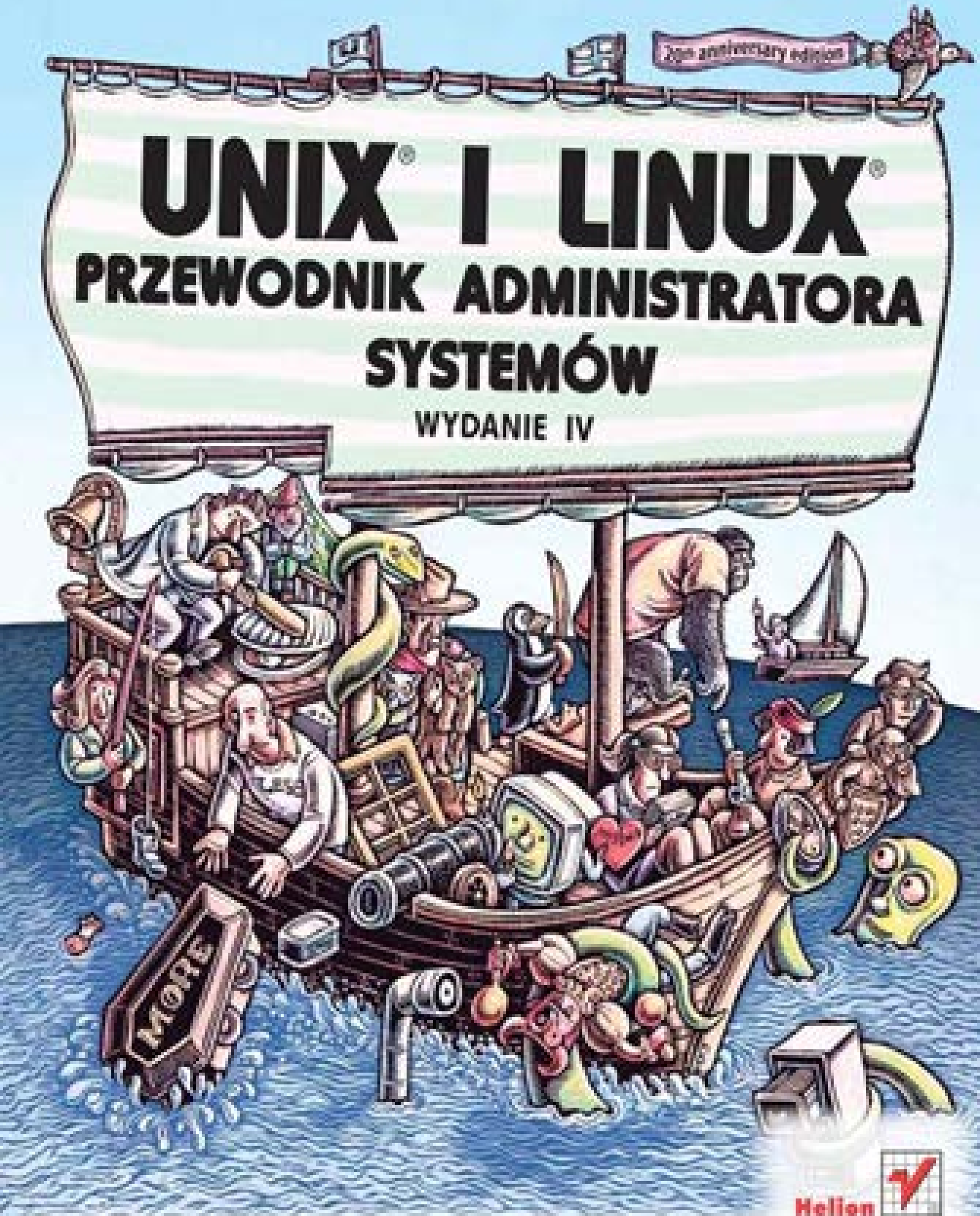
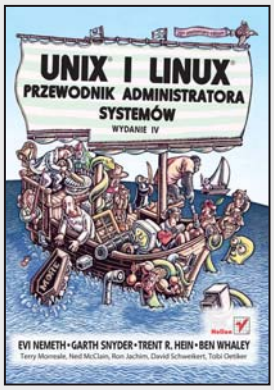
UNIX® I LINUX®
PRZEWODNIK ADMINISTRATORA SYSTEMÓW
WYDANIE IV



Helion

EVI NEMETH • GARTH SNYDER • TRENT R. HEIN • BEN WHALEY
Terry Morreale, Ned McClain, Ron Jachim, David Schweikert, Tobi Oetiker

Unix i Linux. Przewodnik administratora systemów. Wydanie IV



Autorzy: Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley
Tłumaczenie: Marek Pętlicki, Leszek Sagalara
ISBN: 978-83-246-2968-8
Tytuł oryginału: UNIX and Linux System Administration Handbook (4th Edition)
Format: 172×245, stron: 1552

Najobszerniejszy przewodnik po świecie systemów Unix i Linux!

- Jakie obowiązki ciążą na administratorze?
- Jak zarządzać użytkownikami?
- Jak poprawić wydajność systemu?

Systemy Unix i Linux wykorzystywane są wszędzie tam, gdzie wymagana jest najwyższa niezawodność, wydajność i elastyczność. Ich potencjał, możliwości oraz odporność na niesprzyjające otoczenie sprawiły, że są dominującymi systemami operacyjnymi w zastosowaniach sieciowych. Jeżeli jednak systemy te kojarzą Ci się wyłącznie z czarnym ekranem konsoli oraz trybem tekstowym, jesteś w błędzie. Już od dawna systematycznie zwiększają one swój udział w zastosowaniach domowych. Co w sobie kryją? Jak nimi administrować i jak zwiększać ich wydajność?

Na te i setki innych pytań odpowiada ten obszerny przewodnik. Autorzy rozłożyli na czynniki pierwsze i omówili każde zagadnienie, z którym możesz mieć styczność w codziennej pracy. Po lekturze tej książki będziesz posiadał obszerną wiedzę w tej materii. Instalacja systemu, skrypty powłoki, kontrolowanie procesów, konfiguracja uprawnień to zadania, które nie będą Ci sprawiały żadnych problemów. Ponadto dowiesz się, jak zarządzać użytkownikami, przestrzenią dyskową, zadaniami okresowymi oraz backupami. Dużą uwagę powinieneś poświęcić rozdziałom omawiającym zagadnienia sieciowe. Jest pewne, że serwer, który trafi w Twoje ręce, będzie oferował wiele usług dostępnych przez sieć. Protokoły routingu, usługa DNS, firewall, serwer plików to elementy, które niejednokrotnie będziesz musiał konfigurować.

Nie znajdziesz na rynku obszerniejszej książki poświęconej systemom Unix i Linux. Jeżeli jesteś z nimi związany w pracy zawodowej, interesujesz się systemami operacyjnymi lub korzystasz z nich w domu, ta książka obowiązkowo musi znaleźć się na Twojej półce!

Ta książka to obowiązkowa pozycja dla wszystkich administratorów!

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 32 230 98 63
e-mail: helion@helion.pl

SPIS TREŚCI

2 Pisanie skryptów i powłoka 87

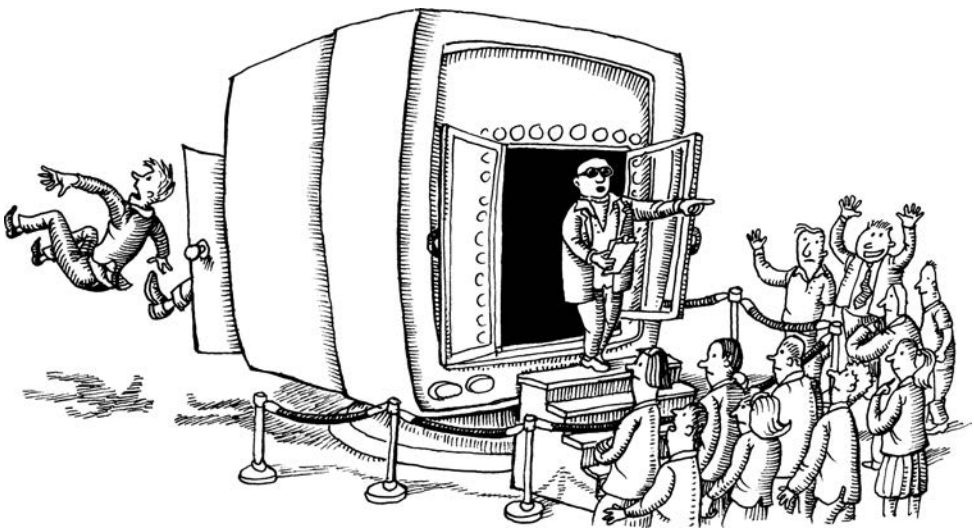
28 Informatyka a ekologia 1331

29 Wydajność 1349

DODATKI

7

DODAWANIE NOWYCH UŻYTKOWNIKÓW



W większości systemów jednym z rutynowych obowiązków jest dodawanie i usuwanie nowych użytkowników. Są to zadania proste, ale i nużące, dlatego większość administratorów, aby zautomatyzować ten proces, wprowadza do narzędzi dostarczanych wraz z systemem operacyjnym pewne ulepszenia, a następnie przekazuje właściwą pracę asystentowi lub operatorowi.

Oprócz rozproszonych serwerów, na których jest jedynie dwóch użytkowników, obserwujemy obecnie ponowny wzrost liczby serwerów scentralizowanych, zawierających setki kont. Administratorzy muszą posiadać gruntowną znajomość systemu kont

użytkowników, aby zarządzać usługami sieciowymi i konfigurować konta w sposób właściwy dla lokalnego środowiska komputerowego. Zarządzanie kontami na serwerach to często jedynie jeden z wielu elementów obsługi kont dla całego przedsiębiorstwa.

Współczesne środowiska produkcyjne wymagają nie tylko narzędzi do dodawania użytkowników do określonych komputerów, ale i takich, które umożliwią zarządzanie użytkownikami i ich niezliczonymi kontami oraz hasłami w całym środowisku komputerowym, czyli potrzebują systemu zarządzania tożsamością. Usługi katalogowe, takie jak Active Directory firmy Microsoft, OpenLDAP czy Fedora Directory Server, są w powszechnym użyciu, dlatego omówimy tu dokładniej wpływ tego rodzaju systemów na zadania związane z zarządzaniem kontami (jak zwykle, krótkowzroczny Microsoft nie współpracuje zbyt dobrze z innymi, chyba że przekażesz zarządzanie Active Directory).

Potrzeby niektórych ośrodków mogą przekraczać możliwości nawet takich systemów. Nie będziemy tu omawiać komercyjnych systemów zarządzania tożsamością, ale wskażemy kilku kandydatów. Prawdopodobnie systemy te będą odpowiednim rozwiązaniem dla bardzo dużych ośrodków, zwłaszcza takich, w których wymagane jest przestrzeganie regulacji prawnych, takich jak ustawa HIPAA (ang. *Health Insurance Portability and Accountability Act*) lub ustawa Sarbanesa-Oxleya (w przypadku USA); patrz strona 292.

Higiena obsługi kont jest kluczowym warunkiem bezpieczeństwa systemu. Rzadko używane konta są głównym celem ataków, podobnie jak konta z łatwymi do odgadnięcia hasłami. Jeśli nawet do dodawania i usuwania użytkowników wykorzystujesz zautomatyzowane narzędzia dostępne w Twoim systemie, ważne jest, abyś rozumiał, jakie zmiany wprowadzają te narzędzia. Dlatego też temat zarządzania kontami zaczniemy od omówienia zwykłych plików, które należy zmodyfikować, aby dodać użytkowników do pojedynczego komputera.

Następnie zajmiemy się zautomatyzowanymi narzędziami występującymi w każdym z naszych przykładowych systemów operacyjnych oraz plikami konfiguracyjnymi, które sterują ich zachowaniem. Co dziwne (a może raczej dezorientujące), we wszystkich naszych przykładowych systemach narzędzia do zarządzania kontami noszą nazwy `useradd`, `userdel` i `usermod`, choć niekoniecznie są to dokładnie takie same programy (poza tym w systemie AIX ta zgodność nazewnicza została uzyskana przez zastosowanie skryptów osłonowych dla rodzimych narzędzi `mkuser`, `rmuser` i `chuser`).

Domyślne narzędzie `useradd` jest całkiem dobre i powinno wystarczyć do obsługi większości ośrodków. Niestety, narzędzie `userdel` nie zawsze jest tak dokładne, jak byśmy sobie tego życzyli.

Większość systemów zawiera również proste narzędzia graficzne umożliwiające dodawanie i usuwanie użytkowników, choć zazwyczaj nie oferują one trybu wsadowego ani zaawansowanych ustawień lokalizacyjnych. Są na tyle proste, że nie warto tu szczegółowo omawiać ich działania, wskażemy jednak, gdzie można uzyskać dokumentację producenta dla każdego z tych narzędzi.

W tym rozdziale skupimy się głównie na dodawaniu i usuwaniu użytkowników. Wiele kwestii zawiązanych z zarządzaniem użytkownikami omawiamy w innych rozdziałach, a tutaj będziemy się jedynie do nich odwoływać, np.:

- mechanizm PAM (ang. *Pluggable Authentication Modules*) odpowiadający za szyfrowanie uwierzytelniania i wymuszający stosowanie silnych haseł, omawiany jest w rozdziale 22., „Bezpieczeństwo" (od strony 1113),
- skrytki i depozyty ułatwiające zarządzanie hasłami opisane są w rozdziale 4., „Kontrola dostępu i uprawnienia administratora" (patrz strona 189),
- usługi katalogowe, takie jak NIS i OpenLDAP, przedstawione zostały w rozdziale 19., „Współdzielenie plików systemowych", od strony 899; kilka komentarzy na temat Active Directory można też znaleźć w rozdziale 30., „Współpraca z Windows", na stronie 1399,
- kwestie proceduralne i prawne są głównym tematem rozdziału 32., „Zarządzanie i reguły w IT".

W trzech następnych podrozdziałach przyjrzymy się podstawowym plikom, które odgrywają dużą rolę w zarządzaniu użytkownikami.

7.1. PLIK ETC/PASSWD

Plik */etc/passwd* zawiera listę użytkowników rozpoznawanych przez system. Może on zostać rozszerzony lub zastąpiony przez usługę katalogową, stanowi więc kompletne i autorytatywne rozwiązanie tylko w samodzielnych systemach.

Podczas logowania system odwołuje się do pliku */etc/passwd*, aby określić m.in. identyfikator użytkownika (UID) i jego katalog domowy. Każdy wiersz w tym pliku reprezentuje jednego użytkownika i zawiera siedem pól rozdzielonych dwukropkami. Oto te pola:

- nazwa użytkownika,
- symbol zastępujący zaszyfrowane hasło (patrz strona 263),
- numer UID (identyfikator użytkownika),
- domyślny numer GID (identyfikator grupy),
- informacje „GECOS": pełna nazwa, numer pokoju, telefon wewnętrzny, telefon domowy,
- katalog domowy,
- domyślna powłoka.

