz opcję `-`, spowoduje to, że nowa powłoka będzie uruchamiana w trybie pełnej sesji logowania, czyli będzie działała dokładnie tak, jakbyś właśnie przed momentem zalogował się do powłoki na konto użytkownika root. Polecenie `su` może również przyjmować jako argument nazwę konta użytkownika; domyślną wartością argumentu jest `root`, ale jeżeli chcesz uruchomić nową powłokę na prawach innego użytkownika, powinieneś wykonać polecenie `su - nazwakontaużytkownika`. Niezależnie od tego, jaki sposób uruchomienia powłoki wybierzesz, zostaniesz zapytany o hasło danego użytkownika, na którego konto teraz się przełączasz. Ponieważ w zasadzie nie powinieneś znać haseł dostępu do kont innych użytkowników, a poza tym być może nie znasz hasła konta użytkownika root, to możesz ominąć to wymaganie, korzystając z polecenia `sudo` (jeżeli pracujesz na koncie użytkownika root — a właśnie to uzyskujesz, korzystając z polecenia `sudo` — to polecenie `su` nie będzie Cię już prosiło o podawanie hasła dostępu do innych kont). Przyjrzyj się przykładowej sesji przedstawionej poniżej:

```
helion@debian:~$ sudo su -
root@debian:~# whoami
root
root@debian:~# exit
helion@debian:~$ sudo su - test
test@debian:~$ whoami
test
test@debian:~$ exit
helion@debian:~$
```

Przebieg sesji powinien być chyba dla Ciebie oczywisty. Pierwsze wywołanie polecenia `su` spowodowało uruchomienie pełnej sesji powłoki dla użytkownika root. Następnie sprawdziliśmy, czy rzeczywiście pracujemy na koncie tego użytkownika, i powróciliśmy do naszej bieżącej sesji powłoki. Kolejne wywołanie polecenia `su` spowodowało uruchomienie pełnej sesji powłoki dla użytkownika *test*, po czym ponownie sprawdziliśmy,

na koncie jakiego użytkownika aktualnie pracujemy, i powróciliśmy do naszej bieżącej sesji powłoki. Zwróć uwagę, że w żadnym momencie wywoływane polecenie `su` (wywoływane poprzez polecenie `sudo`) nie prosiło Cię o podawanie hasła dostępu dla kont innych użytkowników.

Dodawanie nowych kont użytkowników oraz grup

Jeżeli jesteś jedynym użytkownikiem danego systemu, to zazwyczaj nie będziesz miał potrzeby tworzenia zbyt wielu dodatkowych kont użytkowników. Konta użytkowników systemowych (wykorzystywane przez stale działające demony usług bądź inne, wydzielone podsystemy systemu Linux) są zazwyczaj tworzone przez odpowiednie skrypty instalacyjne danego pakietu oprogramowania. Nie zmienia to jednak w niczym faktu, że nawet jeżeli jesteś jedynym użytkownikiem danego systemu, to od czasu do czasu możesz mieć potrzebę utworzenia tymczasowego, dodatkowego konta przeznaczonego do testowania np. nowych ustawień czy aplikacji. Ponieważ dostęp do niektórych urządzeń sprzętowych, takich jak karta dźwiękowa czy drukarki, jest często kontrolowany poprzez specjalne grupy, takie jak np. `audio` czy `lpadmin`, oraz ponieważ zmiana praw własności węzłów urządzeń przechowywanych w katalogu */dev/* po to, abyś mógł korzystać z karty dźwiękowej czy też drukarki ze swojego normalnego konta użytkownika, zdecydowanie nie jest zalecana, to staje się najzupełniej oczywiste, że administrator takiego systemu powinien bez dwóch zdań doskonale orientować się we wszelkich zagadnieniach związanych z tworzeniem nowych kont użytkowników i zarządzaniem nimi.

Dodawanie nowego konta użytkownika przy użyciu polecenia adduser

Zastosowanie polecenia `adduser`, stanowiącego część pakietu `adduser`, jest chyba najwygodniejszym sposobem na utworzenie nowego konta użytkownika i wykonanie wszystkich niezbędnych czynności związanych z początkową konfiguracją takiego konta. Polecenie `adduser` tworzy odpowiedni wpis o danym koncie użytkownika w pliku */etc/passwd*, dodaje początkowe hasło użytkownika do pliku */etc/shadow*, tworzy osobistą grupę tego użytkownika w pliku */etc/group*, tworzy odpowiedni katalog domowy użytkownika oraz umieszcza tam szereg plików konfiguracyjnych. Spróbuj teraz utworzyć w swoim systemie nowe konto użytkownika o nazwie *testuser*, korzystając z polecenia `adduser`:

```
helion@debian:~$ sudo adduser testuser
password:
Adding user testuser...
Adding new group testuser (1001).
Adding new user testuser (1001) with group testuser.
Creating home directory /home/testuser.
Copying files from /etc/skel
Enter new UNIX password: ********
Retype new UNIX password: ********
passwd: password updated successfully
Changing the user information for testuser
Enter the new value, or press ENTER for the default
        Full Name []: Użytkownik testowy
        Room Number[]:
```

```
        Work Phone[]:
        Home Phone[]:
        Other[]:
Is the information correct? [y/N] y
helion@debian:~$
```

Hasło reprezentowane przez wytłuszczone znaki gwiazdek przedstawione w naszym przykładzie nie będzie wyświetlane na Twoim ekranie, ale mimo to powinieneś wpisać tam odpowiednie hasło. Systemy UNIX i Linux podczas wpisywania hasła przez użytkownika zazwyczaj nie wyświetlają niczego na ekranie. W zasadzie to by było na tyle — jak widać tworzenie nowego konta użytkownika przy użyciu polecenia `adduser` nie jest zadaniem specjalnie skomplikowanym. Skrypt najpierw wyświetla na ekranie trochę informacji o wykonywanych operacjach, a następnie prosi o podanie hasła dla tworzonego konta użytkownika. Na koniec skrypt prosi o podanie opcjonalnych, dodatkowych informacji na temat samego użytkownika; pamiętaj, że warto tutaj podać przynajmniej pełne imię i nazwisko użytkownika.

Od tej chwili nowy użytkownik może już logować się do systemu i pracować w interaktywnej sesji powłoki. Warto zauważyć, że w systemie Debian domyślnie dla każdego tworzonego konta użytkownika tworzona jest nowa grupa; nazwa grupy jest identyczna jak nazwa utworzonego konta. Takie rozwiązanie zapewnia, że jedynym i wyłącznym właścicielem plików danego użytkownika jest on sam; jeżeli wszyscy użytkownicy domyślnie byliby członkami jednej, domyślnej grupy użytkowników (w niektórych starszych systemach, w których takie rozwiązanie jest nadal stosowane, taka domyślna grupa użytkowników nosi nazwę `users`), to wtedy wszyscy członkowie grupy `users` byliby grupowym właścicielem wszystkich plików przez nich utworzonych.

