Donald A. Tevault

Bezpieczeństwo systemu **Linux**

Hardening i najnowsze techniki zabezpieczania przed cyberatakami

Wydanie III

‹packt›

Helion

Tytuł oryginału: Mastering Linux Security and Hardening: A practical guide to protecting your Linux system from cyber attacks, 3rd Edition

Tłumaczenie: Magdalena A. Tkacz

ISBN: 978-83-289-0292-3

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
https://helion.pl/user/opinie/besyl3
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *https://helion.pl* (księgarnia internetowa, katalog książek)

Printed in Poland.

Spis treści

ROZDZIAŁ 3.

ROZDZIAŁ 7.
Utwardzanie SSH 236

CZĘŚĆ 2.
Kontrola dostępu do plików i katalogów (DAC)

CZĘŚĆ 3.
Zaawansowane techniki utwardzania systemu

Zabezpieczanie kont zwykłych użytkowników

Rozdział 3

Zarządzanie użytkownikami jest jednym z trudniejszych aspektów administracji IT. Trzeba mieć pewność, że użytkownicy zawsze mają dostęp do swoich zasobów i do tych wymaganych do wykonania przydzielonych im zadań. Jednocześnie trzeba mieć także pewność, że zasoby użytkowników są zawsze zabezpieczone przed nieautoryzowanym dostępem. W tym rozdziale przyjrzymy się, jak skonfigurować konta użytkowników i jak zabezpieczyć ich dane uwierzytelniające tak, aby chronić je przed atakującymi i szpiegami. Na zakończenie przyjrzymy się kilku scentralizowanym systemom zarządzania użytkownikami. A zatem w tym rozdziale:

- Zabezpieczanie katalogów domowych użytkowników.
- Egzekwowanie kryteriów dla silnych haseł.
- Ustawianie i egzekwowanie wygasania haseł i kont.
- Zapobieganie atakom siłowym (ang. *brute-force*) na hasła.
- Blokowanie kont użytkowników.
- Konfigurowanie wiadomości powitalnej z ostrzeżeniem (ang. *security banner*).
- Wykrywanie ujawnionych (ang. *compromised*) haseł.
- Zrozumienie centralnych systemów zarządzania użytkownikami.

Zabezpieczanie katalogów domowych użytkowników w systemach Red Hat

Jest to kolejny obszar, w którym różne rodziny dystrybucji Linuksa różnią się od siebie. Jak zobaczymy, każda rodzina dystrybucji ma inne domyślne ustawienia zabezpieczeń dla katalogów domowych użytkowników. Administrator bezpieczeństwa, który nadzoruje mieszane środowisko złożone z różnych linuksowych dystrybucji, musi to wziąć pod uwagę.

Tradycyjnie Red Hat Enterprise Linux i wszystkie wywodzące się z niego dystrybucje, takie jak CentOS i AlmaLinux, mają lepsze zabezpieczenia bezpośrednio po zainstalowaniu niż jakakolwiek inna rodzina dystrybucji Linuksa. To sprawia, że utwardzanie systemów typu Red Hat jest szybsze i łatwiejsze, ponieważ większość pracy została już wykonana. Jedną z rzeczy, która została już dla nas zrobiona, jest zablokowanie dostępu do katalogów domowych użytkowników:

```
[donnie@localhost home]$ sudo useradd charlie
[sudo] password for donnie:
[donnie@localhost home]$
[donnie@localhost home]$ ls -l
total 0
drwx------. 2 charlie charlie 59 Oct 1 15:25 charlie
drwx------. 2 donnie donnie 79 Sep 27 00:24 donnie
drwx------. 2 frank frank 59 Oct 1 15:25 frank
[donnie@localhost home]$
```

Domyślnie narzędzie `useradd` w systemach typu Red Hat tworzy katalogi domowe użytkowników z ustawieniem uprawnień `700`. Oznacza to, że tylko użytkownik, który jest właścicielem katalogu domowego, może uzyskać do niego dostęp. Wszyscy inni zwykli użytkownicy mają zablokowany dostęp. Dlaczego tak jest, możesz zobaczyć, zerkając do pliku */etc/login.defs*. Na maszynie wirtualnej CentOS 7 przewiń plik do dołu, a zobaczysz to:

```
CREATE_HOME yes
UMASK 077
```

W pliku *login.defs* dystrybucji typu RHEL 8 lub RHEL 9, takiej jak AlmaLinux, zobaczysz, że ustawienie `UMASK` daje szerokie uprawnienia, co wydaje się nieco dziwne. Oto jak wygląda:

```
UMASK           022
```

Ale kilka linijek poniżej zobaczysz zupełnie nową dyrektywę, której nigdy wcześniej nie było, a wygląda ona następująco:

```
HOME_MODE    0700
```

Więc nawet jeśli `UMASK` pozwala na dostęp, katalogi domowe nowych użytkowników nadal są odpowiednio zabezpieczone.