Przykładowo poniższe wiersze są prawidłowymi wpisami pliku */etc/passwd*:

```
root:x:0:0:The System,,x6096,:/:/bin/sh
jl:!:100:0:Jim Lane,ECOT8-3,,:/staff/jl:/bin/sh
dotty:x:101:20::/home/dotty:/bin/tcsh
```

Zaszyfrowane hasła umieszczane były w drugim polu, ale nie jest to już bezpieczne rozwiązanie; przy wykorzystaniu szybkiego sprzętu ich złamanie (odszyfrowanie) jest kwestią minut. Obecnie wszystkie wersje systemów Unix i Linux ukrywają zaszyfrowane hasła i umieszczają je w oddzielnym pliku, który nie jest przeznaczony do odczytu dla innych. W systemach Linux, Solaris i HP-UX plik *passwd* w polu przeznaczonym na zaszyfrowane hasło zawiera znak x, a w systemie AIX jest to ! lub * (w przypadku AIX znak * jest symbolem zastępczym, który wyłącza konto).

W rzeczywistości zaszyfrowane hasło w systemach Linux, Solaris i HP-UX przechowywane jest w pliku */etc/shadow*, a w systemie AIX w pliku */etc/security/passwd*. Mają one różne formaty.

Jeśli konta użytkowników są współdzielone za pośrednictwem usługi katalogowej, np. NIS lub LDAP, w pliku *passwd* mogą znajdować się specjalne wpisy rozpoczynające się znakiem + lub –. Wpisy te informują system o tym, w jaki sposób dane usługi katalogowej mają być zintegrowane z zawartością pliku */etc/passwd*. Sposób integracji można również ustawić w pliku */etc/nsswitch.conf* (lub */etc/nscontrol.conf* w systemie AIX).

➲ *Więcej informacji na temat pliku* nsswitch.conf *można znaleźć na stronie 912.*

W następnych punkach omówimy szczegółowo plik */etc/passwd*.

Nazwa użytkownika

Nazwy użytkowników muszą być unikatowe i — w zależności od systemu operacyjnego — muszą przestrzegać ograniczeń dotyczących długości i zestawu znaków. W tabeli 7.1 przedstawione zostały zasady obowiązujące w naszych przykładowych systemach. Nazwy użytkowników nie mogą nigdy zawierać dwukropków ani znaków nowego wiersza, ponieważ znaki te są używane odpowiednio jako separatory pól i separatory wpisów. Jeśli korzystasz z NIS lub NIS+, nazwy użytkowników mogą zawierać maksymalnie osiem znaków, niezależnie od stosowanego systemu operacyjnego.

➲ *Więcej informacji na temat NIS można znaleźć na stronie 908.*

Pierwotnie systemy Unix dopuszczały stosowanie wyłącznie znaków alfanumerycznych i ograniczały długość nazwy użytkownika do ośmiu znaków. Każdy system ma nieco inne reguły, dlatego musisz przyjąć najbardziej restrykcyjne ograniczenia, aby zapobiec ewentualnym konfliktom. Taki konserwatyzm zagwarantuje, że użytkownicy będą mogli posługiwać się tą samą nazwą użytkownika na każdym komputerze. Najbardziej uniwersalnym rozwiązaniem jest stosowanie nazw składających się z ośmiu (lub mniej) znaków i zawierających wyłącznie małe litery, cyfry i znaki podkreślenia.

Tabela 7.1. Reguły tworzenia nazw użytkowników

System	Liczba znaków	Zestaw znaków	Pierwszy znak	Dodatkowe wymogi
Linux	32[a]	a-z0-9_-	a-z_	niektóre dystrybucje są bardziej pobłażliwe
Solaris	8[b]	A-Za-z0-9+.-_	A-Za-z	co najmniej jedna mała litera
HP-UX	8	A-Za-z0-9_	A-Za-z	
AIX	8[c]	POSIX; bez spacji, cudzysłowów i znaków #,=/?\	inny niż -@~	niedozwolone same duże litery niedozwolone są nazwy „default" i „ALL"

[a] Choć Linux dopuszcza 32 znaki, tradycyjne programy (np. `top` lub `rsh`) oczekują 8 lub mniej znaków.

[b] To ograniczenie zostało zwiększone.

[c] W systemie AIX w wersji 5.3 lub wyższej można to zmienić, patrz następna strona.

W nazwach użytkowników uwzględnia się wielkość znaków. Dokument RFC 822 nakazuje jednak, aby w adresach e-mail wielkość znaków była ignorowana. Nic nam nie wiadomo o jakichkolwiek problemach związanych z łączeniem dużych znaków z małymi w nazwach użytkowników, ale tradycyjnie stosuje się nazwy zawierające małe znaki, poza tym są one łatwiejsze do wpisywania. Jeśli nazwy użytkowników *jan* i *Jan* przypisane będą różnym osobom, mogą wystąpić problemy związane z pocztą.

Nazwy użytkowników powinny być łatwe do zapamiętania, dlatego przypadkowa sekwencja znaków nie jest dobrą nazwą. Unikaj stosowania przezwisk nawet wtedy, kiedy w Twojej organizacji panują nieformalne stosunki. Nazwy w rodzaju *DarkLord* lub *FajnaLaska* pasują bardziej do *@hotmail.com*. Jeśli Twoi użytkownicy nie mają żadnego szacunku dla siebie, weź pod uwagę choćby ogólną wiarygodność Twojej organizacji.

Ponieważ nazwy użytkowników są często używane w adresach e-mail, warto ustalić standardowy sposób ich tworzenia. Użytkownicy powinni samodzielnie odgadnąć nazwy pozostałych użytkowników. Rozsądny schemat nazewniczy musi raczej zakładać użycie imion, nazwisk, inicjałów lub kombinacji tych elementów.

Bez względu na przyjęty schemat wyboru nazw użytkowników i tak w końcu dojdzie do sytuacji, w której wystąpią zduplikowane lub zbyt długie nazwy, dlatego czasami trzeba robić wyjątki. Określ standardowy sposób postępowania w przypadku konfliktów, np. dodanie numeru na końcu nazwy. Przy zbyt długich nazwach możesz wykorzystać funkcję nazw zastępczych swojego systemu pocztowego, aby utożsamić dwie wersje nazwy, co przynajmniej rozwiązuje problem z pocztą.

W dużych organizacjach często stosuje się adresy e-mail zawierające pełne imię i nazwisko (np. *Jan.Kowalski@mojawitryna.com*), dzięki czemu właściwe nazwy użytkowników pozostaną nieznane dla osób postronnych. To dobry pomysł, ale nie zwalnia od obowiązku stosowania przedstawionych wyżej wskazówek. Najlepiej, aby nazwy użytkowników miały ścisły oraz przewidywalny związek z ich faktycznymi imionami i nazwiskami, choćby dla samego ułatwienia administrowania, jeśli nie ma innych powodów.

Nazwy użytkowników powinny być unikatowe z dwóch powodów. Po pierwsze, użytkownik powinien na każdym komputerze logować się za pomocą tej samej nazwy. Ta zasada służy głównie dla wygody zarówno Twojej, jak i użytkownika.

Po drugie, określona nazwa użytkownika powinna zawsze odnosić się do tej samej osoby. Niektóre polecenia (np. `ssh`) mogą być skonfigurowane w taki sposób, że zdalni użytkownicy będą weryfikowani na podstawie ich nazw. Jeśli *scott@boulder.colorado.edu* i *scott@refuge.colorado.edu* to dwie różne osoby, przy niewłaściwie skonfigurowanych kontach jeden użytkownik *scott* mógłby zalogować się na konto drugiego użytkownika o tej samej nazwie bez podawania hasła.

Z doświadczenia wynika, że zduplikowane nazwy prowadzą do pomyłek w dostarczaniu poczty. System pocztowy może doskonale odróżniać obu użytkowników o nazwie *scott*, ale często inni użytkownicy będą wysyłali pocztę pod zły adres.

Jeśli w Twojej organizacji istnieje globalny plik nazw zastępczych (aliasów), nowe nazwy użytkowników nie mogą się pokrywać z żadną z nazw określonych w tym pliku. W przeciwnym razie poczta zostanie dostarczona użytkownikowi, do którego należy dana nazwa zastępcza, a nie nowemu użytkownikowi.

➲ *Więcej informacji na temat stosowania nazw zastępczych w adresach pocztowych można znaleźć na stronie 935.*

W systemie AIX można zmienić maksymalną długość nazwy użytkownika za pomocą polecenia `chdev`. Odpowiednie urządzenie nosi nazwę `sys0`. Polecenie `lsattr -D -l sys0` wyświetli listę domyślnych atrybutów urządzenia. Jednym z nich jest atrybut `max_logname`, który odpowiada za maksymalną długość nazwy użytkownika. Poniższe polecenie wyświetli tylko ten konkretny atrybut:

```
aix$ lsattr -El sys0 -a max_logname
max_logname 9 Maximum login name length at boot time True
```

Aby zmienić to ograniczenie, użyj poniższego polecenia. Zmiana zostanie wprowadzona przy następnym uruchomieniu systemu[1].

```
aix$ sudo su -
aix# chdev -l sys0 -a max_logname=16
```

System podaje, że domyślna długość nazwy to dziewięć znaków, ale specyfikacja długości w systemie AIX to rozmiar bufora, a w nim mieści się jeszcze znak null, który przerywa łańcuch. Dlatego też faktyczny domyślny limit to tylko osiem znaków, a nasze polecenie `chdev` ustawia limit na piętnaście znaków.

[1] Na początku nam się to nie udawało, ponieważ korzystaliśmy z `sudo` (patrz strona 186), a zmienne środowiskowe ustawione przez polecenie `sudo` zazwyczaj są inne niż wtedy, gdy najpierw wydajemy polecenie `sudo su -`, a dopiero potem właściwe polecenie. Polecenie `chdev` zwraca na to uwagę. Nowe wersje polecenia `sudo` (1.70 lub wersje późniejsze) mają opcję `–i`, którą można zastosować w takich sytuacjach.

AIX obsługuje znaki wielobajtowe (np. dla języków azjatyckich), ale nie zaleca się ich używania. Sugerowanym rozwiązaniem jest stosowanie znaków dopuszczalnych w nazwach plików zgodnie ze standardem POSIX.

Zaszyfrowane hasło

Nowoczesne systemy nie przechowują hasła w pliku */etc/passwd*, a przy pierwszym logowaniu proszą użytkownika o podanie prawdziwego hasła. Ponadto, oprócz standardowego algorytmu `crypt` systemu Unix, obsługują również kilka innych metod szyfrowania. Zaszyfrowane hasło jest oznaczane w celu identyfikacji sposobu szyfrowania użytego do jego wygenerowania. Nasze przykładowe systemy obsługują wiele różnych algorytmów szyfrowania: tradycyjny `crypt` (oparty na DES), MD5, Blowfish oraz iteratywną wersję MD5 zapożyczoną z projektu serwera WWW Apache.

Kolejną ważną kwestią jest długość hasła często określana przez algorytm wykorzystany do szyfrowania. W tabeli 7.2 przedstawione zostały maksymalne i minimalne długości haseł oraz algorytmy szyfrowania dostępne w naszych przykładowych systemach. Niektóre systemy umożliwiają wpisanie dowolnie długiego hasła, ale „po cichu" przycinają je zgodnie z limitem podanym w tabeli.

Tabela 7.2. Algorytmy szyfrowania i ograniczenia długości hasła

System	Minimum	Maksimum	Algorytmy	Miejsce ustawienia
Linux	5	8	`crypt`, MD5, Blowfish[a], SHA256, SHA 512	*/etc/login.defs*
Solaris	6	8[b]	`crypt`, MD5, Blowfish, SHA256, SHA 512	*/etc/security/policy.conf* */etc/security/crypt.conf*
HP-UX	6[c]	8	`crypt`	*/usr/include/limits.h*[d]
AIX	0	8	`crypt`, MD5 (BSD), Apache	argument polecenia `passwd`

[a] W systemach SUSE i openSUSE domyślnym algorytmem jest Blowfish; większość pozostałych używa MD5.

[b] Maksymalna długość zależy od wybranego algorytmu.

[c] Administrator (*root*) może ustawić nazwę użytkownika o dowolnej długości.

[d] Ten plik zawiera bardzo dużo konstrukcji `#ifdef`, dlatego jest dość trudny do odczytania i zrozumienia.

Jeśli pominiesz systemowe narzędzia do dodawania użytkowników i zdecydujesz się utworzyć nowe konto przez ręczną edycję pliku */etc/passwd* (oczywiście, za pomocą `vipw`; patrz strona 274), w polu przeznaczonym na zaszyfrowane hasło wpisz znak gwiazdki lub `x`. Zapobiegnie to możliwości nieautoryzowanego użycia konta przed ustawieniem właściwego hasła przez Ciebie lub użytkownika. Nigdy, w żadnym przypadku nie pozostawiaj tego pola pustego, bo możesz doprowadzić do ogromnej luki w bezpieczeństwie systemu, ponieważ wtedy dostęp do tego konta nie wymaga podania żadnego hasła.

Algorytm MD5 pod względem kryptograficznym jest nieco lepszy niż poprzedni standard DES używany przez `crypt`, poza tym mechanizm MD5 umożliwia stosowanie haseł o dowolnej długości. Dłuższe hasła są bezpieczniejsze — pod warunkiem, że z nich

korzystasz. W algorytmie MD5 odkryto pewne niedoskonałości, ale na standard DES udało się przeprowadzić skuteczne ataki typu brute-force. Algorytmy SHA256 i Blowfish to prawdziwi siłacze w dziedzinie szyfrowania. Na stronie 1111 podane zostały wskazówki dotyczące wyboru haseł.

Zaszyfrowane hasła mają stałą długość (trzydzieści cztery znaki dla MD5, trzynaście dla DES), bez względu na długość niezaszyfrowanego hasła. Hasła są szyfrowane w połączeniu z wartościami losowymi (ang. *salt*), aby dane hasło mogło przyjmować wiele zaszyfrowanych form. Jeśli dwóch użytkowników wybierze to samo hasło, nie można będzie odkryć tego faktu przez porównanie zaszyfrowanych haseł. Hasła MD5 są łatwe do zauważenia, ponieważ zawsze rozpoczynają się od `$1$` lub `$md5$`[2]. Hasła zaszyfrowane algorytmem Blowfish rozpoczynają się od `$2a$`, a hasła SHA256 — od `$5$`.

Dla nowych haseł system SUSE stosuje domyślnie algorytm Blowfish, co jest bardzo rozsądnym rozwiązaniem. Szukaj prefiksu `$2a$`.

Obecnie w OpenSolaris domyślnym algorytmem jest SHA256 (prefiks `$5$`), choć wcześniejsze wersje używały domyślnie MD5.

Numer UID (identyfikator użytkownika)

Numer UID identyfikuje użytkownika w systemie. Nazwy użytkowników służą tylko ich wygodzie, bo oprogramowanie i system plików posługują się wewnętrznie numerami UID. Identyfikatory UID to zwykle 32-bitowe liczby całkowite bez znaku.