Dodawanie nowych grup przy użyciu polecenia addgroup

Polecenie `addgroup` działa w bardzo podobny sposób, jak polecenie `adduser`; jako argument wywołania wystarczy podać nazwę tworzonej grupy.

```
helion@debian:~$ sudo addgroup testgroup
Adding group testgroup (1002)...
Done.
helion@debian:~$
```

Po wywołaniu polecenia `addgroup` została utworzona grupa `testgroup`, która w chwili obecnej nie ma żadnych członków. Więcej szczegółowych informacji na temat dodawania użytkowników do poszczególnych grup znajdziesz w podrozdziale „Zarządzanie grupami użytkowników”, znajdującym się w dalszej części rozdziału.

Usuwanie kont użytkowników i grup

Usuwanie kont użytkowników oraz grup użytkowników musi być realizowane z nieco większą ostrożnością, ponieważ w systemie plików mogą nadal znajdować się pliki, których usuwany użytkownik lub grupa są właścicielem. Co więcej, bardzo często takie pliki mogą zawierać jakieś istotne dane, które usuwany użytkownik może chcieć wcześniej zachować.

Usuwanie kont użytkowników przy użyciu polecenia deluser

Zanim rozpoczniesz proces usuwania kont użytkowników bądź usuwania grup użytkowników, powinieneś zapoznać się z kilkoma opcjami polecenia `deluser`, które mogą być bardzo pomocne w „sprzątaniu" pozostałości po usunięciu konta użytkownika: `--remove-all-files` oraz `--backup`. Łączne zastosowanie tych dwóch opcji spowoduje, że polecenie `deluser` utworzy archiwum wszystkich plików danego użytkownika, umieszczając ich kopie w pojedynczym pliku archiwum, a następnie usunie takie pliki z systemu plików. Spróbuj teraz skorzystać z polecenia `deluser` i usunąć konto użytkownika `testuser`, które dodałeś w jednej z poprzednich części rozdziału. Ponieważ użyjemy tutaj opcji `--remove-all-files` oraz `--backup`, to cały proces może potrwać nawet kilka minut.

Pamiętaj jednak, że jeżeli chcesz skorzystać z opisanych tutaj poleceń, to będziesz musiał wcześniej zainstalować pakiet `perl-modules`. Aby tego dokonać, powinieneś wykonać polecenie `sudo apt-get install perl-modules`.

Kiedy już rozwiążesz wszystkie problemy związane z instalacją niezbędnych pakietów dodatkowych, będziesz mógł wykonać następujące polecenie:

```
helion@debian:~$ sudo deluser --remove-all-files --backup testuser
Looking for files to backup/remove...
Backing up files to be removed to. ...
/bin/tar: Removing leading `/' from member names
Removing files...
Removing user testuser...
Done.
helion@debian:~$
```

Od czasu do czasu możesz otrzymać na ekranie komunikaty o błędach informujące, że nie można otworzyć niektórych plików i katalogów zlokalizowanych w katalogu */proc/*; nie przejmuj się — możesz je bezpiecznie zignorować. Katalog */proc/* jest swego rodzaju wirtualnym systemem plików, który po prostu eksportuje informacje bezpośrednio z jądra systemu. Domyślnie polecenie `deluser` tworzy archiwum usuwanych plików w bieżącym katalogu roboczym, nadając plikowi archiwum nazwę *nazwaużytkownika.tar.gz*. Przyjrzyj się teraz plikowi archiwum, jaki został utworzony podczas procesu usuwania użytkownika `testuser` w poprzednim przykładzie.

```
helion@debian:~$ ls -l testuser.tar.gz
-rw-r--r--  1 root root 1.1K 2005-11-14 22:20 testuser.tar.gz
helion@debian:~$
```

Teraz posiadasz już archiwum, które możesz w prosty sposób przekazać danemu użytkownikowi (np. takiemu, którego konto zostało usunięte z systemu); dodatkowo dzięki takiemu archiwum możesz w prosty sposób odtworzyć pliki takiego użytkownika (jeżeli zajdzie potrzeba).

Usuwanie grup użytkowników przy użyciu polecenia delgroup

Proces usuwania grup jest nieco bardziej złożony niż proces usuwania kont użytkowników. Po pierwsze i najważniejsze, do grupy, którą chcesz usunąć, mogą być nadal przypisane jakieś konta użytkowników. Jeżeli nie jesteś pewny, czy tak nie jest, możesz zastosować opcję `--only-if-empty`, która spowoduje, że jeżeli do usuwanej grupy nadal przypisane są jakieś konta użytkowników, to cała operacja zostanie anulowana. Polecenie `delgroup` nie powoduje usunięcia żadnych plików ani katalogów, których właścicielem jest usuwana grupa. Jeżeli kiedyś w przyszłości utworzysz kolejną grupę, która otrzyma identyczny, numeryczny identyfikator grupy (ang. *GID*, *Group ID*), to grupa taka automatycznie stanie się właścicielem wszystkich istniejących jeszcze do tej pory plików i katalogów, których poprzednim właścicielem była grupa usunięta wcześniej — co z punktu widzenia bezpieczeństwa systemu może stanowić poważne zagrożenie, zwłaszcza w sytuacji, kiedy przeznaczenie nowej grupy zupełnie nie wiąże się z rolą, jaką pełniła grupa usunięta. Z tego powodu przedstawiamy poniżej trzy kolejne etapy postępowania, przez które musisz zawsze przejść, kiedy masz zamiar usunąć daną grupę.

1. Usuń wszystkie konta użytkowników z grupy, którą zamierzasz usunąć.
2. Wykonaj kopię bezpieczeństwa, a następnie usuń bądź też zmień grupowego właściciela wszystkich plików i katalogów, których obecnym właścicielem jest grupa przeznaczona do usunięcia.
3. Usuń żądaną grupę przy użyciu polecenia `delgroup`.

Usuwanie kont użytkowników z danej grupy zostanie omówione w następnym podrozdziale, ale ponieważ nasza grupa testowa, `testgroup`, jest pusta, więc nie musisz się o to martwić w tym momencie. Utworzymy jednak tutaj plik testowy, którego właścicielem grupowym będzie nasza grupa `testgroup`, dzięki czemu będziemy mogli do tego zagadnienia powrócić nieco później. Dla naszego pliku testowego ustawimy odpowiednie prawa dostępu oraz ustawimy znacznik `setgid`:

```
helion@debian:~$ touch plik-testowy-grupy
helion@debian:~$ sudo chgrp testgroup plik-testowy-grupy
helion@debian:~$ sudo chmod g+xs plik-testowy-grupy
helion@debian:~$ ls -l plik-testowy-grupy
-rw-r-sr--  1 helion testgroup 0 2005-12-04 22:07 plik-testowy-grupy
helion@debian:~$
```

Mamy teraz przykładową grupę, którą będziemy chcieli usunąć. Grupa ta jest pusta (tzn. żadne konta użytkowników nie zostały do niej przypisane) oraz jest grupowym właścicielem naszego pliku testowego. Najpierw musimy zająć się plikami, których właścicielem jest grupa przeznaczona do usunięcia. Istnieje wiele sposobów podejścia do takiego zagadnienia, ale z naszego doświadczenia wynika, że najlepszym sposobem jest zmiana grupowego właściciela takich plików na grupę `root`. W późniejszym czasie właściciele plików mogą spokojnie zmienić grupowego właściciela plików na inną grupę, której są nadal członkami (oczywiście, jeżeli tylko będą chcieli wykonać taką operację). Zmiana właściciela grupowego na grupę `root` jest o tyle dobra, że nikt poza administratorami systemu nie należy do grupy `root`, stąd jest to bardzo bezpieczne rozwiązanie. Jak zapewne pamiętasz z lektury rozdziału 3., poszczególne pliki mogą mieć ustawione prawa *setuid* oraz *setgid* (należące do kategorii zaawansowanych praw dostępu),

oznaczające, że pliki wykonywalne mogą być uruchamiane tylko i wyłącznie przez użytkownika bądź grupę, którzy są ich właścicielem, niezależnie od tego, jaki użytkownik wywołuje dany plik wykonywalny. Plik, który będziesz chciał usunąć, może być plikiem wykonywalnym i do tego posiadać ustawiony bit *setgid*, dlatego bardzo istotnym zagadnieniem staje się usunięcie znacznika *setgid*, zanim zmienisz grupowego właściciela danego pliku. W przeciwnym wypadku, jeżeli ktoś będzie usiłował uruchomić taki plik, to dzięki takiemu zaniechaniu może otrzymać częściowe uprawnienia użytkownika root. Poniżej przedstawiamy odpowiednie skrypty powłoki. Umożliwią Ci odszukanie wszystkich plików, których właścicielem jest grupa *testgroup*, a następnie zmianę praw dostępu, tak aby znacznik *setgid* został usunięty (nie przejmuj się, jeżeli na tym etapie lektury książki niektóre elementy przedstawionych poniżej skryptów nie są dla Ciebie do końca zrozumiałe).

```
sudo find / -group testgroup -print0 2> /dev/null | xargs -0 sudo chmod g-s
```

Kolejnym etapem jest ponowne odszukanie tych samym plików i zmiana ich właściciela grupowego na grupę `root`. Możesz tego dokonać za pomocą nieco zmodyfikowanej wersji skryptu przedstawionego powyżej:

```
sudo find / -group testgroup -print0 2> /dev/null | xargs -0 sudo chgrp root
```

Przyjrzyj się teraz poniższemu zapisowi sesji — zobaczysz omówione powyżej elementy w działaniu:

```
helion@debian:~$ ls -l plik-testowy grupy
-rw-r-sr--  1 helion testgroup 0 2005-12-04 22:07 plik-testowy-grupy
helion@debian:~$ sudo find / -group testgroup -print0 2> /dev/null | xargs -0 sudo
chmod g-s
helion@debian:~$ ls -l plik-testowy grupy
-rw-r-xr--  1 helion testgroup 0 2005-12-04 22:07 plik-testowy-grupy
helion@debian:~$ sudo find / -group testgroup -print0 2> /dev/null | xargs -0 sudo
chgrp root
helion@debian:~$ ls -l plik-testowy grupy
-rw-r-xr--  1 helion root 0 2005-12-04 22:07 plik-testowy-grupy
helion@debian:~$
```

Teraz, kiedy już zająłeś się w odpowiedni sposób wszystkimi plikami, których właścicielem była grupa przeznaczona do usunięcia, możesz z czystym sumieniem rozpocząć procedurę usunięcia samej grupy. Aby tego dokonać, powinieneś wykonać poniższe polecenie:

```
helion@debian:~$ sudo delgroup --remove-only-if-empty testgroup
Removing group testgroup...
Done.
helion@debian:~$
```

Zarządzanie grupami użytkowników

Domyślnie w systemie Debian każdy użytkownik posiada swoją własną, indywidualną grupę, ale oprócz tego istnieje cały szereg innych, standardowych grup, które kontrolują dostęp do różnych zasobów systemowych, takich jak na przykład pliki dzienników zdarzeń czy też niektóre wybrane urządzenia sprzętowe. Co więcej, jeżeli jesteś administratorem systemu wieloużytkownikowego, to zazwyczaj będziesz chciał utworzyć odpowiednie

grupy użytkowników kontrolujące poszczególne kategorie dostępu użytkowników do systemu. W tabeli 5.1 zamieszczone zostało zestawienie niektórych wybranych standardowych grup użytkowników wraz z krótkim opisem ich przeznaczenia.

Tabela 5.1. *Wybrane grupy standardowe*

Nazwa grupy	Opis
`adm`	Do tej grupy należą zazwyczaj administratorzy systemu. Grupa `adm` kontroluje dostęp do wielu plików dzienników zdarzeń, które są dostępne tylko i wyłącznie dla użytkowników będących członkami tej grupy.
`lp`	Użytkownicy, którzy są członkami grupy `lp` (ang. *line printer*), mogą korzystać z drukarki systemowej.
`mail`	Użytkownicy będący członkami grupy `mail` uważani są za administratorów serwera poczty elektronicznej i w związku z tym są uprawnieni do wykonywania całego szeregu różnych operacji związanych z zarządzaniem tym serwerem (włączając w to możliwość czytania poczty innych użytkowników).
`dialout`	Użytkownicy z tej grupy mają możliwość inicjowania połączeń modemowych ze zdalnymi serwerami — przykładowo z serwerami dostępowymi dostawców Internetu (ang. *ISP*).
`cdrom`	Do tej grupy należą użytkownicy, którzy mają prawa bezpośredniego korzystania ze wszystkich napędów CD zainstalowanych w danym systemie, np. w celu odtwarzania zapisów muzycznych, nagrywania płyt CD-R itp.
`floppy`	Użytkownicy, którzy są członkami grupy `floppy`, mają prawa dostępu do dowolnej stacji dyskietek spośród zainstalowanych w danym systemie.
`audio`	Użytkownicy będący członkami grupy `audio` mają pełne prawa dostępu do urządzeń audio zamontowanych w danym systemie, takich jak np. karta dźwiękowa umożliwiająca odtwarzanie i zapisywanie danych audio.
`video`	Użytkownicy z tej grupy mają możliwość bezpośredniego korzystania z urządzeń wideo zainstalowanych w danym systemie, takich jak np. karta graficzna umożliwiająca wykonywanie zrzutów ekranów z takich urządzeń jak kamera wideo itp.