Użytkownik *root* z definicji ma identyfikator UID o numerze 0. Większość systemów definiuje również pseudoużytkowników, takich jak *bin* i *daemon*, którzy są właścicielami plików konfiguracyjnych. Tradycyjnie tacy sztuczni użytkownicy umieszczani są na początku pliku */etc/passwd* i mają przyznane niskie numery UID oraz fałszywą powłokę (np. */bin/false*), żeby nikt nie mógł się zalogować na ich konta. Aby pozostawić dużo miejsca dla takich pseudoużytkowników, których mógłbyś dodawać w przyszłości, zalecamy, aby identyfikatory UID prawdziwych użytkowników rozpoczynały się od numeru 500 lub wyższego (pożądany zakres dla nowych identyfikatorów można określić w plikach konfiguracyjnych dla polecenia `useradd`).

➲ *Opis konta* root *można znaleźć na stronie 176.*

Inny specjalny identyfikator UID należy do użytkownika o nazwie *nobody*; ma on zwykle przypisaną wartość –1 lub –2, co odpowiada najwyższemu i drugiemu w kolejności identyfikatorowi spośród wszystkich możliwych numerów UID. Nazwa *nobody* używana jest wtedy, gdy na jednym z komputerów użytkownik *root* próbuje uzyskać przez NFS dostęp do plików innego komputera, który nie ufa temu pierwszemu.

[2] `$1$` to oznaczenie algorytmu MD5 w BSD; Sun stosuje własny wariant MD5 i oznacza go `$md5$`.

➲ *Więcej informacji na temat konta* nobody *można znaleźć na stronie 862.*

Posiadanie wielu kont o identyfikatorze UID równym 0 to zły pomysł. Możliwość logowania się z uprawnieniami administratora na różne powłoki lub przy użyciu różnych haseł może wydawać się udogodnieniem, jednak takie ustawienie powoduje tylko większe zagrożenie dla bezpieczeństwa i konieczność zabezpieczenia wielu kont. Jeśli użytkownicy muszą mieć więcej sposobów logowania się z uprawnieniami użytkownika *root*, lepiej będzie, jeśli skorzystają z programów typu `sudo`.

Unikaj powtórnego wykorzystywania numerów UID nawet wtedy, kiedy użytkownicy opuścili już organizację, a ich konta zostały usunięte. Dzięki podjęciu takich środków ostrożności unikniesz zamieszania w sytuacji, gdyby kiedyś zaszła konieczność odtworzenia plików z kopii zapasowej, gdzie użytkownicy są identyfikowani za pomocą numerów UID, a nie nazw.

Numery UID muszą być unikatowe w całej organizacji. Oznacza to, że określony numer UID powinien odnosić się do tej samej nazwy użytkownika i tej samej osoby na wszystkich komputerach, do korzystania z których jest uprawniona. Nieprzestrzeganie zasady niepowtarzalności numerów UID może zwiększyć zagrożenie dla bezpieczeństwa w takich systemach jak NFS, a także wprowadzać zamęt w sytuacji, gdy użytkownik zmieni swoją grupę roboczą.

Jeśli grupy komputerów są zarządzane przez różne osoby lub organizacje, utrzymanie unikatowych numerów UID może być trudne. Wiążą się z tym zarówno problemy natury technicznej, jak i politycznej. Najlepszym rozwiązaniem jest posiadanie centralnej bazy danych lub serwera katalogowego, który zawiera wpisy każdego użytkownika i wymusza niepowtarzalność. Prostszy schemat polega na przypisaniu określonego zakresu numerów UID każdej grupie w organizacji i pozwolenie jej na samodzielne zarządzanie swoimi zestawami. Takie rozwiązanie pozwala utrzymać odrębne przestrzenie numerów UID, ale nie rozwiązuje występujących równolegle problemów z unikatowymi nazwami użytkowników.

W zarządzaniu numerami UID i informacjami o kontach użytkowników dużą popularność zdobywa LDAP. Temat ten zostanie przedstawiony pobieżnie w tym rozdziale, od strony 290, a bardziej dokładnie omówimy go w rozdziale 19., „Współdzielenie plików systemowych", od strony 899.

Domyślne numery GID

Podobnie jak UID, numery identyfikacyjne grup również są 32-bitowymi liczbami całkowitymi. GID o numerze 0 jest zarezerwowany dla grupy o nazwie *root* lub *system*. Tak samo jak w przypadku identyfikatorów UID, system dla własnych potrzeb administracyjnych korzysta z kilku predefiniowanych grup. Niestety, producenci nie zachowują tu żadnej konsekwencji. Przykładowo w systemach Red Hat i SUSE grupa *bin* ma identyfikator GID 1, a w systemach Ubuntu, Solaris, HP-UX i AIX grupa ta ma identyfikator o numerze 2.

W dawnych czasach, gdy moc obliczeniowa była droga, grupy służyły do celów rozliczeniowych. Chodziło o to, aby opłaty za wykorzystanie sekund czasu procesora, minut czasu zalogowania i kilobajtów zajętego miejsca na dysku zostały naliczone właściwemu wydziałowi. Obecnie grupy służą głównie do współdzielenia dostępu do plików.

Plik */etc/group* definiuje grupy; korzysta przy tym z pola GID w pliku */etc/passwd*, przyznając domyślny (lub „efektywny") numer GID podczas logowania. Przy określaniu praw dostępu domyślny numer GID nie jest traktowany w jakiś szczególny sposób, ma znaczenie jedynie podczas tworzenia nowych plików i katalogów. Standardowo właścicielem nowych plików jest Twoja efektywna grupa, jeśli jednak chcesz udostępnić pliki innym użytkownikom w ramach jakiegoś projektu zespołowego, musisz pamiętać o tym, aby ręcznie zmienić właściciela grupowego tych plików.

➲ *Więcej informacji na temat ustawiania bitu setgid dla katalogów można znaleźć na stronie 233.*

Aby ułatwić współpracę, możesz ustawić dla katalogu bit setgid (02000) lub zamontować system plików z opcją `grpid`. Oba te rozwiązania sprawią, że nowo tworzone pliki będą domyślnie należeć do tej samej grupy, co ich katalog nadrzędny.

Pole GECOS

Pole GECOS używane jest czasem do zapisywania informacji osobistych o każdym użytkowniku. Nie ma ściśle zdefiniowanej składni. Choć możesz tu przyjąć dowolną konwencję formatowania, polecenie `finger` interpretuje wpisy GECOS rozdzielone przecinkami w następującej kolejności:

- imię i nazwisko (często jest to jedyne używane pole),
- numer budynku i pokoju,
- wewnętrzny numer telefonu,
- domowy numer telefonu.

Polecenie `chfn` umożliwia użytkownikom zmianę swoich informacji GECOS[3]. Polecenie jest przydatne do zapisywania aktualnych informacji o użytkowniku, takich jak numer telefonu, ale może również zostać wykorzystane niewłaściwie. Użytkownik może np. wprowadzić tam niewłaściwe informacje lub treści obsceniczne. W niektórych systemach można ograniczyć pola, które mogą być modyfikowane za pomocą polecenia `chfn`; wiele ośrodków akademickich wyłącza je całkowicie. W większości systemów polecenie `chfn` może operować tylko na pliku */etc/passwd*, jeśli więc do przechowywania informacji o użytkownikach wykorzystujesz LDAP lub inną usługę katalogową, `chfn` może w ogóle nie działać.

[3] Wyjątkiem jest system Solaris, w którym nie ma polecenia `chfn`. Superużytkownik może zmienić informacje GECOS użytkownika za pomocą polecenia `passwd –g`.

➲ *Więcej informacji na temat LDAP można znaleźć na stronie 899.*

W systemie AIX polecenie `chfn` akceptuje opcję `–R` *`moduł`*, która wczytuje podany *`moduł`* w celu przeprowadzenia właściwej aktualizacji. Dostępne moduły znajdują się z katalogu */usr/lib/security*, a jeden z nich może korzystać z LDAP.

Katalog domowy

Katalog domowy użytkownika jest podczas logowania katalogiem domyślnym. Musisz zdawać sobie sprawę z tego, że katalogi domowe zamontowane na sieciowym systemie plików mogą być niedostępne w przypadku problemów z serwerem lub siecią. Jeśli w czasie logowania katalog domowy użytkownika jest niedostępny, system wyświetli komunikat informujący o braku katalogu domowego[4] i przeniesie użytkownika do katalogu /. System Linux nie pozwoli się zalogować, jeśli w pliku */etc/login.defs* w wierszu `DEFAULT_HOME` wpisana będzie wartość `no`.

Powłoka logowania

Powłoka logowania to zazwyczaj interpreter poleceń, np. powłoka Bourne'a lub C (`/bin/sh` lub `/bin/csh`), ale może to być dowolny program. W systemie Unix tradycyjnie powłoką domyślną jest `sh`, a w systemach Linux i Solaris jest nią `bash` (ang. *„Bourne again" shell*[5]). W systemie AIX powłoką domyślną jest `ksh`, powłoka Korna; `tcsh` to rozszerzona powłoka C z możliwością edycji poleceń. W systemach linuksowych `sh` i `csh` są w rzeczywistości po prostu dowiązaniami odpowiednio do powłok `bash` i `tcsh`.

Niektóre systemy umożliwiają użytkownikom zmianę swoich powłok za pomocą polecenia `chsh`, ale podobnie jak w przypadku `chfn`, polecenie to może nie działać, jeśli do zarządzania informacjami o użytkownikach wykorzystujesz LDAP lub inną usługę katalogową. Gdy używasz pliku */etc/passwd*, administrator zawsze może zmienić powłokę użytkownika, edytując plik *passwd* za pomocą polecenia `vipw`.

Linux obsługuje polecenie `chsh` i ogranicza możliwość zmian do powłok wymienionych w pliku */etc/shells*. SUSE również narzuca wybór z pliku */etc/shells*, ale system Red Hat w przypadku wybrania powłoki spoza listy tylko ostrzega. Dodając nowe wpisy do pliku *shells*, pamiętaj o konieczności podania ścieżek bezwzględnych, ponieważ oczekują tego `chsh` i inne programy.

---

[4] Taki komunikat pojawi się tylko przy logowaniu na konsoli lub przez terminal, ale nie podczas logowania za pośrednictwem graficznych menadżerów logowania, takich jak `xdm`, `gdm` lub `kdm`. Wtedy nie tylko nie zobaczysz żadnego komunikatu, ale zostaniesz natychmiast wylogowany, ponieważ menadżer wyświetlania nie będzie miał możliwości zapisu do właściwego katalogu (np. *~/.gnome*).

[5] Gra słów; w języku angielskim brzmi to jak *born again shell*, czyli odrodzona powłoka — *przyp. tłum.*

W systemie AIX użytkownik może zmienić swoją powłokę za pomocą polecenia `chsh` i wybrać inną z obszernej listy. Autorytatywna lista sprawdzonych powłok znajduje się w pliku */etc/security/login.cfg*[6]. Plik */etc/shells* zawiera jedynie ich podzbiór i używany jest tylko przez demona FTP, `in.ftpd`. Wiele powłok z tej długiej listy to po prostu dowiązania twarde do pojedynczego pliku binarnego. Przykładowo `sh`, `ksh`, `rksh`, `psh` i `tsh` (zarówno w */bin*, jak i w */usr/bin*) są tym samym programem, który zmienia swoje zachowanie w zależności od tego, pod jaką nazwą został wywołany. Podobnie jak `chfn`, polecenie `chsh` przyjmuje opcję `–R` *`moduł`* w celu obsługi LDAP i innych systemów usług katalogowych.

W systemie Solaris tylko superużytkownik może zmienić powłokę użytkownika (używając polecenia `passwd -e`). Plik */etc/shells* (którego domyślnie nie ma, choć istnieje jego strona podręcznika systemowego) zawiera listę dozwolonych powłok.

7.2. PLIKI /ETC/SHADOW I /ETC/SECURITY/PASSWD

Plik *shadow* (zwany też plikiem przesłaniania haseł lub plikiem cieni) może być odczytany tylko przez superużytkownika. W pliku przechowywane są zaszyfrowane hasła, chronione przed wścibskimi oczami i programami do łamania haseł. Mieszczą się w nim również pewne dodatkowe informacje o kontach, które nie były uwzględnione w pierwotnym formacie */etc/passwd*. Obecnie przesłanianie haseł jest domyślnie włączone niemal we wszystkich systemach.

W systemie AIX plik przechowujący zaszyfrowane hasła nosi nazwę */etc/security/passwd*, podczas gdy we wszystkich pozostałych systemach jest to */etc/shadow*. Oczywiście, ich formaty i zawartość są różne. Najpierw przyjrzymy się plikowi */etc/shadow*.

Plik *shadow* nie jest nadzbiorem pliku *passwd*, a plik *passwd* nie jest generowany na jego podstawie. Musisz utrzymywać oba te pliki lub skorzystać z takich narzędzi jak `useradd`, które będą to robić za Ciebie. Podobnie jak */etc/passwd*, plik */etc/shadow* zawiera jeden wiersz dla każdego użytkownika. Każdy wiersz to dziewięć pól rozdzielonych dwukropkami, a przeznaczonych na:

- nazwę użytkownika,
- zaszyfrowane hasło,
- datę ostatniej zmiany hasła,
- minimalną liczbę dni między zmianami hasła,
- maksymalną liczbę dni między zmianami hasła,

[6] Do wprowadzania zmian w plikach z katalogu */etc/security* użyj polecenia `chsec`, zamiast edytować je bezpośrednio.

- liczbę dni określającą wyprzedzenie, z jakim należy ostrzegać użytkowników o wygaśnięciu hasła,
- liczbę dni od wygaśnięcia hasła, po upływie których konto zostanie wyłączone (Linux) lub liczbę dni, po których konto automatycznie wygaśnie (Solaris),
- datę ważności konta,
- pole zarezerwowane, które obecnie jest zawsze puste, z wyjątkiem systemu Solaris.

Jedyne wymagane wartości to nazwa użytkownika i hasło. W pliku */etc/shadow* pola z datami bezwzględnymi podawane są jako liczba dni (**nie** sekund), jakie upłynęły od 1 stycznia 1970 roku, co nie jest standardowym sposobem odliczania czasu w systemach Unix i Linux. Możesz jednak zamienić sekundy na dni, jakie upłynęły od początku ery Uniksa, stosując polecenie:

```
solaris$ expr `date +%s` / 86400
```
[7]

Typowy wpis w pliku *shadow* wygląda tak:

```
millert:$md5$em5J8hL$a$iQ3pXeOsakdRaRFyy7Ppj.:14469:0:180:14:::
```

Oto bardziej szczegółowe opisy tych pól.

- Nazwa użytkownika jest taka sama jak w pliku */etc/passwd*. Pole łączy wpisy użytkownika w plikach *passwd* i *shadow*.
- Zaszyfrowane hasło jest identyczne w koncepcji i wykonaniu z tym, które kiedyś przechowywane było w pliku */etc/passwd*; tutaj przedstawione zostało fikcyjne hasło MD5 z systemu Solaris.
- W polu ostatniej zmiany zapisany jest czas, w którym hasło użytkownika było ostatnio zmieniane. Pole to jest wypełniane przez polecenie `passwd`.
- Czwarte pole określa liczbę dni, które muszą upłynąć między zmianami hasła. Chodziło o to, aby zmusić użytkownika do autentycznej zmiany hasła, uniemożliwiając mu natychmiastowe przywrócenie poprzedniego hasła po wymuszonej zmianie. Uważamy jednak, że funkcja ta może być nieco niebezpieczna w razie naruszenia bezpieczeństwa. Zalecamy ustawienie w tym polu wartości 0.
- Piąte pole określa maksymalną liczbę dozwolonych dni między zmianami hasła. Dzięki temu administrator może wymusić zmiany haseł; więcej informacji na ten temat można znaleźć na stronie 1111. W systemie Linux rzeczywista maksymalna liczba dni jest sumą wartości tego pola oraz wartości z pola siódmego (tzw. okres łaski).
- Szóste pole określa liczbę dni przed wygaśnięciem hasła, kiedy program `login` powinien ostrzec użytkownika o nieuchronnie zbliżającym się terminie.

[7] Dzień ma 86 400 sekund: 60*60*24.

- W systemach Solaris i HP-UX siódme pole interpretowane jest inaczej niż w systemie Linux. W Linuksie siódme pole określa, ile dni po przekroczeniu maksymalnego terminu ważności hasła należy odczekać, zanim konto zostanie zablokowane.

  W systemach Solaris i HP-UX działanie tego pola wygląda następująco: jeśli użytkownik nie zaloguje się w czasie wyznaczonym przez liczbę dni podaną w siódmym polu, konto zostanie wyłączone. Nieużywane konta są ulubionym celem hakerów, a ta funkcja daje możliwość pozbycia się takich kont. Działa to jednak tylko wtedy, gdy użytkownik jest wymieniony w pliku */var/adm/lastlog*, wiec konta użytkowników, którzy nigdy się nie zalogowali, nie będą automatycznie wyłączane. A zatem funkcja ta nie sprawdzi się w środowisku sieciowym, ponieważ każdy komputer ma własny plik *lastlog*.
- Ósme pole określa dzień (liczony od 1 stycznia 1970 roku) wygaśnięcia konta użytkownika. Po tej dacie użytkownik nie będzie mógł się zalogować, chyba że administrator ponownie wprowadzi tu jakąś wartość. Jeśli pole pozostanie puste, konto nigdy nie wygaśnie.

  W systemie Linux można ustawić datę wygaśnięcia za pomocą polecenia `usermod`, które przyjmuje daty w formacie `rrrr-mm-dd`. W systemie Solaris polecenie `usermod` potrafi również obliczać dni, jakie upłynęły od początku ery Uniksa. Akceptuje ono daty w ok. trzydziestu formatach określonych w pliku */etc/datemsk*, niestety, nie ma wśród formatu `rrrr-mm-dd` używanego w systemie Linux.
- Dziewiąte pole jest zarezerwowane do wykorzystania w przyszłości. Jest to uznawane przez systemy Linux i HP-UX, ale Solaris wykorzystuje ostatnie cztery bity do odliczania nieudanych prób zalogowania.

Popatrzmy ponownie na nasz przykładowy wiersz z pliku *shadow*:

```
millert:$md5$em5J8hL$a$iQ3pXeOsakdRaRFyy7Ppj.:14469:0:180:14:::
```

W tym przykładzie użytkownik *millert* po raz ostatni zmieniał swoje hasło 13 sierpnia 2009 roku. Hasło musi zostać zmienione ponownie w ciągu 180 dni, a przez ostatnie dwa tygodnie przed tym terminem użytkownik *millert* będzie otrzymywał ostrzeżenia o konieczności zmiany hasła. Konto nie ma ustawionej daty wygaśnięcia.

W systemach Solaris, HP-UX i Linux możesz użyć narzędzia `pwconv`, aby uzgodnić zawartość pliku *shadow* z plikiem *passwd* przez dodanie nowych użytkowników i usunięcie tych, których nie ma już w pliku *passwd*. W systemie Linux program `pwconv` wypełnia większość wpisów w pliku *shadow* wartościami domyślnymi podanymi w pliku */etc/login.defs*.

W systemie Solaris użytkownik *root* za pomocą polecenia `passwd -f` *`nazwa_użytkownika`* może wymusić na użytkowniku zmianę hasła przy następnym logowaniu. Jest to przydatne w sytuacji, gdy regularnie uruchamiasz programy do łamania zabezpieczeń w celu wykrycia źle wybranych (słabo zabezpieczonych) haseł (w systemie Linux ta sama opcja `-f` umożliwia użytkownikowi zmianę informacji na swój temat).

W systemie AIX nie używa się terminu przesłaniania haseł, ale koncepcja jest taka sama. Zaszyfrowane hasła przechowywane są w pliku */etc/security/passwd*, w całkowicie innym formacie niż ten, który występuje w pliku */etc/passwd*. Oto przykład pochodzący z czystej instalacji systemu AIX, gdzie domyślnym algorytmem szyfrowania haseł jest `crypt`[8]:

```
trent:
    password = u10.OaYxRx4qI
    lastupdate = 1224876639
    flags = ADMCHG

evi:
    password = PiIr2qOPabZ.Q
    lastupdate = 1235785246
    flags =
```

Ten format nie wymaga dokładniejszych wyjaśnień. Poszczególne wpisy oddzielone są jedną lub kilkoma pustymi wierszami. Taki sam format stosowany jest w większości plików konfiguracyjnych w katalogu */etc/security*, gdzie nazwa użytkownika wyznacza dowolny kontrolowany lub rejestrowany obiekt.

System AIX dostarcza niezliczoną liczbę ustawień kontrolujących wszystkie aspekty logowania i haseł. Niektóre opcje dotyczą użytkowników, a inne portów (aby kontrolować porty TTY, na których dany użytkownik może się zalogować). Szczegółowe informacje znajdują się komentarzach zawartych w plikach *etc/security/login.cfg* i */etc/security/user*. Przydatnym poleceniem jest `pwdadm`, które umożliwia wymuszenie na użytkowniku zmiany hasła przy następnym logowaniu.

7.3. PLIK /ETC/GROUP

Plik */etc/group* zawiera nazwy grup w systemie Unix oraz listę członków każdej grupy. Oto fragment pliku *group* z systemu AIX:

```
system:!:0:root,pconsole,esaadmin
staff:!:1:ipsec,esaadmin,trent,ben,garth,evi
bin:!:2:root,bin
sys:!:3:root,bin,sys
adm:!:4:bin,adm
nobody:!:4294967294:nobody,lpd
```

Każdy wiersz odpowiada jednej grupie i zawiera cztery pola:

- nazwę grupy,
- zaszyfrowane hasło lub znak zastępczy,

[8] Wybranie silniejszego algorytmu szyfrowania powinno być jedną z pierwszych czynności do wykonania po zainstalowaniu systemu AIX.

- numer GID,
- listę członków grupy, rozdzielonych przecinkami (uważaj, aby nie wpisywać spacji).

Podobnie jak w przypadku pliku */etc/passwd*, poszczególne pola rozdzielane są dwukropkami. Dla zachowania zgodności nazwy grup powinny być ograniczone do ośmiu znaków, choć wiele systemów tego nie wymaga. Możliwe jest ustawienie hasła dla grupy, dzięki czemu użytkownicy niebędący jej członkami mogą ją wybrać za pomocą polecenia `newgrp`, ale rzadko się z tego korzysta. Tylko Linux w pełni obsługuje hasła grup[9]. Hasło można ustawić za pomocą polecenia `gpasswd`; będzie ono przechowywane w zaszyfrowanej formie w pliku */etc/gshadow*. Zazwyczaj nie stosuje się haseł dla grup, występują one bardzo rzadko.

Podobnie jak nazwy użytkowników i numery UID, nazwy grup i numery GID powinny być jednakowe na wszystkich komputerach, które współdzielą pliki za pośrednictwem sieciowego plików. Może to być trudne do uzyskania w środowisku heterogenicznym, ponieważ odmienne systemy operacyjne używają różnych numerów GID dla grup o tych samych nazwach.

Odkryliśmy, że najlepszym sposobem poradzenia sobie z tym problemem jest unikanie używania grupy systemowej jako domyślnej grupy logowania użytkownika. Niektóre systemy kontrolują wykonywanie poleceń przy użyciu własności grupy w połączeniu z bitami uprawnień. Niejednolite numery GID na różnych systemach powodują zamieszanie przy instalowaniu i aktualizacji oprogramowania w ramach całej organizacji.

Jeśli użytkownik ma przypisaną domyślną grupę w pliku */etc/passwd*, ale nie jest wymieniony jako członek tej grupy w pliku */etc/group*, ostatnie słowo należy do */etc/passwd*. Członkostwo grupy przydzielane podczas logowania jest połączeniem tego, co znajduje się w plikach *passwd* i *group*.

Niektóre systemy ograniczają liczbę grup, do których użytkownik może należeć. Najczęściej spotykany limit to 8 grup, ale w systemie Solaris wynosi on 16, w HP-UX — 20, a w systemach AIX i Linux liczba grup jest pozornie nieograniczona.

Aby zminimalizować potencjalne konflikty z numerami GID przydzielanymi domyślnie przez producentów, zalecamy rozpoczęcie numerowania lokalnych grup od 500 lub wyższych wartości.

Pierwotnie w systemie Unix nowym użytkownikom przydzielana była grupa reprezentująca ich ogólną kategorię, np. „studenci” lub „finanse”. Taka konwencja zwiększała jednak prawdopodobieństwo, że przez niedbałe ustawienia dostępu użytkownicy będą mogli czytać nawzajem swoje pliki, nawet jeśli nie było to intencją właściciela. Aby uniknąć tego problemu, należy tworzyć unikatową grupę dla każdego użytkownika. Grupa może nosić tę samą nazwę co użytkownik. Możesz też przydzielić im ten sam numer GID i UID.

[9] Aby ustawić hasło dla grupy w systemie Solaris, trzeba posłużyć się plikiem *shadow* i ręcznie skopiować oraz wkleić łańcuch z hasłem do pliku */etc/group*. Nie ma tu pliku */etc/gshadow* ani jego odpowiednika.

We wszystkich dystrybucjach Linuksa, z wyjątkiem SUSE, narzędzie `useradd` domyślnie tworzy dla użytkownika prywatną grupę. Systemy Unix domyślnie umieszczają wszystkich nowych użytkowników w tej samej grupie, ale program `useradd` można w nich skonfigurować w taki sposób, aby obsługiwał również prywatne grupy.

Użytkownik powinien być jedynym członkiem swojej prywatnej grupy. Jeśli chcesz umożliwić użytkownikom współdzielenie plików z wykorzystaniem mechanizmu grup, utwórz w tym celu osobne grupy. Koncepcja grupy prywatnej nie miała na celu zniechęcenia do korzystania z grup jako takich — chodziło po prostu o wprowadzenie bardziej restrykcyjnej grupy **domyślnej** dla każdego użytkownika, aby nie dochodziło do niezamierzonego udostępniania plików. Możesz również osiągnąć ten cel za pośrednictwem polecenia powłoki `umask` (patrz strona 238).

Systemy Linux, Solaris i HP-UX zawierają polecenia służące do tworzenia, modyfikowania i usuwania grup: `groupadd`, `groupmod` i `groupdel`. System AIX wymaga ręcznej modyfikacji pliku */etc/group* za pomocą edytora tekstu. Zawiera on jednak polecenie `grpck` służące do sprawdzania składni tego pliku.

7.4. DODAWANIE UŻYTKOWNIKÓW — PODSTAWY

Zanim utworzysz konto dla nowego użytkownika w firmie, urzędzie lub instytucji naukowej, musisz pamiętać o tym, aby użytkownik podpisał i opatrzył datą umowę użytkownika i regulamin korzystania z kont. (Co?! Nie masz umowy użytkownika i regulaminu? Zajrzyj na stronę 1473, gdzie dowiesz się, dlaczego ich potrzebujesz i co powinny zawierać).

Użytkownicy nie mają żadnego określonego powodu, aby podpisywać regulamin, więc zdobądź ich podpisy, póki jeszcze możesz to wymusić. Stwierdziliśmy, że zdobycie podpisu pod regulaminem wymaga znacznie więcej wysiłku, gdy konto zostało już założone. Jeśli tylko procedura to dopuszcza, wykonaj całą papierkową robotę przed utworzeniem konta.

Proces dodawania nowego użytkownika składa się z kilku kroków wymaganych przez system, dwóch kroków mających na celu skonfigurowanie wygodnego środowiska pracy dla nowego użytkownika oraz kilku dodatkowych kroków dla wygody administratora.

Działania wymagane to:

- podpisanie regulaminu przez użytkownika,
- zmodyfikowanie plików *passwd* i *shadow* w celu zdefiniowania konta użytkownika,
- dodanie użytkownika do pliku */etc/group* (nie jest to konieczne, ale warto to zrobić),
- ustawienie początkowego hasła,
- utworzenie katalogu domowego użytkownika i ustawienie jego własności i praw dostępu (polecenia `chown` i `chmod`),
- skonfigurowanie ról i uprawnień (jeśli korzystasz z RBAC, zajrzyj na stronę 277).

Dla użytkownika:

- skopiowanie domyślnych plików startowych do katalogu domowego użytkownika,
- skonfigurowanie skrzynki pocztowej użytkownika i ustawienie aliasów pocztowych.

Dla administratora:

- sprawdzenie, czy konto jest poprawnie skonfigurowane,
- wprowadzenie informacji kontaktowych o użytkowniku i statusu konta do bazy danych.

Ta lista aż prosi się o skrypt lub inne narzędzie, na szczęście, każdy z naszych przykładowych systemów zawiera coś takiego w postaci polecenia `useradd`.

Aby dodawać użytkowników, musisz być użytkownikiem *root*, a w systemie AIX musisz mieć uprawnienie UserAdmin. To doskonała okazja do skorzystania z `sudo`; patrz strona 186.

Edycja plików passwd i group

Jeśli musisz ręcznie dodać użytkownika, do modyfikowania plików *passwd* i *shadow* używaj polecenia `vipw`. Wbrew nazwie nie wymaga ono korzystania z `vi`, możesz użyć swojego ulubionego edytora zdefiniowanego w zmiennej środowiskowej `EDITOR`. Co ważniejsze, powoduje ono zablokowanie pliku, dzięki czemu Twoje modyfikacje i operacje zmiany hasła przez użytkowników nie będą ze sobą kolidować.

W systemach Solaris i Red Hat po zmodyfikowaniu pliku *passwd* polecenie `vipw` automatycznie spyta, czy chcesz dokonać edycji pliku *shadow*. W systemach SUSE i Ubuntu funkcję tę spełnia polecenie `vipw –s`.

W systemach HP-UX i AIX nie zaleca się ręcznej edycji pliku *passwd*, bez względu na to, czy miałoby się to odbywać za pomocą polecenia `vipw`, czy bez niego (w systemie AIX w ogóle nie ma tego polecenia). Zamiast tego należy skorzystać odpowiednio z polecenia `useradd` lub wielozadaniowych narzędzi administracyjnych `smh` i SMIT. Polecenie `useradd` zostanie omówione szczegółowo, początek na stronie 278.