Posiadanie tak wielu różnych grup może się na pierwszy rzut oka wydawać grubą przesadą, niemniej jednak jeżeli wszystkie urządzenia byłyby domyślnie, w jednakowym stopniu dostępne dla wszystkich użytkowników, to w większych systemach mogłoby się zdarzyć, że np. ktoś włączył przypadkowo odtwarzanie lub, co gorsza, nagrywanie dźwięku w gabinecie prezesa firmy czy też włączył sobie podgląd kamery internetowej zainstalowanej w serwerowni i obserwował, jakie hasła wpisuje administrator na konsoli serwera.

Dodawanie użytkowników do grup jest zadaniem bardzo prostym. Aby tego dokonać, wystarczy wykonać następujące polecenie: `adduser` *`nazwaużytkownika nazwagrupy`*. Przykładowo, z pewnością chciałbyś mieć możliwość odczytywania różnych plików dzienników systemowych bezpośrednio z poziomu Twojego normalnego konta użytkownika, bez konieczności ciągłego używania poleceń takich jak `sudo` czy `su`. Aby tego dokonać, wystarczy — jak się już zapewne sam domyśliłeś — dodać nazwę Twojego konta użytkownika do grupy `adm`:

```
helion@debian:~$ sudo adduser helion adm
Adding user helion to group adm...
Done.
helion@debian:~$
```

Zwróć jednak uwagę, że nowe uprawnienia nie są od razu widoczne:

```
helion@debian:~$ groups
helion
helion@debian:~$
```

Aby członkostwo w nowych grupach przyniosło oczekiwany efekt, powinieneś się wylogować, a następnie zalogować ponownie — mimo że istnieją również inne sposoby, to jednak często wylogowanie okazuje się najbardziej efektywnym z nich, a co więcej, taki sposób postępowania będzie wymagany dla każdej sesji środowiska graficznego. Kiedy już zalogujesz się ponownie, znowu wykonaj polecenie `groups`:

```
helion@debian:~$ groups
helion  adm
helion@debian:~$
```

Aby usunąć danego użytkownika z określonej grupy, powinieneś wykonać polecenie `deluser` *`nazwaużytkownika nazwagrupy`*. Przyjrzyj się teraz zapisowi poniższej sesji:

```
helion@debian:~$ sudo deluser helion adm
Removing user helion from group adm...
Done.
helion@debian:~$ groups
helion  adm
helion@debian:~$
```

Zwróć uwagę, że podobnie jak to miało miejsce podczas dodawania użytkownika do grupy, usunięcie z grupy będzie „widoczne” dopiero po zakończeniu i ponownym uruchomieniu bieżącej sesji powłoki. Aby nowe przydziały grup stały się aktywne, również wszystkie bieżące procesy drugoplanowe muszą zostać zakończone i uruchomione na nowo.

Podstawowe informacje o połączeniach sieciowych

W rozdziale 2., kiedy omawialiśmy sposób instalacji systemu Debian, miałeś możliwość podłączenia swojego komputera do sieci Internet bądź też do swojej sieci lokalnej. Skrócony sposób, w jaki zostały tam potraktowane zagadnienia związane z połączeniami sieciowymi, oczywiście nie wyczerpał tego tematu, dlatego teraz zajmiemy się nim ponownie.

W niniejszym podrozdziale omówimy zagadnienia związane z wykorzystaniem wielu niskopoziomowych narzędzi sieciowych pozwalających na funkcjonowanie w sieci, do której jesteś podłączony. Postaramy się przybliżyć Ci tematykę powiązań systemu Debian z takimi sieciami oraz omówić, w jaki sposób system ten z nimi współpracuje.

W niniejszym rozdziale niestety nie będziemy w stanie omówić szczegółowo wszystkich zagadnień związanych z połączeniami sieciowymi w systemie Linux — w zasadzie przedstawimy jedynie nieco powierzchowny przegląd najważniejszych zagadnień. Raczej nietrudno sobie wyobrazić, że tematyka połączeń sieciowych to ogromny temat, zwłaszcza jeżeli weźmie się pod uwagę złożoność spraw związanych ze współpracą pomiędzy różnymi rodzajami sieci komputerowych. Więcej szczegółowych informacji na temat współpracy systemu Linux z sieciami komputerowymi znajdziesz w dokumencie *Linux Networking HOWTO*, dostępnym na stronie *http://www.tldp.org/HOWTO/Net-HOWTO/*.

IP — Internet Protocol

Internet (akronim angielskiego określenia *inter-network*) jest po prostu ogromnym zbiorem mniejszych sieci, połączonych ze sobą w wielu miejscach i na wiele sposobów. Takie mniejsze sieci, czasami nazywane również sieciami *intranet*, są zazwyczaj od siebie dobrze odseparowane, główne ze względu na to, że każdą siecią zarządza inny administrator — Ty jesteś odpowiedzialny za sprawne i bezpieczne funkcjonowanie Twojej sieci, Twój dostawca Internetu jest odpowiedzialny za działanie jego sieci, poszczególne firmy są odpowiedzialne za ich sieci itd. Takie rozproszenie i separacja poszczególnych sieci oznacza, że muszą istnieć jakieś podstawowe uzgodnienia wyznaczające sposób, w jaki poszczególne sieci ze sobą współpracują — i właśnie takie uzgodnienia nazywamy, ogólnie rzecz biorąc, *protokołem*. Protokół ściśle definiuje sposób, w jaki komputer A będzie się komunikował z komputerem B.

Protokół TCP/IP

Cała komunikacja w sieci Internet oraz w nowoczesnych sieciach intranet odbywa się według zasad zdefiniowanych w specyfikacji protokołu IP (ang. *Internet Protocol*). Sama specyfikacja protokołu IP jest bardzo rozbudowanym i szczegółowym dokumentem. Podstawową jednostką transmisji danych wykorzystywaną przez protokół IP jest *pakiet*. Każdy strumień danych jest dzielony na małe części (zwane właśnie pakietami), które następnie są wysyłane poprzez sieć do miejsca przeznaczenia. W podstawowym pakiecie protokołu IP znajdują się informacje na temat nadawcy pakietu (czyli inaczej informacje o tym, skąd pochodzi dany pakiet), informacje na temat odbiorcy pakietu (czyli inaczej mówiąc informacje o tym, dokąd podróżuje dany pakiet w sieci) oraz „ładunek" pakietu, czyli inaczej mówiąc dane, jakie przenosi on od nadawcy do odbiorcy. Ładunek pakietu może być niemal dowolny (tak długo, jak długo jego format pozostaje w zgodzie z wymogami definicji protokołu IP) — w pakiecie IP mogą znajdować się dowolne dane. Istnieją jednak również inne protokoły i standardy transmisji, które bardziej szczegółowo definiują zawartość poszczególnych pakietów — przykładem takiego protokołu jest TCP (ang. *Transmission Control Protocol*).