Jeśli nowy użytkownik ma należeć do kilku innych grup, oprócz grupy domyślnej określonej w pliku *passwd*, musisz zmodyfikować plik */etc/group* i dopisać nazwę użytkownika do każdej z dodatkowych grup.

Ustawianie hasła

Nigdy nie pozostawiaj nowego konta — ani żadnego innego konta, które ma dostęp do powłoki — bez ustawionego hasła. Złożoność hasła można wymusić za pomocą plików konfiguracyjnych; pod koniec tego rozdziału podamy listę plików i zmiennych używanych w poszczególnych systemach operacyjnych. Hasło dla nowego użytkownika możesz ustawić przy użyciu polecenia:

```
$ sudo passwd nazwa_nowego_użytkownika
```

➲ *Zasady wyboru dobrego hasła znajdziesz na stronie 183.*

Zostaniesz poproszony o podanie rzeczywistego hasła. Niektóre zautomatyzowane systemy dodawania nowych użytkowników nie wymagają podawania początkowego hasła. W zamian użytkownik zmuszony jest ustawić hasło podczas pierwszego logowania. Jest to wprawdzie wygodne, ale stanowi olbrzymie zagrożenie: każdy, kto odgadnie nazwę nowego użytkownika (lub odszuka ją w pliku */etc/passwd*), może się zaczaić i przejąć konto, zanim uprawniony użytkownik będzie miał sposobność się zalogować.

Tworzenie katalogu domowego i instalowanie plików startowych

Katalog domowy dla nowego użytkownika możesz utworzyć zwykłym poleceniem `mkdir`. Musisz również ustawić odpowiednie prawa własności i uprawnienia do katalogu, ale najlepiej zrobić to dopiero po zainstalowaniu lokalnych plików startowych.

Tradycyjnie nazwy plików startowych rozpoczynają się kropką i kończą literami *rc*, co jest skrótem od „run command" (uruchom polecenie), pozostałości po systemie operacyjnym CTSS. Początkowa kropka powoduje, że polecenie `ls` ukrywa te „interesujące" pliki z listy zawartości katalogu, chyba że użyta zostanie opcja `-a`.

Zalecamy, aby dostarczyć użytkownikom domyślne pliki startowe dla każdej powłoki występującej w Twoich systemach, aby nadal mieli sensowne środowisko domyślne, jeśli zdecydują się zmienić powłokę. W tabeli 7.3 zamieściliśmy listę typowych plików startowych.

Przykładowe pliki startowe tradycyjnie przechowywane są w katalogu */etc/skel* (Linux, Solaris, HP-UX) lub */etc* (wszystkie systemy). W systemie AIX, gdzie zawsze wszystko jest trochę inne, są one ukryte w katalogu */etc/security*). Jeśli modyfikujesz przykładowe pliki startowe dostarczone przez producenta systemu, warto ich zmodyfikowane kopie umieścić w katalogu */usr/local/etc/skel*. Linux część plików startowych przechowuje w katalogu */etc/profile.d*, gdzie powłoka szuka informacji dotyczących kolorowania wyjścia polecenia `ls`, aby było czytelne na ciemnym tle, lub ścieżki dostępu do plików binarnych Kerberosa.

W zależności od powłoki użytkownika, katalog */etc* może zawierać ogólnosystemowe pliki startowe, które są przetwarzane przed plikami startowymi użytkownika. Przykładowo powłoki `bash` i `sh` wczytują najpierw plik */etc/profile*, a dopiero potem przystępują do przetwarzania plików *~/.profile* i *~/.bash_profile*. Pliki te są dobrym miejscem na wprowadzenie ogólnosystemowych ustawień domyślnych, należy jednak pamiętać o tym, że użytkownik może nadpisać te ustawienia własnymi plikami startowymi. Szczegóły dotyczące innych powłok znajdziesz na stronach podręcznika systemowego.

Pamiętaj o tym, aby ustawić sensowne wartości domyślne dla `umask`; zalecamy tu 077, 027 lub 002, w zależności od poziomu zaufania i rozmiaru organizacji. Jeśli nie używasz indywidualnych grup, zalecamy maskę 077, ponieważ daje pełne prawa dostępu

Tabela 7.3. Pliki startowe i ich zastosowanie

Polecenie	Nazwa pliku	Typowe zastosowanie
`sh`	*.profile*	ustawienie ścieżki wyszukiwania, typu terminala i środowiska
`bash`[a]	*.bashrc*	ustawienie typu terminala (jeśli jest to konieczne) ustawienie opcji `biff` i `mesg`
	.bash_profile	ustawienie zmiennych środowiskowych ustawienie nazw zastępczych (aliasów) dla poleceń ustawienie ścieżki wyszukiwania ustawienie wartości `umask` w celu ustalenia uprawnień ustawienie zmiennej `CDPATH` do wyszukiwania plików ustawienie zmiennych `PS1` (znak zachęty) i `HISTCONTROL`
`csh` (`tcsh`)	*.login*	podobne do *.bashrc* dla `csh`
	.cshrc	podobne do *.login* dla `csh`
`vi` (`vim`)	*.exrc* (*.vimrc*)	ustawienie opcji edytora `vi` (`vim`)
`emacs`	*.emacs*	ustawienie opcji i przypisań klawiszy edytora `emacs`
`mail` (`mailx`)	*.mailrc*	definicje prywatnych aliasów pocztowych ustawienie opcji czytnika poczty (oryginalnego klienta poczty systemu UNIX)
GNOME	*.gconf*	środowisko GNOME: konfiguracja użytkownika wprowadzona przez `gconf`
	.gconfpath	ścieżka dodatkowej konfiguracji użytkownika przez `gconf`
KDE	*.kde/*	środowisko KDE: katalog z plikami konfiguracyjnymi

[a] W trybie emulacji `sh` `bash` wczytuje również plik *.profile* lub */etc/profile*.

właścicielowi, odbierając dostęp grupie i pozostałym użytkownikom. Szczegółowe informacje na temat `umask` znajdują się na stronie 238.

Pliki startowe i katalogi z ustawieniami dla środowisk GNOME i KDE to tylko wierzchołek góry lodowej; `gconf` to narzędzie przechowujące ustawienia aplikacji dla programów działających pod GNOME, co przypomina nieco rejestr w systemie Windows.

Ustawianie uprawnień i praw własności

Po skonfigurowaniu katalogu domowego przekaż go użytkownikowi i upewnij się, że ma on odpowiednie uprawnienia. Można to wykonać za pomocą polecenia:

```
$ sudo chown -R nowy_użytkownik:nowa_grupa ~nowy_użytkownik
```

Zauważ, że nie możesz użyć polecenia:

```
$ sudo chown nowy_użytkownik:nowa_grupa ~nowy_użytkownik/.*
```

aby ustawić właściciela plików rozpoczynających się znakiem kropki, ponieważ *nowy_*
↪użytkownik nabędzie prawa własności nie tylko do własnych plików, ale również do katalogu nadrzędnego (np. */home*). To bardzo częsta i niebezpieczna pomyłka.

Ustawienie miejsca odbioru poczty

Rozwiązaniem wygodnym dla użytkowników jest odbieranie poczty tylko na jednym komputerze. Często jest to realizowane za pomocą wpisu w globalnym pliku aliasów */etc/mail/aliases* lub opcji `userDB` programu `sendmail` na głównym serwerze pocztowym. Ogólne informacje na temat poczty znajdują się w rozdziale 20.

Konfigurowanie ról i uprawnień administracyjnych

Kontrola dostępu oparta na rolach (RBAC) umożliwia rozdzielanie uprawnień systemowych na poszczególnych użytkowników i jest dostępna na wielu naszych przykładowych systemach. Mechanizm RBAC nie jest tradycyjną częścią systemu Unix ani modelu kontroli dostępu w Linuksie, ale jeśli stosuje się go w Twojej organizacji, konfigurowanie ról musi być częścią procesu dodawania użytkowników. RBAC został szczegółowo omówiony od strony 179 w rozdziale 4., „Kontrola dostępu i uprawnienia administratora".

Regulacje prawne, takie jak ustawa Sarbanesa-Oxleya (SOX) i ustawa Gramma-Leacha-Blileya (GLBA) w Stanach Zjednoczonych, skomplikowały wiele aspektów administracji systemem w przedsiębiorstwach, w tym również zarządzanie użytkownikami. Może się okazać, że role są jedynym realnym rozwiązaniem, które będzie w stanie zapewnić zgodność z wymogami prawnymi.

➲ *Więcej informacji na temat ustaw SOX i GLBA można znaleźć w rozdziale 32.*

Końcowe kroki

Aby sprawdzić, czy nowe konto zostało poprawnie utworzone, najpierw musisz się wylogować, a następnie załogować ponownie jako nowy użytkownik i wykonać poniższe polecenia:

```
$ pwd       /* Aby zweryfikować katalog domowy */
$ ls -la    /* Aby sprawdzić własność i grupę plików startowych */
```

Musisz podać nowym użytkownikom ich nazwy i początkowe hasła. Często informacje te przesyłane są pocztą elektroniczną, ale ze względu na bezpieczeństwo zazwyczaj nie jest to dobry pomysł. Zrób to osobiście lub telefonicznie, chyba że tworzysz 500 kont na komputerach CS-1 dla nowych studentów w miasteczku uniwersyteckim. Wtedy przerzuć problem na ich wykładowców! Jest to również dobry moment na przekazanie użytkownikom dodatkowej dokumentacji omawiającej lokalne zwyczaje, jeśli ją opracowałeś.

Jeśli Twoja organizacja wymaga od użytkowników podpisania regulaminu lub zasad właściwego korzystania z konta, upewnij się, że zostało to zrobione przed aktywacją. Pozwoli to zapobiec niedopatrzeniom i da podstawę do ewentualnych sankcji, które być może trzeba będzie w przyszłości zastosować.

➲ *Więcej informacji na temat podpisywania umowy użytkownika można znaleźć na stronie 1487.*

Przypomnij nowym użytkownikom o konieczności natychmiastowej zmiany hasła. Jeśli chcesz, możesz to wymusić, ustawiając szybkie wygaśnięcie hasła. Innym rozwiązaniem jest użycie skryptu, który przetestuje konta nowych użytkowników i sprawdzi, czy ich zaszyfrowane hasła w pliku *shadow* zostały zmienione[10].

W środowisku, w którym osobiście znasz użytkowników, możesz stosunkowo łatwo śledzić, kto i dlaczego korzysta z systemu. Jeśli jednak zarządzasz dużą i dynamicznie zmieniającą się bazą użytkowników, musisz znaleźć bardziej sformalizowany sposób śledzenia aktywności kont. Jeśli kiedyś fakt utworzenia konta zatrze się w Twojej pamięci, utrzymywanie bazy danych zawierającej informacje kontaktowe i statusy kont pozwoli Ci łatwo sprawdzić, kim jest dany użytkownik i dlaczego ma założone konto.

7.5. DODAWANIE UŻYTKOWNIKÓW ZA POMOCĄ USERADD

W każdym systemie program `useradd` realizuje tę samą opisaną wcześniej prostą procedurę. Jest on jednak konfigurowalny i zapewne warto dostosować go do swojego środowiska. Ponieważ każdy system ma własną koncepcję na temat tego, co powinno być dostosowane, gdzie należy wyprowadzić zmiany i jakie powinno być domyślne zachowanie, omówmy te szczegóły osobno.

W tabeli 7.4 zawarliśmy podręczny spis poleceń i plików konfiguracyjnych związanych z zarządzaniem użytkownikami. Każdy z naszych przykładowych systemów ma pakiet poleceń służących do działań na użytkownikach, zwykle zawiera on co najmniej `useradd`, `usermod` i `userdel`. Ponieważ wszystkie te polecenia są konfigurowane w podobny sposób, zaprezentujemy je tu na przykładzie polecenia `useradd`, uzupełniając jego wpisy o inne polecenia specyficzne dla danego systemu.

Warto zauważyć, że choć w każdym systemie polecenia te noszą nazwy `useradd` itd., same narzędzia różnią się w zależności od systemu.

Polecenie useradd w systemie Ubuntu

Ubuntu umożliwia dodawanie użytkowników na dwa sposoby: za pomocą `adduser` i `useradd`. Polecenie `adduser` to napisany w Perlu skrypt osłonowy dla polecenia `useradd`, który jest nieco bardziej pomocny (tworzy katalogi domowe, kopiuje pliki startowe itd.).

Polecenie `adduser` jest konfigurowane w pliku */etc/adduser.conf*, który zawiera m.in. następujące opcje:

[10] Ponieważ to samo hasło może mieć wiele zaszyfrowanych form, ta metoda sprawdza tylko, czy użytkownik zresetował hasło, a nie to, czy rzeczywiście zmienił je na inne.

Tabela 7.4. Polecenia i pliki konfiguracyjne służące do zarządzania użytkownikami

System	Polecenia	Pliki konfiguracyjne	Uwagi
Ubuntu	`useradd`	*/etc/login.defs*	
		/etc/default/useradd	
	`adduser`	*/etc/adduser.conf*	bardziej przyjazna wersja Perla
SUSE	`useradd`	*/etc/login.defs*	
		/etc/default/useradd	
		/etc/default/passwd	
		/usr/sbin/useradd.local	dla dostosowań lokalnych
		/usr/sbin/userdel.local	dla dostosowań lokalnych
		/usr/sbin/userdel-pre.local	dla dostosowań lokalnych
		/usr/sbin/userdel-post.local	dla dostosowań lokalnych
Red Hat	`useradd`	*/etc/login.defs*	
		/etc/default/useradd	
Solaris	`useradd`	*/etc/default/*{*login,passwd*}	
		/etc/security/policy.conf	
HP-UX	`useradd`	*/etc/default/useradd*	
		/etc/default/security	
	`smh`		narzędzie GUI, zwane również `sam`
AIX	`useradd`	*/etc/security/user*	
		/etc/security/login.cfg	
		/etc/security/mkuser.default[a]	
	`mkuser`		wywoływane przez `useradd`
	`chuser`		wywoływane przez `usermod`
	`rmuser`		wywoływane przez `userdel`
	SMIT		narzędzie GUI

[a] W starszych systemach AIX plik ten znajduje się w katalogu */usr/lib/security*.

- reguły lokalizacji katalogów domowych: pod względem grup, nazw użytkowników itd.,
- ustawienia uprawnień dla nowych katalogów domowych,
- zakresy numerów UID i GID dla użytkowników systemowych i ogólnych,
- opcję tworzenia indywidualnej grupy dla każdego użytkownika,
- udziały dyskowe (niestety, tylko jako wartość logiczną),
- dopasowanie nazw użytkowników i grup na podstawie wyrażeń regularnych.

Inne typowe parametry polecenia `useradd`, takie jak reguły dotyczące haseł, ustawiane są jako parametry modułu PAM odpowiedzialnego za regularne uwierzytelnianie

haseł (omówienie PAM, czyli dołączalnych modułów uwierzytelniania, znajduje się na stronie 1113). Polecenie `adduser` ma bliźniacze polecenie `addgroup` oraz pokrewne `deluser` i `delgroup`.