Sam protokół IP jest relatywnie prostym protokołem komunikacyjnym i nie posiada zaimplementowanych żadnych mechanizmów gwarantujących, że wysłany dany pakiet dotrze do miejsca przeznaczenia. Protokół TCP, a w zasadzie TCP/IP (ang. *TCP over IP*) jest w obecnych czasach najbardziej rozpowszechnionym protokołem sieciowym, stąd niemal każda usługa sieciowa, z jakiej korzystasz na co dzień, jest oparta właśnie na tym protokole. Zastosowanie protokołu TCP/IP pozwala na sprawną i pewną wymianę danych pomiędzy komputerami połączonymi poprzez sieć Internet.

Adresy sieciowe

Każdy komputer podłączony do sieci IP posiada swój własny, unikalny adres IP, aczkolwiek pełna konfiguracja połączenia sieciowego to coś znacznie więcej niż tylko prosty adres IP. Pomimo iż w każdym wysłanym w sieć pakiecie znajduje się adres IP nadawcy oraz adres IP odbiorcy, to jednak zlokalizowanie miejsca w sieci, do którego powinien zawędrować taki pakiet, jest zagadnieniem nieco bardziej złożonym. Każdy komputer podłączony do sieci musi posiadać zdefiniowane trzy bardzo ważne elementy konfiguracji sieciowej: adres IP (ang. *IP address*), maskę podsieci (ang. *netmask*) oraz bramę domyślną (ang. *gateway*). Maska podsieci pozwala danej maszynie na wyróżnienie adresów komputerów znajdujących się w tej samej sieci; jeżeli Twój komputer jest częścią sieci lokalnej LAN (ang. *Local Area Network*), to może się bezpośrednio komunikować z innymi komputerami znajdującymi się w tym samym segmencie sieci (inaczej mówiąc, komunikacja pomiędzy nimi może się odbywać bez żadnego „pośrednika"). Jeżeli jednak adresat pakietu znajduje się poza siecią lokalną, to odpowiednie pakiety danych muszą być najpierw przesłane do *routera* czy też bramy, która skieruje taki pakiet do kolejnego routera i tak dalej, aż dany pakiet danych zostanie dostarczony do komputera docelowego.

Adresy IP, a ściślej mówiąc, adresy IPv4, są zapisywane w tzw. 4-bajtowej notacji kropkowej (ang. *dotted quad notation*). Ta groźna skądinąd nazwa oznacza po prostu, że każdy adres IP składa się z czterech liczb z zakresu od 0 do 255, oddzielonych od siebie kropkami. Przykładowo, zapis `1.2.3.4` jest prawidłowo podanym adresem IP.

Więcej szczegółowych informacji na temat masek podsieci znajdziesz na stronie *http://www.computerhope.com/jargon/n/netmask.htm*.

Porty

Oprócz wspomnianych już wcześniej adresów nadawcy oraz adresów odbiorcy (lub jak kto woli, adresów źródła i adresów przeznaczenia) istnieją jeszcze tzw. *porty*. Ponieważ dany komputer może oferować użytkownikom całą gamę różnego rodzaju usług sieciowych (np. serwer WWW, FTP, poczty elektronicznej i wiele innych), to bardzo ważne jest, aby każda z takich usług była dostępna poprzez ściśle określony i powszechnie znany numer portu (ang. *well-known ports*). Numer portu jest wykorzystywany w komunikacji pomiędzy komputerami łącznie z adresem IP, kiedy dany komputer usiłuje skorzystać z określonej usługi sieciowej oferowanej przez inny komputer. Przykładowo, jeżeli oglądasz zawartość ulubionej strony WWW, to Twoja przeglądarka sieciowa próbuje skontaktować się z portem 80 danego serwera, który jest domyślnym portem wykorzystywanym przez serwer WWW. Posługując się nieco uproszczoną analogią, można powiedzieć, że adres IP to coś w rodzaju pełnego adresu mieszkania czy też domu, natomiast numer portu odpowiada tutaj numerowi pokoju. W systemie GNU/Linux definicje wszystkich powszechnie znanych portów i odpowiadających im usług sieciowych możesz znaleźć w pliku */etc/services*.

Przypisywanie nazwy komputera do jego adresu

Jak doskonale wiesz, komputery znają tylko i wyłącznie liczby — nawet litery rozpoznają jedynie wtedy, kiedy są wewnętrznie reprezentowane przez odpowiednie liczby. Takie twierdzenie dotyczy wszystkich spraw, o których „wie" Twój komputer — włączając w to jego „znajomości" z innymi komputerami. Przykładowo, jeżeli korzystając z przeglądarki sieciowej, chcesz zajrzeć na stronę *http://www.debian.org/*, to Twój komputer z pewnością nie będzie od razu, w jakiś magiczny sposób „wiedział", z którym komputerem w sieci Internet powinien się skontaktować. Każdy człowiek może po prostu spojrzeć na podany powyżej adres i… po prostu wie, że chcesz zajrzeć na serwer WWW projektu Debian — jednak dla Twojego komputera zorientowanie się, który z milionów komputerów podłączonych do sieci Internet kryje się za adresem *http://www.debian.org*, jest zadaniem daleko bardziej złożonym.

Rozwiązywanie nazw hostów

Całą zabawę rozpoczniemy od zainstalowania pakietu `host`. W tym celu powinieneś wykonać polecenie `sudo apt-get install host`. Następnie powinieneś sprawdzić, czy działa ono poprawnie, wykonując przedstawione poniżej polecenie:

```
helion@debian:~$ host www.debian.org
www.debian.org          A       192.25.206.10
helion@debian:~$
```

W terminologii sieciowej każdy węzeł sieci — czyli inaczej mówiąc każdy komputer — jest nazywany *hostem*. Polecenie `host` pozwala Ci na powtórzenie tych wszystkich operacji, jakie musi wykonać Twój komputer za każdym razem, kiedy chce się skontaktować z innym komputerem, którego adres podałeś w postaci nazwy. W naszym przypadku, kiedy próbowaliśmy się skontaktować z komputerem o nazwie *www.debian.org*, polecenie `host` ujawniło nam, że za tą nazwą tak naprawdę kryje się komputer o adresie `192.25.206.10` — jest to numeryczny adres serwera WWW projektu Debian, a mówiąc ściślej, jego adres IP.

Proces zamiany nazw hostów z postaci tekstowej, przyjaznej dla użytkownika, na adresy numeryczne, bardziej zrozumiałe dla komputera, jest nazywany rozwiązywaniem nazw hostów (ang. *hostname resolution*).

Więcej szczegółowych informacji na temat systemu nazw domen DNS (ang. *Domain Name System*) znajdziesz w rozdziale 17. niniejszej książki. Omówimy tam nie tylko zagadnienia związane z zamianą nazw komputerów na odpowiadające im adresy IP, ale również powiemy, w jaki sposób możesz zainstalować i skonfigurować swój własny serwer DNS.