Polecenie useradd w SUSE

Polecenie `useradd` w systemie SUSE domyślnie nie tworzy katalogu domowego dla nowego użytkownika ani nie kopiuje plików startowych. Musisz zażądać tych udogodnień, podając opcję `–m` (pliki startowe pobierane są z katalogu */etc/skel*); `useradd` nie tworzy też plików poczty dla nowych użytkowników.

Polecenie `useradd` w SUSE również nie powoduje powstania indywidualnych grup dla użytkowników; domyślny numer GID w pliku *passwd* ustawiany jest na podstawie zmiennej `GROUP` w pliku */etc/default/useradd*. Ponadto nowi użytkownicy dodawani są do grup określonych przez zmienną `GROUPS`. Domyślnie są to grupy *video* i *dialout*, które umożliwiają dostęp do bufora ramek i oprogramowania `pppd` do obsługi połączeń dodzwanianych.

Aby zrekompensować te różnice, polecenie `useradd` w systemie SUSE wywołuje `/usr/sbin/useradd.local`, skrypt powłoki `bash`, który możesz dostosować do swoich potrzeb.

Plik */etc/login.defs* w systemie SUSE reguluje następujące kwestie:

- pozwolenie na zalogowanie, jeśli katalog domowy użytkownika nie istnieje,
- stopień tolerancji (zwłoka i blokada) przy nieudanych próbach zalogowania,
- położenie plików „wiadomości dnia” i *ttytype* (typy terminali),
- ograniczenie dotyczące korzystania z poleceń `chsh` i `chfn`,
- wygasanie haseł,
- zakresy numerów UID i GID dla systemu i użytkowników,
- reguły tworzenia właściwych nazw użytkowników i grup,
- maski użytkowników (domyślnie 022),
- lokalne skrypty korzystające z poleceń `useradd` i `userdel`.

Zarówno strona podręcznika systemowego dla pliku *login.defs*, jak i komentarze w samym pliku zawierają cenne informacje opisujące różne parametry i ich znaczenie.

Polecenie useradd w systemie Red Hat

W systemie Red Hat Linux Enterprise polecenie `useradd` pobiera swoje parametry konfiguracyjne z pliku */etc/login.defs*, który odpowiada za takie kwestie jak wygasanie haseł, algorytmy szyfrowania, pliki poczty oraz zakresy numerów UID i GID. Poszczególne parametry wyjaśnione są w komentarzach zawartych w tym pliku.

Polecenie `useradd –D` wyświetla parametry domyślne, które będą zastosowane dla nowych użytkowników. Są one zapisane w pliku */etc/default/useradd*. W systemie Red Hat nowi użytkownicy umieszczani są w swoich indywidualnych grupach. We wpisach w pliku *passwd* użyto znaku „x" jako symbolu zastępczego dla hasła, a w pliku *shadow* użyto kodu „!!", który wyłącza logowanie i wymaga, aby administrator systemu ustawił hasło dla nowego użytkownika. Domyślnym algorytmem szyfrowania jest MD5. W katalogu nowego użytkownika umieszczane są pliki startowe z katalogu */etc/skel*.

Polecenie useradd w systemie Solaris

W systemie Solaris niektóre domyślne parametry dotyczące logowania i haseł przechowywane są w plikach */etc/default/login* i */etc/default/passwd*; pozostałe zostały wbudowane w program `useradd`. Polecenie `useradd –D` wyświetli domyślne wartości dla kilku parametrów. Za pomocą dodatkowych opcji można je wykorzystać do przywrócenia niektórych ustawień domyślnych.

Formaty plików *default/login* i *default/passwd* są podobne do formatu pliku *login.defs* w systemie Linux. Puste wiersze i wiersze rozpoczynające się znakiem # są ignorowane, a każdy wiersz niebędący komentarzem stanowi przypisanie wartości lub zmiennej. Składnia wygląda jednak tak:

```
NAZWA=wartość
```

a nie tak:

```
NAZWA <puste znaki> wartość
```

Plik */etc/default/passwd* kontroluje następujące elementy:

- minimalną długość hasła,
- wygaśnięcie hasła,
- wymagany stopień złożoności hasła,
- sprawdzenie możliwości złamania hasła.

Plik */etc/default/login* odpowiada za takie kwestie jak:

- strefa czasowa,
- ograniczenie wielkości plików, jakie użytkownik może tworzyć,
- możliwość logowania się użytkownika *root* tylko na konsoli,
- wymagalność hasła dla każdego użytkownika,
- obsługa nieudanych prób zalogowania,
- początkowa ścieżka wyszukiwania danego użytkownika,
- domyślna maska użytkownika (domyślnie 022),
- opcje programu `syslog`: czy ma rejestrować logowania użytkownika *root* i nieudane próby zalogowania.

Pliki */etc/security/policy.conf* i */etc/security/crypt.conf* określają algorytmy szyfrowania, które mogą być używane dla haseł.

Polecenie useradd w systemie HP-UX

Domyślnie polecenie `useradd` w systemie HP-UX nie tworzy katalogów domowych ani nie umieszcza użytkownika w jego indywidualnej grupie. Opcja `–m` spowoduje jednak, że `useradd` utworzy katalogi domowe i umieści w nich pliki startowe z katalogu */etc/skel*.

Parametry konfiguracyjne polecenia `useradd` ustawione są w plikach */etc/default/useradd* i */etc/default/security*, przy czym plik *useradd* przyjmuje format w stylu Linuksa, czyli `NAZWA <puste znaki> wartość`, a plik *security* używa formatu Solarisa, tj. `NAZWA=wartość`. HP potrafi nieźle namieszać! Składnię można wywnioskować z innych wpisów w pliku, jeśli jednak użyjesz złej składni, wartość, którą spróbujesz ustawić, zostanie po cichu zignorowana. Nie pojawi się żaden komunikat o błędzie.

Plik */etc/default/useradd* kontroluje takie opcje jak:

- domyślna grupa i powłoka,
- główny poziom drzewa katalogu domowego,
- wygasanie konta,
- tworzenie katalogów domowych,
- pozwalanie na duplikowanie numerów UID.

Plik */etc/default/security* zawiera dodatkowe parametry konfiguracyjne, z których część dotyczy zarządzania użytkownikami, czyli:

- pozwolenia na logowanie w przypadku braku katalogu domowego,
- dopuszczania pustych haseł,
- minimalnej długość hasła,
- obsługi nieudanych prób logowania,
- obsługi nieaktywnych kont,
- domyślnej maski dla nowych użytkowników (domyślnie 027).

Nazwy zmiennych w tym pliku są długie i trafne, dzięki czemu dokładnie określają, za co odpowiada każda zmienna.

Polecenie useradd w systemie AIX

W systemie AIX polecenie `useradd` to w rzeczywistości skrypt osłonowy powłoki `ksh` dla `mkuser`, rodzimego odpowiednika `useradd` w AIX. Podobnie polecenie `usermod` wywołuje `chuser`, a `userdel` wywołuje `rmuser`. Strony podręcznika systemowego

dla tych poleceń można wywołać, odwołując się zarówno do ich oryginalnych nazw, jak i tych, pod jakimi występują w pozostałych systemach.

Pliki konfiguracyjne kontrolujące wiele różnych aspektów logowania i haseł znajdują się w katalogu */etc/security*. Są tam trzy istotne pliki: *login.cfg*, *user* i *mkuser.default*. Pierwsze dwa używają znaku * dla oznaczania komentarzy; trzeci nie zawiera komentarzy. Pliki te podzielone są na sekcje w następującej postaci:

```
etykieta:
    atrybut = wartość
    następny-atrybut = wartość

następna-etykieta:
    atrybut = wartość
```

Przykładowo w pliku */etc/security/user* etykietami są nazwy użytkowników (lub słowo `default`); możliwe atrybuty podane są w tabeli 7.5 na następnej stronie.

Całkiem spora lista! Komentarze w pliku często podają wartości domyślne, które są dość sensowne dla instalacji o niskim poziomie bezpieczeństwa. Zalecamy wprowadzenie kilku zmian:

- zmianę wartości `umask` z 022 na 077,
- zmianę wartości `loginretries` z 0 (brak ograniczeń) na małą liczbę całkowitą, np. 5,
- zmianę wartości `minlen` z 0 (dopuszczalny brak hasła) na co najmniej 6 lub 7,
- zmianę wartości `expires` z 0 (nigdy) na rok (tylko jeśli dysponujesz narzędziem umożliwiającym okresowe odświeżanie dat wygaśnięcia dla prawowitych użytkowników).

Niestety, to tylko jeden z plików konfiguracyjnych kontrolujących logowanie nowych użytkowników. Plik */etc/security/login.cfg* zawiera parametry kontrolujące nieudane próby logowania (opóźnienie między wyświetleniem znaków zachęty do podania loginu i hasła, liczbę dozwolonych prób zalogowania przed wyłączeniem konta, czas trwania blokady konta, czas jego przywrócenia itd.), przedziały czasowe, w których można się zalogować, znak zachęty wyświetlany podczas prośby o podanie hasła, listę dozwolonych powłok, maksymalną dopuszczalną liczbę jednoczesnych logowań, czas oczekiwania na wprowadzenie hasła, rodzaj używanego uwierzytelniania (jeśli korzystasz z PAM[11], musisz to podać właśnie w tym pliku) oraz algorytmy szyfrowania haseł (domyślnie jest to `crypt`). Możliwości konfiguracji są niesamowicie rozbudowane, aż do przesady. Jeszcze większy zamęt powoduje to, że niektóre parametry pojawiają się w obu plikach, przy czym czasem występują pod tą samą nazwą (np. `logintimes`), a czasem nazywają się inaczej (np. `loginretries` i `logindisable`). Krótko mówiąc, jeden wielki bałagan i mętlik.

[11] Mechanizm PAM pojawił się w AIX stosunkowo niedawno; pełną funkcjonalność powinien mieć w wersjach 5.3 lub późniejszych.

Tabela 7.5. Opcje konta użytkownika w pliku /etc/security/user (AIX)

Opcja	Wartość	Znaczenie
account_locked	logiczna	wartość true blokuje możliwość logowania
admin	logiczna	wartość true nadaje uprawnienia administratora
auth1	lista metod	główna metoda uwierzytelniania
auth2	lista metod	drugorzędna metoda uwierzytelniania
dictionlist	nazwy plików	słowniki, w których nie może występować hasło
expires	data	data wygaśnięcia konta użytkownika
histexpire	tygodnie	okres, przez jaki użytkownik nie może ponownie użyć wykorzystanego hasła
histsize	numer	liczba poprzednich haseł, które nie mogą zostać użyte ponownie
login	logiczna	czy pozwolić na zalogowanie (przydatne dla kont typu bin)
loginretries	numer	liczba prób zalogowania przed zablokowaniem konta
logintimes	przedział czasu	ograniczenie czasu, w którym użytkownik może się zalogować
maxexpired	tygodnie	okres łaski dla haseł, które wygasły
maxrepeats	numer	liczba dopuszczalnych powtórzeń znaku w haśle
minage, maxage	tygodnie	minimalny i maksymalny okres ważności hasła
minalpha	numer	minimalna liczba liter w haśle
mindiff	znaki	liczba znaków starego hasła, które mogą wystąpić w nowym haśle
minlen	numer	minimalna długość hasła (nie ustawiaj tu wartości 0)
minother	numer	minimalna liczba znaków nieliterowych w haśle
pwdchecks	nazwy plików	funkcje wywoływane w celu sprawdzenia bezpieczeństwa haseł
pwdwarntime	dni	okres łaski ostrzegający użytkownika o konieczności zmiany hasła
rlogin	logiczna	określa, czy użytkownik może się zalogować na to konto przez rlogin lub telnet
su	logiczna	określa, czy inni użytkownicy mogą zalogować się na to konto przez su
ttys	lista urządzeń	terminale, na których użytkownik może się logować
umask	ósemkowa	domyślne prawa dostępu do plików tworzonych przez użytkownika

➲ *Więcej informacji na temat PAM można znaleźć na stronie 1113.*

W AIX polecenie useradd nie ma opcji –D, która pokazywałaby domyślne wartości dla nowych użytkowników. Polecenie useradd umieszcza nowych użytkowników w pojedynczej grupie i nie tworzy ich katalogów domowych, chyba że zostanie wywołane z opcją –m (w tym przypadku skopiuje ono również plik *.profile* z katalogu */etc/security*).

Przykłady użycia polecenia useradd

Aby utworzyć nowego użytkownika „hilbert", używając systemowych wartości domyślnych w Linuksie, wystarczy po prostu wydać polecenie:

```
$ sudo useradd hilbert
```

Polecenie to powinno utworzyć w pliku */etc/passwd* następujący wpis:

```
hilbert:x:1005:20::/home/hilbert:/bin/sh
```

Polecenie `useradd` wyłączyło konto, wstawiając znak `x` w polu przeznaczonym na hasło. Aby można było korzystać z konta, musisz mu przypisać prawdziwe hasło.

Poniżej przedstawiamy bardziej realistyczny przykład. Określimy, że podstawową grupą użytkownika *hilbert* ma być „faculty" i powinien on również zostać dodany do grupy „famous". Zmienimy też domyślne położenie katalogu domowego oraz powłokę i poprosimy `useradd` o utworzenie katalogu domowego, jeśli jeszcze nie istnieje:

```
$ sudo useradd -c "David Hilbert" -d /home/math/hilbert -g faculty –G famous -m -s
↪/bin/tcsh hilbert
```

Polecenie to utworzy następujący wpis w pliku *passwd*:

```
hilbert:x:1005:30:David Hilbert:/home/math/hilbert:/bin/tcsh
```

Przypisany numer UID jest większy niż najwyższy UID w systemie, a odpowiedni wpis w pliku *shadow* to:

```
hilbert:!:14322:0:99999:7:0::
```

Znaki zastępcze dla hasła w plikach *passwd* i *shadow* mogą różnić się w zależności od systemu operacyjnego. Do czasu nadania hasła dla konta znak zastępczy w pliku *shadow* blokuje możliwość zalogowania. Ponadto polecenie `useradd` doda użytkownika *hilbert* do odpowiednich grup w pliku */etc/group*, utworzy katalog */home/math/hilbert* i skopiuje do niego pliki z katalogu */etc/skel*.

7.6. HURTOWE DODAWANIE UŻYTKOWNIKÓW ZA POMOCĄ NEWUSERS (LINUX)

W systemie Linux istnieje polecenie `newusers`, za pomocą którego jednocześnie utworzysz wiele kont na podstawie zawartości pliku tekstowego. Jest ono dość ułomne, ale może okazać się przydatne, gdy musisz dodać większą liczbę użytkowników naraz, np. przy tworzeniu kont dla całego rocznika studentów. Polecenie `newusers` oczekuje pliku wejściowego zawierającego takie same wiersze jak plik */etc/passwd*, z tym, że pole z hasłem musi zawierać hasło początkowe zapisane jawnym tekstem, dlatego dobrze pilnuj tego pliku.