Pliki konfiguracyjne mechanizmu rozwiązywania nazw

Spróbujmy teraz zorientować się, jakie pliki konfiguracyjne zlokalizowane w Twoim systemie mają wpływ na funkcjonowanie mechanizmu rozwiązywania nazw. W tabeli 5.2 przedstawiono krótki opis trzech najbardziej standardowych plików konfiguracyjnych, jakie możesz znaleźć w niemal każdym systemie GNU/Linux, włączając w to system Debian.

Tabela 5.2. *Pliki konfiguracyjne mechanizmu rozwiązywania nazw*

Nazwa pliku konfiguracyjnego	Opis
/etc/nsswitch.conf	Plik konfiguracyjny usługi NSS (ang. *Name Service Switch*).
/etc/resolv.conf	Plik konfiguracyjny usługi rozwiązywania nazw.
/etc/hosts	Lokalna baza danych nazw hostów i odpowiadających im adresów IP.

Kiedy Twój komputer próbuje skontaktować się z określonym hostem — na przykład *www.debian.org* — to w pierwszej kolejności sięga do pliku */etc/nsswitch.conf*. Format tego pliku jest bardzo prosty; aby się o tym przekonać, powinieneś otworzyć ten plik w swoim ulubionym edytorze tekstu. Pomimo iż ten plik konfiguracyjny w zasadzie opisuje zachowanie wszystkich rodzajów konwersji typu nazwa na liczbę, to jednak teraz skoncentrujemy się wyłącznie na rozwiązywaniu nazw hostów. Po otwarciu pliku */etc/nsswitch.conf* do edycji powinieneś odszukać w nim wiersz przedstawiony poniżej:

```
hosts:          files dns
```

Taka postać wiersza informuje wszystkie aplikacje, że jeżeli chcą rozwiązać nazwę danego hosta, to w pierwszej kolejności powinny sprawdzić bazę hostów zapisanych w odpowiednim pliku (słowo kluczowe `files`). Tym „tajemniczym" plikiem jest */etc/hosts*. Ma on postać zwykłego pliku tekstowego, w którym zapisane są statyczne definicje par składających się z nazwy danego hosta i odpowiadającego mu adresu IP.

Tradycja korzystania z pliku */etc/hosts* jest bardzo, bardzo stara. Wystarczy powiedzieć, że już całe dziesięciolecia temu, zanim w ogóle system DNS został opracowany i zaimplementowany, każdy komputer podłączony do sieci posiadał swój odpowiedni wpis w plikach */etc/hosts* pozostałych komputerów. Jeżeli odpowiednia informacja o nazwie i adresie IP jakiegoś komputera nie znalazła się w tym pliku, to siłą rzeczy nie byłeś w stanie się z takim komputerem skontaktować po nazwie. Tyle historia — nie zmienia to jednak w niczym faktu, że niezależnie od zaawansowanego wieku tego pliku-staruszka, jest on nadal używany w niemal każdym komputerze, który jest podłączony do sieci. Wynika to z prostej potrzeby, że niektóre nazwy komputerów i odpowiadające im adresy IP muszą pozostać dla systemu znane, nawet jeżeli awarii ulegnie serwer DNS i nie będziesz w stanie dynamicznie rozwiązywać nazw hostów. Szczególnym przypadkiem zastosowania tego pliku jest zdefiniowana w nim nazwa własna Twojego komputera, która jest niezbędna do prawidłowego funkcjonowania komputera nawet na bardzo elementarnym poziomie.

Przyjrzyjmy się teraz zawartości pliku */etc/hosts*. Zapewne zarówno w Twoim pliku, jak i w naszej wersji tego pliku na samym początku znajduje się następujący wiersz:

```
127.0.0.1           localhost
```

Jak już wspominaliśmy wcześniej, format tego pliku jest bardzo prosty — pierwszy element to po prostu adres IP danego hosta, po którym następuje jedna lub więcej przypisanych mu nazw. Na naszym przykładzie widoczna jest tylko jedna nazwa — `localhost`. Jest to bardzo specjalna nazwa, która zawsze, w każdej sytuacji reprezentuje Twój własny, lokalny komputer. Nawet w sytuacji, kiedy nie masz dostępu ani do sieci Internet, ani do żadnej innej sieci lokalnej, Twój komputer z zainstalowanym systemem GNU/Linux

jest maszyną wyposażoną w znakomity, w pełni funkcjonalny i gotowy do pracy w sieci system operacyjny — nawet poszczególne podzespoły tego komputera mogą być adresowane jako niezależne elementy wewnętrznej sieci komputera. Warto tutaj jeszcze tylko zauważyć, że nazwa `localhost` jest zawsze powiązana z odpowiadającym jej adresem IP `127.0.0.1`.

Ponieważ `localhost` jest zdefiniowany w pliku */etc/hosts*, to kiedy dana aplikacja chce się skontaktować z komputerem o nazwie `localhost`, nie musi za każdym razem poszukiwać odpowiadającego mu adresu IP poprzez sieciową usługę DNS (która tak naprawdę w takiej sytuacji wcale nie musi być dostępna).

Jeżeli jednak nazwa hosta, którą usiłujesz rozwiązać, nie jest zdefiniowana w pliku */etc/hosts*, to wykorzystywane jest drugie słowo kluczowe z pliku */etc/nsswitch.conf*. W naszym systemie jest to słowo `dns`, które powoduje, że do rozwiązywania nazw hostów wykorzystywany jest DNS (ang. *Domain Name System*) — a tutaj do głosu dochodzi nasz kolejny plik konfiguracyjny: */etc/resolv.conf*. I uwaga — między zawartością Twojego a naszego pliku */etc/resolv.conf* mogą być znaczne różnice, jako że nasze komputery z pewnością znajdują się w różnych sieciach. Podobnie jak w poprzednim przypadku, powinieneś teraz otworzyć ten plik do edycji w swoim ulubionym edytorze tekstu. Powinieneś znaleźć tam jeden lub więcej wierszy podobnych (ale niekoniecznie identycznych) z przedstawionymi poniżej:

```
nameserver   10.0.0.1
nameserver   10.0.0.2
```

Kiedy Twój komputer otrzymuje przypisany do niego adres IP (niezależnie od tego, czy jest to dynamiczny adres IP przyznawany automatycznie, poprzez serwer DHCP, czy też statyczny adres IP nadawany ręcznie przez administratora systemu), to otrzymuje również adresy IP serwerów nazw, z których powinien korzystać — i w zasadzie do tego sprowadza się zawartość poszczególnych wierszy pliku */etc/resolv.conf*. Przykładowo, jeżeli chcemy się skontaktować z serwerem *www.debian.org*, to nasz komputer w pierwszej kolejności sprawdza, czy odpowiedni wpis znajduje się w pliku */etc/hosts*. Jeżeli nie, to usiłuje uzyskać odpowiedni adres IP od serwera (bądź serwerów) DNS, zdefiniowanych w pliku */etc/resolv.conf* — w naszym przypadku są to serwery o adresach `10.0.0.1` oraz `10.0.0.2`.