Polecenie `newusers` uznaje parametr wygasania haseł ustawiony w pliku */etc/login.defs*, ale w odróżnieniu od `useradd` nie kopiuje domyślnych plików startowych. Jednym plikiem startowym, jaki zapewnia, jest *.xauth*.

Dla uniwersytetu potrzebny jest skrypt wsadowy `adduser`, który na podstawie listy studentów pobranej z danych rekrutacyjnych lub rejestracyjnych wygeneruje plik wejściowy dla `newusers`, w którym użytkownicy będą mieli przypisane niepowtarzalne i zgodne z lokalnie przyjętymi regułami nazwy kont, silne, losowo wygenerowane hasła oraz numery UID i GID przyporządkowane rosnąco dla każdego użytkownika. Prawdopodobnie lepiej będzie napisać własny skrypt osłonowy dla `useradd` w języku Perl lub Python, niż zmusić do działania polecenie `newusers`, aby robiło dokładnie to, czego oczekujesz.

7.7. USUWANIE UŻYTKOWNIKÓW

Gdy użytkownik opuszcza organizację, jego konto i pliki powinny zostać usunięte z systemu. Procedura ta wymaga usunięcia wszystkich odwołań do nazwy użytkownika, które były dodane przez Ciebie lub przez program `useradd`. Jeśli usuwasz użytkownika ręcznie, możesz posłużyć się poniższą listą czynności do wykonania:

- usuń użytkownika ze wszystkich spisów telefonów i lokalnych baz danych o użytkownikach,
- usuń użytkownika z pliku *aliases* lub dodaj adres do przekazywania poczty,
- usuń plik `crontab` użytkownika i wszystkie jego zadania `at` lub kolejki wydruku,
- zabij wszystkie działające procesy użytkownika,
- usuń użytkownika z plików *passwd*, *shadow*[12], *group* i *gshadow*,
- usuń katalog domowy użytkownika,
- usuń skrzynkę pocztową użytkownika,
- usuń jego wpisy z kalendarzy grupowych, systemów rezerwacji pomieszczeń itd.,
- usuń lub przekaż uprawnienia do list dyskusyjnych prowadzonych przez usuniętego użytkownika.

Zanim usuniesz czyjś katalog domowy, pamiętaj o tym, aby przenieść w inne miejsce wszelkie pliki, które mogą być potrzebne innym użytkownikom. Zazwyczaj trudno określić, o jakie pliki może chodzić, dlatego przed ich usunięciem zawsze warto wykonać dodatkową kopię zapasową katalogu domowego użytkownika i skrzynki pocztowej.

Po usunięciu użytkownika możesz sprawdzić, czy jego stary numer UID nie jest właścicielem plików w systemie. Aby wyszukać takie osierocone pliki, możesz użyć polecenia

[12] */etc/security/{passwd,group}* w systemie AIX.

`find` z argumentem `–nouser`. Ponieważ polecenie `find` potrafi „uciec" na serwery sieciowe, jeśli nie zachowasz ostrożności, zwykle najbezpieczniej sprawdzać osobno poszczególne systemy plików za pomocą opcji `–xdev`:

```
$ sudo find system_plików -xdev -nouser
```

Zabicie działających procesów usuniętego użytkownika może być nieco utrudnione w środowiskach rozproszonych. W kalendarzach grupowych i systemach rezerwacji pomieszczeń mogą istnieć aktywne wpisy wprowadzone przez nieistniejącego już użytkownika i należy je stamtąd usunąć. Prawdopodobnie w swoim środowisku znajdziesz więcej tego rodzaju miejsc, z których trzeba usunąć ślady po użytkowniku — zrób własną listę, np. w formie skryptu czyszczącego.

Jeśli w Twojej organizacji przydziela się użytkownikom własne stacje robocze, na ogół najprostszym i najbardziej wydajnym rozwiązaniem jest powtórne odtworzenie całego systemu z głównego obrazu przed przekazaniem go nowemu użytkownikowi. Przed przeprowadzeniem nowej instalacji warto jednak wykonać kopię lokalnych plików znajdujących się na dysku twardym na wypadek, gdyby były potrzebne w przyszłości.

W każdym z naszych przykładowych systemów występuje polecenie `userdel`, które automatyzuje proces usuwania użytkownika. Najprawdopodobniej nie przeprowadzi ono tego zdania tak dokładnie, jak byś sobie tego życzył, chyba że skrupulatnie wprowadzisz do niego nowe funkcje, rozszerzające liczbę miejsc, w których przechowywane są informacje związane z użytkownikiem.

W systemie Ubuntu `deluser` to skrypt Perla, który wywołuje tradycyjne polecenie `userdel`; cofa on wszystkie operacje wykonane przez `adduser`. Aby ułatwić lokalizację ustawień, wywołuje skrypt `deluser.local`, jeśli taki istnieje. Plik konfiguracyjny */etc/deluser.conf* umożliwia ustawienie następujących opcji.

- Czy usunąć katalog domowy i skrzynkę pocztową użytkownika?
- Czy wykonać kopię plików użytkownika i gdzie je umieścić?
- Czy usunąć wszystkie pliki należące do użytkownika?
- Czy usunąć grupę, jeśli nie ma już żadnych członków?

SUSE obsługuje zestaw skryptów do wykonywania zadań wstępnych i końcowych, a także skrypt `userdel.local`, który wspomaga polecenie `userdel`, informując domyślne narzędzie o lokalnych ustawieniach. Możesz je skonfigurować w pliku */etc/login.defs*.

W systemie Red Hat istnieje skrypt `userdel.local`, ale nie ma żadnych skryptów do wykonywania zadań wstępnych i końcowych, które automatyzowałyby takie rzeczy jak wykonywanie kopii plików użytkownika przeznaczonego do usunięcia.

Systemy Solaris i AIX mają kilka dodatkowych miejsc, w których przechowywane są informacje o użytkownikach, przede wszystkich chodzi

tu o pliki kontrolujące role i klasy autoryzacji. Dlatego też polecenia `userdel` w tych systemach mają nieco więcej pracy do wykonania przy czyszczeniu wszystkich odwołań do usuniętego użytkownika.

Przykładowo, oprócz */etc/passwd* i */etc/group*, polecenie `userdel` w systemie Solaris aktualizuje również pliki */etc/shadow*, */etc/project* i */etc/user_attr*. W systemie AIX polecenie `userdel` modyfikuje w katalogu */etc/security* następujące pliki: *user*, *user.roles*, *lastlog*, *environ*, *audit/config*, *limits*, *passwd* i *group*. Solaris nie jest tak skrupulatny, jak można by oczekiwać: jego polecenie `userdel` pozostawiło testowy login z profilem skonfigurowanym w pliku *user_attr*, który powinien zostać wyczyszczony.

W systemie HP-UX polecenie `userdel` to banalny, prosty skrypt, który usuwa zmiany wprowadzone przez `useradd`. Modyfikuje tylko pliki *passwd*, *shadow* i *group*.

7.8. WYŁĄCZANIE KONT

Czasami zdarza się, że konto użytkownika musi zostać tymczasowo wyłączone. Najprostszym sposobem jest dopisanie gwiazdki lub innego znaku na początku zaszyfrowanego hasła użytkownika w plikach */etc/security/passwd* (AIX) lub */etc/shadow*. Uniemożliwi to większość prób dostępu na hasło, ponieważ tak zmodyfikowanego hasła nie da się sensownie odszyfrować. Jednak takie polecenia jak `ssh`, które niekoniecznie muszą sprawdzać hasła systemowe, mogą nadal funkcjonować.

We wszystkich dystrybucjach Linuksa polecenia `usermod -L` *`użytkownik`* i `usermod -U` *`użytkownik`* umożliwiają łatwe zablokowanie i odblokowanie hasła. Jest to po prostu szybszy sposób wykonania na haśle opisanych powyżej operacji: opcja `-L` umieszcza znak `!` na początku zaszyfrowanego hasła w pliku */etc/shadow*, a opcja `-U` go usuwa.

W systemie Solaris użytkownik *root* może zablokować konto za pomocą polecenia `passwd -l` *`nazwa_użytkownika`*, zmusić użytkownika do zmiany hasła za pomocą opcji `-f` lub odblokować konto za pomocą opcji –u. Zablokowanie konta powoduje wstawienie znaków `*LK*` do pola z hasłem w pliku */etc/shadow*. Jest to również wartość ustawiana dla użytkowników przez polecenie `useradd`.

System HP-UX obsługuje tylko hasła zaszyfrowane algorytmem `crypt`. Znak `*` nigdy nie występuje w polu hasła wygenerowanym przez `crypt`, dlatego dopisanie go do zaszyfrowanego hasła uniemożliwi zalogowanie się użytkownika.

W systemie AIX konto jest zablokowane, jeśli plik */etc/passwd* zamiast znaku `!` zawiera `*`. Za pomocą polecenia `pwdadm` można wymusić na użytkowniku zmianę hasła lub zablokować jego konto, tak aby tylko administrator mógł zmienić hasło.

Niestety, zmodyfikowanie hasła użytkownika po prostu uniemożliwi logowanie. Użytkownik nie zostanie powiadomiony o zawieszeniu konta ani poinformowany o powodach,

dla których konto nie działa. Innym sposobem wyłączenia konta jest zastąpienie powłoki użytkownika programem, który wyświetli komunikat z wyjaśnieniem i wytłumaczy, co należy zrobić w takiej sytuacji. Następnie program zakończy działanie, zamykając sesję logowania.

Takie podejście ma zarówno zalety, jak i wady. Nie zostaną wyłączone te formy dostępu, które sprawdzają hasło, ale nie zwracają uwagi na powłokę. Aby ułatwić przeprowadzenie sztuczki z wyłączeniem powłoki, wiele demonów zapewniających dostęp bez logowania do systemu (np. `ftpd`) sprawdza, czy powłoka logowania użytkownika jest wymieniona w pliku */etc/shells*; jeśli jej tam nie ma, następuje odmowa dostępu. O takie zachowanie nam chodzi. Niestety, nie jest ono powszechne, jeśli więc zdecydujesz się na wyłączanie kont przez modyfikowanie powłoki, będziesz musiał przeprowadzić dość wyczerpujące testy.

Kolejny problem polega na tym, że jeśli użytkownik spróbuje się zalogować przez system X Window lub za pośrednictwem emulatora terminala, który nie pozostawia widocznego wyjścia po wylogowaniu, może nigdy nie zobaczyć Twojego starannie napisanego wyjaśnienia powodów zawieszenia konta.

Domyślnie program `sendmail` nie dostarczy poczty użytkownikowi, którego powłoka nie występuje w pliku */etc/shell*. Zakłócanie przepływu poczty to zły pomysł, nawet jeśli odbiorca i tak nie może jej natychmiast przeczytać. Możesz obejść to domyślne zachowanie programu `sendmail`, dodając do pliku */etc/shells* fałszywą powłokę o nazwie `/SENDMAIL/ANY/SHELL/`.

7.9. ZARZĄDZANIE UŻYTKOWNIKAMI ZA POMOCĄ NARZĘDZI SYSTEMOWYCH

Systemy HP-UX i AIX wyposażono we wszechstronne narzędzia do administracji systemem, które potrafią zarządzać użytkownikami, przynajmniej w podstawowym zakresie. W systemie AIX jest to SMIT (ang. *System Management Interface Tool*), a w HP-UX nosi ono obecnie nazwę `smh` (ang. *System Management Homepage*)[13]. Każde z tych narzędzi zawiera ekrany przeznaczone do dodawania użytkowników i zarządzania nimi, przy wykorzystaniu interfejsu graficznego lub w formie terminala z menu opartym na bibliotece `curses`. Jeśli jesteś zupełnie początkującym administratorem lub wcześniej pracowałeś w całkiem innym systemie operacyjnym, narzędzia te będą sensownym punktem wyjścia dla wielu często wykonywanych zadań administracyjnych.

Program SMIT z systemu AIX ma przydatną opcję: jeśli naciśniesz klawisz *F6*, wyświetlone zostaną polecenia i argumenty zaplanowane do wykonania. Ponadto rejestruje on całą interakcję i przechowuje plik skryptowy zawierający polecenia, które zostały wykonane w Twoim imieniu. Może to być dobre narzędzie do nauki w miarę zapoznawania

[13] We wcześniejszych wersjach systemu HP-UX miało ono nazwę `sam` (ang. *System Administration Manager*).

się z zawiłościami systemu AIX. Oparty na bibliotece `curses` program `smh` z systemu HP-UX oferuje przydatne, jednoznakowe skróty. Są one widoczne na każdej stronie menu i umożliwiają szybkie przejście do potrzebnego polecenia.

7.10. MINIMALIZOWANIE RYZYKA ZA POMOCĄ PAM

Dołączalne moduły uwierzytelniania (PAM, ang. *Pluggable Authentication Module*) omawiane są w rozdziale 22., „Bezpieczeństwo", na stronie 1113. Umożliwiają one centralne zarządzanie systemowymi funkcjami uwierzytelniania za pośrednictwem standardowych procedur bibliotecznych, dzięki czemu takie programy jak `login`, `sudo`, `passwd` i `su` nie muszą dostarczać własnego kodu uwierzytelniającego. Mechanizm PAM zmniejsza ryzyko związane z pisaniem bezpiecznego oprogramowania, umożliwia administratorom określanie zasad bezpieczeństwa w ramach całego ośrodka i definiuje łatwy sposób dodawania do systemu nowych metod uwierzytelniania.

Dodawanie i usuwanie użytkowników nie wiąże się z modyfikowaniem konfiguracji PAM, ale narzędzia biorące w tym udział działają zgodnie z regułami i ograniczeniami PAM. Poza tym wiele parametrów konfiguracyjnych PAM przypomina te, które są używane przez `useradd` lub `usermod`. Jeśli zmienisz parametry w sposób przedstawiony w tym rozdziale, a `useradd` nie będzie na nie zwracać uwagi, sprawdź, czy konfiguracja PAM w systemie nie nadpisuje Twoich nowych wartości.

7.11. SCENTRALIZOWANE ZARZĄDZANIE KONTAMI

Pewne formy scentralizowanego zarządzania kontami są niezbędne w średnich i dużych organizacjach, niezależnie od tego, czy będą to przedsiębiorstwa, uczelnie, czy urzędy państwowe. W całej organizacji użytkownicy potrzebują wygody i bezpieczeństwa przy stosowaniu tej samej nazwy użytkownika, numeru UID i hasła. Administratorzy muszą dysponować scentralizowanym system umożliwiającym natychmiastowe rozpropagowanie zmian (takich jak unieważnienie konta) do każdego miejsca.

Takie scentralizowanie można osiągnąć na wiele sposobów, z których większość (włącznie z systemem Active Directory firmy Microsoft) wymaga zastosowania LDAP (ang. *Lightweight Directory Access Protocol*), przynajmniej w pewnym zakresie. Istnieje wiele dostępnych rozwiązań: od schematycznych instalacji LDAP opartych na oprogramowaniu open source, po wyrafinowane komercyjne systemy zarządzania tożsamością, które kosztują masę pieniędzy.