W pliku */etc/resolv.conf* możesz spotkać również jeszcze inny wpis: wiersz `search`. Przed chwilą, dla uproszczenia powiedzieliśmy, że plik */etc/resolv.conf* jest wykorzystywany w sytuacji, kiedy usiłujesz rozwiązać nazwę hosta, która nie została uprzednio zdefiniowana w pliku */etc/hosts*. To nie jest do końca prawda. Wiersze rozpoczynające się od słowa kluczowego `nameserver` są wykorzystywane tylko i wyłącznie wtedy, kiedy usiłujesz rozwiązać nazwę hosta poprzez serwer DNS, natomiast jeżeli wiersz `search` został dopisany do pliku */etc/resolv.conf*, to jest on wykorzystywany zawsze. Poniżej przedstawiamy przykład takiego wpisu (podobnie jak poprzednio, może on się znacznie różnić od tego, co znajdziesz w swojej wersji pliku).

```
search   debian.org
```

Słowo kluczowe `search` informuje aplikację, że jeżeli odpowiednia nazwa hosta w ogóle nie zostanie odnaleziona (ani poprzez plik */etc/hosts*, ani poprzez serwery DNS), to powinna ona ponownie rozpocząć poszukiwania, tym razem jednak poszukując hosta, którego

nazwa została poszerzona o nazwę domeny (bądź kolejnych domen), zdefiniowaną w wierszu `search` i dodaną jako przyrostek nazwy hosta. Przyjrzyj się teraz poniższemu zapisowi sesji powłoki:

```
helion@debian:~$ cat /etc/resolv.conf
nameserver   24.153.23.66
nameserver   24.153.22.67
helion@debian:~$ host www
www does not exist, try again
helion@debian:~$ editor /etc/resolv.conf
(...teraz dodaj do pliku wiersz: search  debian.org)
helion@debian:~$ cat /etc/resolv.conf
search  debian.org
nameserver   24.153.23.66
nameserver   24.153.22.67
helion@debian:~$ host www
www.debian.org          A       192.25.206.10
helion@debian:~$
```

Jak widać, bez wiersza `search` Twój komputer nie był w stanie prawidłowo rozwiązać nazwy hosta `www` — co nie powinno być raczej dla nikogo zaskoczeniem; w sieci Internet nie istnieje żaden komputer, którego pełna nazwa brzmiałaby po prostu `www`. Jeżeli teraz dodaliśmy do pliku */etc/resolv.conf* dodatkowy wiersz `search debian.org`, to aplikacja próbująca rozwiązać nazwę najpierw spróbuje znaleźć ją „tradycyjnymi" niejako metodami (poprzez plik */etc/hosts* i odpytanie serwera DNS), a następnie spróbuje odnaleźć właściwy host, dodając do jego nazwy `www` nazwę domeny podaną po słowie kluczowym `search`. Krótko mówiąc, w pierwszym przebiegu aplikacja usiłuje rozwiązać nazwę `www` (co jej się nie udaje), a następnie dodaje do nazwy hosta nazwę domeny i usiłuje rozwiązać nazwę `www.debian.org` — co tym razem zostaje uwieńczone powodzeniem.

W pojedynczym wierszu zawierającym słowo kluczowe `search` możesz zamieszczać cały szereg nazw domen; będą one kolejno wybierane i testowane dopóty, dopóki któraś z prób nie zakończy się pomyślnie, bądź też żadna z nazw domen nie przyniesie oczekiwanego rozwiązania.

Interfejsy sieciowe

Jak zapewne pamiętasz, podczas procesu instalacji systemu Debian przechodziłeś przez proces konfiguracji połączenia sieciowego. Proces ten polegał w głównej mierze na odpowiedniej konfiguracji *interfejsu sieciowego*, co jest żargonowym określeniem całej klasy urządzeń systemu Linux pozwalających na wzajemną komunikację z innymi komputerami. Nietrudno się domyślić, że umiejętności sprawdzania, testowania, konfiguracji oraz naprawiania urządzeń sieciowych jest niezbędna do prawidłowego zarządzania mechanizmami komunikacji sieciowej. Aby przekonać się, jakie interfejsy sieciowe są dostępne na Twoim komputerze, powinieneś wykonać polecenie `sudo ifconfig`.

```
helion@debian:~$ sudo ifconfig
eth0      Link encap:Ethernet  HWaddr 00:00:21:6D:4B:BE
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3425824 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2140112 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 txqueuelen:1000
          RX bytes:672077051 (640.9 MiB)  TX bytes:1358539190 (1.2 GiB)
          Interrupt:10 Base address:0xfc80

eth1      Link encap:Ethernet  HWaddr 00:80:C8:F9:F6:AB
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2465422 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2009695 errors:0 dropped:0 overruns:0 carrier:0
          collisions:3530 txqueuelen:1000
          RX bytes:1778125589 (1.6 GiB)  TX bytes:260135783 (248.0 MiB)
          Interrupt:9 Base address:0xfc00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING MTU:16436  Metric:1
          RX packets:12627 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12627 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1252184 (1.1 MiB)  TX bytes: 1252184 (1.1 MiB)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:65.93.139.76  P-t-P:64.230.254.205 Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1492  Metric:1
          RX packets:2405279 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1950198 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:1721557756 (1.6 GiB)  TX bytes: 215446634 (205.4 MiB)

helion@debian:~$
```

Wyniki działania tego polecenia na Twoim komputerze niemal na pewno będą zupełnie inne, ponieważ nasze komputery znajdują się w zupełnie innych sieciach. W naszym przypadku są to cztery interfejsy sieciowe: `eth0`, `eth1`, `lo` oraz `ppp0`. Na Twoim komputerze również powinieneś znaleźć interfejs `lo`; jest to tzw. interfejs pętli zwrotnej (ang. *loopback interface*), który jest wirtualnym interfejsem wykorzystywanym przez różne mechanizmy systemu operacyjnego do wzajemnej komunikacji. Twoje urządzenie pętli zwrotnej, `loopback`, zawsze wskazuje z powrotem na Twój komputer. Powinieneś również posiadać interfejs `eth0` (interfejs Ethernet) lub interfejs `ppp0` (interfejs obsługujący protokół *Point-to-Point*), natomiast możesz nie mieć interfejsu `eth1` (który jest po prostu drugim interfejsem Ethernet). Jak widać, wyniki działania polecenia `ifconfig` dają całkiem sporą ilość informacji, aczkolwiek tak naprawdę w chwili obecnej będą Cię interesowały tylko po dwa pierwsze wiersze dla każdego interfejsu, opisujące typ poszczególnych interfejsów oraz ich podstawową konfigurację.