LDAP a Active Directory

LDAP to uniwersalne repozytorium bazodanowe, które może przechowywać nie tylko dane związane z zarządzaniem użytkownikami, ale i inne typy danych. Używa się w nim hierarchicznego modelu klient-serwer, który może obsługiwać wiele serwerów i jedno-

czesnych klientów. Jedną z wielkich zalet LDAP jako repozytorium dla danych logowania w ramach całego ośrodka jest to, że może ono wymusić unikatowe numery UID i GID we wszystkich systemach. Poza tym dobrze współpracuje z systemem Windows, choć w drugą stronę tylko w niewielkim stopniu.

➲ *Więcej informacji na temat LDAP i jego implementacji można znaleźć w podrozdziale rozpoczynającym się na stronie 899.*

Usługa Active Directory firmy Microsoft korzysta z protokołów LDAP oraz Kerberos i może zarządzać wieloma rodzajami danych włącznie z informacjami o użytkownikach. Jest nieco egotyczna i w interakcji z repozytoriami LDAP systemu Unix i Linux chce pełnić funkcję nadrzędną. Jeśli potrzebujesz pojedynczego systemu uwierzytelniania dla całego ośrodka, w którym oprócz systemów Unix i Linux są również komputery z systemem Windows, prawdopodobnie najprościej będzie przekazać kontrolę Active Directory, a uniksowe i linuksowe bazy danych LDAP wykorzystywać jako serwery drugorzędne.

Aby zaimplementować taką konfigurację, będziesz potrzebować Active Directory i Services for Unix firmy Microsoft lub komercyjnej platformy integracyjnej Active Directory, takiej jak Quest Authentication Services (dawniej Vintela Authorization Services). Virtual Directory firmy Sun może pomóc Ci w integracji różnych systemów autoryzujących (uwierzytelniających).

Każdy z naszych czterech przykładowych systemów ma wbudowaną obsługę LDAP, choć czasem tylko po stronie klienta (np. HP-UX). LDAP często występuje w parze z PAM w celu przeprowadzania uwierzytelnienia.

LDAP jest bazą danych, dlatego przechowywane tu informacje muszą przestrzegać ściśle określonego schematu. Schematy wyrażane są w postaci plików XML, z polami nazw pochodzących z odpowiednich dokumentów RFC; jeśli chodzi o zarządzanie użytkownikami, jest to głównie RFC 2307. Dokładniejsze informacje na ten temat znajdziesz w rozdziale 19., „Współdzielenie plików systemowych".

Systemy pojedynczego logowania

Systemy pojedynczego logowania (SSO, ang. *single sign-on*) zachowują równowagę między wygodą użytkownika a względami bezpieczeństwa. Idea polega na tym, aby użytkownik mógł się uwierzytelnić po jednorazowym zalogowaniu (po znaku zachęty, na stronie WWW lub w oknie systemu Windows). Użytkownik uzyskuje wtedy potwierdzenie tożsamości (zazwyczaj niezauważalnie, więc nie jest tu wymagane aktywne zarządzanie), które może później wykorzystać do uzyskania dostępu do innych komputerów i aplikacji. Użytkownik musi zapamiętać tylko jeden login i hasło zamiast wielu.

Taki schemat dopuszcza stosowanie bardziej złożonych poświadczeń (ponieważ użytkownik nie musi o nich pamiętać ani nawet się nimi zajmować), co teoretycznie zwiększa poziom bezpieczeństwa. Jednak nieuprawnione przejęcie konta może spowodować większe szkody, ponieważ jeden login zapewni napastnikowi dostęp do wielu komputerów i aplikacji. Systemy SSO stanowią duże zagrożenie w sytuacji, gdy użytkownik

odejdzie od komputera, a wciąż będzie zalogowany. Poza tym serwery uwierzytelniające stają się krytycznym wąskim gardłem. Ich awaria może sparaliżować pracę całego przedsiębiorstwa.

Systemy SSO to stosunkowo prosta idea, jednak wdrożenie jej wiąże się z wieloma komplikacjami, ponieważ poszczególne aplikacje i komputery, z których użytkownik może chcieć korzystać, muszą rozumieć proces uwierzytelniania i znać poświadczenia SSO. W niektórych systemach SSO poświadczeniami użytkowników zarządza Kerberos; zostało to omówione dokładniej w rozdziale 22., „Bezpieczeństwo", od strony 1133.

Istnieje kilka systemów SSO na licencji open source:

- JOSSO, serwer SSO na licencji open source napisany w języku Java,
- CAS (ang. *Central Authentication Service*), opracowany na uniwersytecie Yale (również w języku Java),
- Likewise Open, narzędzie integracyjne, które sprawia, że Active Directory dobrze współpracuje z systemami Linux i Unix.

Dostępnych jest również mnóstwo systemów komercyjnych. Większość z nich jest zintegrowana z pakietami zarządzania tożsamością, które zostaną omówione w następnym punkcie.

Systemy zarządzania tożsamością

„Zarządzanie tożsamością" to najnowsze modne hasło w dziedzinie zarządzania użytkownikami. Mówiąc po ludzku, oznacza to identyfikowanie użytkowników Twojego systemu, poświadczanie ich tożsamości i przyznawanie uprawnień na podstawie tych uwierzytelnionych tożsamości. Działania standaryzacyjne w tej dziedzinie prowadzone są przez World Wide Web Consortium i The Open Group.

Komercyjne systemy zarządzania tożsamością łączą kilka kluczowych pojęć systemu Unix w jeden miły dla oka graficzny interfejs użytkownika wypełniony marketingowym żargonem. Podstawą wszystkich tego rodzaju systemów jest baza danych zawierająca uwierzytelnienia użytkowników i dane autoryzacyjne, często zapisana w formacie LDAP. Kontrolę zapewniają takie pojęcia jak grupy systemu Unix, a ograniczone uprawnienia administracyjne wymuszane są za pośrednictwem narzędzi w rodzaju `sudo`. Większość tego rodzaju systemów została zaprojektowana zgodnie z wymogami prawnymi dotyczącymi odpowiedzialności, monitorowania i dzienników kontroli.

Istnieje wiele takich systemów komercyjnych: Identity Management firmy Oracle, Sun Identity Management Suite[14], Courion, Avatier Identity Management Suite (AIMS) i BMC Identity Management Suite, by wymienić tylko kilka z nich. Dokonując oceny systemów zarządzania tożsamością, zwróć uwagę na następujące cechy:

[14] Teraz, gdy firma Sun została przejęta przez Oracle, nie jest pewne, czy system ten będzie nadal oferowany jako osobny produkt.

- generowanie globalnie unikatowych identyfikatorów użytkownika,
- możliwość tworzenia, zmiany i usuwania kont użytkowników w całej organizacji, niezależnie od sprzętu i systemu operacyjnego,
- bezpieczny interfejs WWW do zarządzania z możliwością dostępu z zewnątrz,
- możliwość łatwego wyświetlania wszystkich użytkowników, którzy mają określony zestaw uprawnień,
- prosty sposób podglądu wszystkich uprawnień przydzielonych określonemu użytkownikowi,
- możliwość pozwolenia użytkownikowi na zmianę (i zresetowanie) własnych haseł, z wymuszeniem reguł wyboru silnych haseł,
- możliwość globalnej zmiany swoich haseł przez użytkowników za pomocą jednej operacji,
- mechanizm obiegowej organizacji pracy, np. warstwowy system zatwierdzeń przed przyznaniem użytkownikowi określonych uprawnień,
- możliwość koordynacji z bazą danych działu kadr w celu automatycznego zablokowania dostępu pracownikom, którzy złożyli wypowiedzenie lub zostali zwolnieni,
- konfigurowalne rejestrowanie wszystkich zmian i działań administracyjnych,
- konfigurowalne raporty tworzone na podstawie zarejestrowanych danych (z podziałem na użytkowników, daty itd.),
- kontrola dostępu oparta na rolach, w tym obsługa kont użytkowników za pomocą ról,
- interfejs, za pomocą którego osoba odpowiedzialna za rekrutację może poprosić o skonfigurowanie kont zgodnie z określoną rolą,
- wyjątki w działaniach opartych na rolach, łącznie z obiegiem określającym proces zatwierdzania wyjątków.

Weź również pod uwagę sposób wdrożenia systemu w miejscach, gdzie odbywa się właściwa autoryzacja i uwierzytelnianie. Czy system wymaga zainstalowania wszędzie własnego agenta, czy dopasowuje się do podległych mu systemów?

7.12. ZALECANA LITERATURA

„The Complete Buyer's Guide for Identity Management", dokumentacja firmy Sun Microsystems, 2008 r., *sun.systemnews.com/articles/129/4/sec/20930*.

7.13. ĆWICZENIA

Ćwiczenie 7.1. W jaki sposób ustalana jest domyślna grupa użytkownika? Jak można ją zmienić?

Ćwiczenie 7.2. Wyjaśnij różnice między następującymi wartościami `umask`: 077, 027, 022 i 755. W jaki sposób ustawiłbyś jedną z tych wartości jako domyślną maskę dla nowych użytkowników w ramach całej organizacji? Czy możesz wymusić na swoich użytkownikach używanie standardowej wartości `umask`?

Ćwiczenie 7.3. Do czego służy plik przesłaniania haseł (*shadow*)?

★ Ćwiczenie 7.4. Określ system uwierzytelniania używany przez program `login` w Twoim systemie. Jeśli korzysta z PAM, ustal, jakie inne programy w systemie również z niego korzystają.

★ Ćwiczenie 7.5. Wymień czynności wymagane w celu dodania użytkownika do systemu bez użycia programu `useradd`. Jakie dodatkowe czynności są wymagane w Twoim lokalnym środowisku?

★ Ćwiczenie 7.6. Ustal konwencję nazewniczą dla nowych użytkowników w Twojej organizacji. Jakie rządzą nią reguły? W jaki sposób zapewniana jest niepowtarzalność nazw? Czy możesz wskazać jakieś wady? Jak są usuwani użytkownicy?

★★ Ćwiczenie 7.7. Znajdź listę nazwisk (np. z lokalnego spisu telefonów dostępnego on-line) i wykorzystaj je jako dane wejściowe dla skryptu tworzącego nazwy użytkowników zgodnie z konwencją nazewniczą przyjętą w Twojej organizacji. Ilu użytkowników udało Ci się dodać przed wystąpieniem kolizji? Ile kolizji było ogółem? Wykorzystaj te dane do oceny konwencji nazewniczej w Twojej organizacji i zaproponuj usprawnienia.

★★ Ćwiczenie 7.8. Napisz skrypt pomocny w monitorowaniu poprawności pliku */etc/passwd* (punkty b i e wymagają dostępu do konta użytkownika *root*, chyba że jesteś zdolny):

a) znajdź wszystkie wpisy z identyfikatorem UID równym 0,
b) znajdź wszystkie wpisy bez hasła (wymaga dostępu do pliku */etc/shadow*),
c) znajdź wszystkie zestawy wpisów o zduplikowanych identyfikatorach UID,
d) znajdź wszystkie wpisy ze zduplikowanymi nazwami użytkownika,
e) znajdź wszystkie wpisy bez daty wygaśnięcia (wymaga dostępu do pliku */etc/shadow*).

★★★★★ Ćwiczenie 7.9. Napisz moduł PAM, który przeprowadza uwierzytelnianie za pomocą losowo generowanego kodu PIN, wysyłając go na telefon komórkowy użytkownika jako wiadomość SMS, a następnie prosi o wpisanie tego kodu w celu weryfikacji. Zainstaluj swój moduł i skonfiguruj go w stosie logowania PAM, aby uzyskać dwustopniowe uwierzytelnianie.

SKOROWIDZ

B

C

D

E

F

G

H

I

J

K

L

Ł

M

N

O

P

Q

R

S

Ś

T

U

V

W

X

Y

Z

Ta książka to obowiązkowa pozycja dla wszystkich administratorów!

Systemy Unix i Linux wykorzystywane są wszędzie tam, gdzie wymagana jest najwyższa niezawodność, wydajność i elastyczność. Ich potencjał, możliwości oraz odporność na niesprzyjające otoczenie sprawiły, że są dominującymi systemami operacyjnymi w zastosowaniach sieciowych. Jeżeli jednak systemy te kojarzą Ci się wyłącznie z czarnym ekranem konsoli oraz trybem tekstowym, jesteś w błędzie. Już od dawna systematycznie zwiększają one swój udział w zastosowaniach domowych. Co w sobie kryją? Jak nimi administrować i jak zwiększać ich wydajność?

Na te i setki innych pytań odpowiada ten obszerny przewodnik. Autorzy rozłożyli na czynniki pierwsze i omówili każde zagadnienie, z którym możesz mieć styczność w codziennej pracy. Po lekturze tej książki będziesz posiadał obszerną wiedzę w tej materii. Instalacja systemu, skrypty powłoki, kontrolowanie procesów, konfiguracja uprawnień to zadania, które nie będą Ci sprawiały żadnych problemów. Ponadto dowiesz się, jak zarządzać użytkownikami, przestrzenią dyskową, zadaniami okresowymi oraz backupami. Dużą uwagę powinieneś poświęcić rozdziałom omawiającym zagadnienia sieciowe. Jest pewne, że serwer, który trafi w Twoje ręce, będzie oferował wiele usług dostępnych przez sieć. Protokoły routingu, usługa DNS, firewall, serwer plików to elementy, które niejednokrotnie będziesz musiał konfigurować.

Nie znajdziesz na rynku obszerniejszej książki poświęconej systemom Unix i Linux. Jeżeli jesteś z nimi związany w pracy zawodowej, interesujesz się systemami operacyjnymi lub korzystasz z nich w domu, ta książka obowiązkowo musi znaleźć się na Twojej półce!

- Skrypty powłoki
- Uruchamianie i zamykanie systemu
- Kontrola dostępu, uprawnienia administratora
- Nadzór nad procesami
- System plików
- Zarządzanie użytkownikami
- Wykorzystanie zasobów dyskowych
- Zadanie okresowe — cron
- Tworzenie kopii zapasowych i odtwarzanie z nich
- Logi systemowe
- Instalacja oprogramowania i zarządzanie nim
- Sterowniki i jądro systemu
- Konfiguracja dostępu do sieci
- Wyznaczanie tras — routing
- Sprzęt sieciowy
- Rozwiązywanie nazw — DNS
- Sieciowe systemy plików
- Poczta elektroniczna
- Zarządzanie siecią, protokół SNMP
- Hosting WWW
- Graficzny interfejs użytkownika — X Window System
- Podstawy centrów danych

Nr katalogowy 64682

Księgarnia internetowa:
http://helion.pl

Zamówienia telefoniczne:
0 801 339900
0 601 339900

helion.pl
helion sa

Sprawdź najnowsze promocje:
http://helion.pl/promocje
Książki najchętniej czytane:
http://helion.pl/bestsellery
Zamów informacje o nowościach:
http://helion.pl/nowosci

Helion
Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
http://helion.pl

PRENTICE HALL

KOD KORZYŚCI

Cena 199,00 zł

ISBN 978-83-246-2968-8

9 788324 629688

Informatyka w najlepszym wydaniu