Interfejsy sieci Ethernet

Najbardziej rozpowszechnionym rodzajem interfejsu sieciowego we współczesnych komputerach jest interfejs sieci *Ethernet*. Standard Ethernet jest w zasadzie mechanizmem enkapsulacji przesyłanych danych i nie posiada szczegółowych definicji czy wymagań co do okablowania sygnałowego samego interfejsu. Zdecydowana większość urządzeń sieciowych Ethernet wykorzystuje jednak proste okablowanie nazywane UTP (ang. *Unshielded Twisted Pair*), czyli inaczej okablowanie mające postać nieekranowanej skrętki.

Jak pamiętasz, na naszym ostatnim przykładzie mieliśmy dwa interfejsy Ethernet — dla przypomnienia zamieszczamy poniżej najbardziej interesujące wiersze konfiguracji dla każdego z nich:

```
eth0      Link encap:Ethernet  HWaddr 00:00:21:6D:4B:BE
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0

eth1      Link encap:Ethernet  HWaddr 00:80:C8:F9:F6:AB
```

Zwróć uwagę, że tylko jeden z interfejsów posiada wiersz konfiguracji rozpoczynający się od słów kluczowych `inet addr`. Oznacza to w praktyce tyle, że tylko interfejs `eth0` może być wykorzystywany bezpośrednio do komunikacji ze zdalnymi komputerami (ponieważ do takiej komunikacji wymagane jest posiadanie ważnego adresu IP). Pierwsze pole tego wiersza, `inet addr`, zawiera adres IP danego interfejsu. Za każdym razem, kiedy nasz komputer wysyła pakiet danych do innego komputera w sieci, to ten właśnie adres IP jest wpisywany jako adres nadawcy pakietu, czyli jako adres źródłowy pakietu. Oprócz pola `inet addr` znajdziesz tutaj również pole o nazwie `Bcast`. Jest to pole definiujące tzw. adres rozgłoszeniowy (ang. *broadcast address*); dowolny pakiet wysłany na adres rozgłoszeniowy sieci będzie rozesłany do wszystkich komputerów danej sieci lokalnej. Wreszcie ostatnie pole, `Mask`, które wykorzystywane jest to wyznaczenia swego rodzaju granic sieci lokalnej. W naszym przypadku mamy maskę sieci `255.255.255.0` oraz adres IP `192.168.1.1`, a zatem wynika stąd, że nasza sieć lokalna, do której podłączony jest nasz komputer, składa się (bądź może się składać) z komputerów o adresach od `192.168.1.0` do `192.168.1.255`.

Każda karta sieciowa Ethernet posiada swój unikalny adres sprzętowy (ang. *hardware address*) zwany inaczej adresem MAC karty (ang. *MAC address*; *MAC — Media Access Control*). W zapisie naszej poprzedniej sesji powłoki możesz go odnaleźć w polu oznaczonym `HWaddr`. Adres sprzętowy MAC jest wykorzystywany wyłącznie w komunikacji w obrębie *lokalnego segmentu sieci Ethernet*. Za każdym razem, kiedy pakiety danych są przekazywane dalej, za pośrednictwem lokalnego routera czy też bramy domyślnej, urządzenia takie przejmują prawa „własności" takich pakietów i wysyłają je już dalej w świat ze swoim adresem sprzętowym MAC w nagłówku — i tak kolejno aż do momentu dostarczenia pakietu do miejsca przeznaczenia.

W systemie Debian GNU/Linux poszczególne interfejsy sieci Ethernet są konfigurowane poprzez odpowiednie wpisy w pliku */etc/network/interfaces*. Więcej szczegółowych informacji na ten temat znajdziesz na stronach podręcznika `man`, dostępnych po wykonaniu polecenia `man 5 interfaces`.

Interfejsy PPP

Interfejsy PPP (ang. *Point-to-Point Protocol*) są zazwyczaj wykorzystywane do komunikacji poprzez modem (połączenia wybierane typu *dial-up*), aczkolwiek są również bardzo często wykorzystywane przez dostawców sieci Internet (ang. *ISP*) oferujących połączenia typu DSL. Nasz przykładowy interfejs PPP ma następującą konfigurację:

```
ppp0      Link encap:Point-to-Point Protocol
          inet addr:65.93.139.76  P-t-P:64.230.254.205 Mask:255.255.255.255
```

Interfejsy PPP są ogólnie rzecz biorąc o wiele mniej złożone niż interfejsy Ethernet, ponieważ zdecydowana większość połączeń realizowanych przez takie interfejsy to połączenia typu punkt-punkt (ang. *point-to-point*). Oznacza to, że dla takich połączeń nie istnieje pojęcie sieci lokalnej; wszystkie pakiety są po prostu przesyłane na drugi koniec łącza (na adres znajdujący się w polu `P-t-P`, którym może być np. Twój ISP) i tam przetwarzane. Jak widać na przykładzie, adres naszego interfejsu PPP to `65.93.139.76`, natomiast adres komputera znajdującego się po drugiej stronie łącza to `64.230.254.205`.

W systemie Debian interfejsy PPP są konfigurowane poprzez odpowiednie pliki znajdujące się w katalogu */etc/ppp/peers/*. Każde utworzone połączenie typu PPP będzie reprezentowane przez odpowiedni plik konfiguracyjny utworzony w tym katalogu, noszący nazwę identyczną jak dane połączenie. Aby uruchomić wybrane połączenie PPP, powinieneś wykonać polecenie `pon` *`nazwapołączenia`*; aby je zakończyć, wykonaj polecenie `poff` *`nazwapołączenia`*. Jeżeli łączysz się z siecią Internet poprzez połączenie DSL PPP, to podczas instalacji urządzeń DSL zostanie utworzony plik konfiguracyjny */etc/ppp/peers/dsl-provider* — aby zatem uruchomić takie połączenie, powinieneś wykonać polecenie `pon dsl-provider`.

Podsumowanie

W niniejszym rozdziale skupiliśmy się głównie na zagadnieniach związanych z zarządzaniem kontami użytkowników i grupami użytkowników, ale za to tematy te zostały ujęte w szerokim kontekście dobrego zarządzania systemem oraz utrzymywania bezpieczeństwa. Po zakończeniu lektury tego rozdziału powinieneś mieć w swoim arsenale wygodny i bezpieczny interfejs pozwalający na korzystanie z uprawnień użytkownika root z poziomu Twojego normalnego konta użytkownika. Powinieneś również być w stanie tworzyć nowe i usuwać istniejące konta użytkowników i grupy użytkowników, a także zarządzać nimi. Lektura tego rozdziału powinna również dać Ci pewną podstawową wiedzę na temat połączeń sieciowych i sposobów ich działania w systemie Linux; powinieneś być również w stanie zrozumieć, w jaki sposób Twój komputer komunikuje się z innymi komputerami poprzez sieć, co będzie dla Ciebie znakomitym fundamentem do dalszego pogłębiania wiedzy na ten temat w dalszej części książki.