Plik *login.defs* jest jednym z dwóch plików, w których konfigurowane są domyślne ustawienia polecenia `useradd`. Linie `UMASK` lub `HOME_MODE` określają wartości uprawnień dla tworzonych katalogów domowych. Dystrybucje typu Red Hat są skonfigurowane tak, aby usunąć wszystkie uprawnienia dla grupy i dla *others*. Linia `HOME_MODE` lub `UMASK` znajduje się w pliku *login.defs* we wszystkich dystrybucjach Linuksa, ale do niedawna dystrybucje typu Red Hat były jedynymi, które domyślnie miały je ustawione na tak restrykcyjną wartość. Większość dystrybucji innych niż Red Hat ma zwykle wartość `UMASK` ustawioną jako `022`, a katalogi domowe tworzy z uprawnieniami o wartości `755`. Przez to każdy może wejść do katalogów domowych innych użytkowników i uzyskać dostęp do ich plików.

Zabezpieczanie katalogów domowych użytkowników w systemach Debian i Ubuntu

Debian i jego potomkowie, tacy jak Ubuntu, mają dwa narzędzia do tworzenia kont użytkowników:

- `useradd`,
- `adduser`.

Przyjrzyjmy się obu narzędziom.

useradd w Debian/Ubuntu

W systemach Debian/Ubuntu narzędzie `useradd` jest dostępne, ale nie jest dostarczane z wygodnymi, wstępnie skonfigurowanymi domyślnymi ustawieniami, tak jak ma to miejsce w dystrybucjach rodziny Red Hat. Gdybyś wydał polecenie `sudo useradd frank` na maszynie z Debianem/Ubuntu, katalog domowy dla Franka nie zostałby utworzony i miałby on przypisaną niewłaściwą domyślną powłokę. Jeśli chciałbyś utworzyć konto użytkownika za pomocą `useradd` w systemie Debian lub Ubuntu, polecenie musiałoby wyglądać mniej więcej tak:

```
sudo useradd -m -d /home/frank -s /bin/bash frank
```

A teraz po kolei, co to wszystko oznacza:

- `-m` tworzy katalog domowy.
- `-d` określa katalog domowy.
- `-s` określa domyślną powłokę dla Franka. (Bez `-s` Debian/Ubuntu przypisałby Frankowi powłokę `/bin/sh`).

Kiedy spojrzysz na katalogi domowe na komputerze z Debianem lub Ubuntu 20.04, zobaczysz, że są one dostępne i wszyscy mają uprawnienia do wykonywania i odczytu:

```
donnie@packt:/home$ ls -l
total 8
drwxr-xr-x 3 donnie donnie 4096 Oct 2 00:23 donnie
drwxr-xr-x 2 frank frank 4096 Oct 1 23:58 frank
donnie@packt:/home$
```

Jak widzisz, zarówno Frank, jak i ja możemy dostać się do swoich zasobów. (Ale nie, ja nie chcę, żeby Frank miał dostęp do moich). Każdy użytkownik mógłby zmienić uprawnienia w swoim katalogu, ale ilu z Twoich użytkowników wiedziałoby, jak to zrobić? Więc naprawmy to:

```
cd /home
sudo chmod 700 *
```

Zobacz, co mamy teraz:

```
donnie@packt:/home$ ls -l
total 8
drwx------ 3 donnie donnie 4096 Oct 2 00:23 donnie
drwx------ 2 frank frank 4096 Oct 1 23:58 frank
donnie@packt:/home$
```

Teraz wygląda to znacznie lepiej.

Aby zmienić domyślne ustawienia uprawnień dla katalogów domowych, otwórz plik */etc/login.defs* do edycji. Poszukaj tej linii:

```
UMASK 022
```

Zmień ją na:

```
UMASK 077
```

Teraz katalogi domowe nowych użytkowników zostaną zabezpieczone już podczas tworzenia, tak jak ma to miejsce w przypadku dystrybucji rodziny Red Hat.

W Ubuntu 22.04 sprawy mają się inaczej. Deweloperzy Ubuntu w końcu zdali sobie sprawę, że katalogi domowe użytkowników powinny być domyślnie niedostępne dla innych. Tak więc ustawienie HOME_MODE w pliku *login.defs* Ubuntu 22.04 wygląda teraz następująco:

```
HOME_MODE       0750
```

Obejmuje to uprawnienia dostępu dla grupy osobistej użytkownika, ale to nie jest problem. Nadal bowiem oznacza to, że tylko właściciele katalogów domowych mogą się dostać do swoich katalogów.

adduser w systemach Debian/Ubuntu

Narzędzie adduser pozwala tworzyć w sposób interaktywny konta użytkowników i ustawiać ich hasła. Można to wykonać za pomocą pojedynczego polecenia, unikalnego dla rodziny dystrybucji Linuksa wywodzących się z Debiana. Większość domyślnych ustawień, których brakuje w implementacji useradd dla Debiana, jest już w adduser. W systemie Debian i Ubuntu 20.04 tworzy on katalogi domowe użytkowników z liberalnymi uprawnieniami 755 (w Ubuntu 22.04 tworzy odpowiednio zabezpieczone katalogi domowe z bardziej restrykcyjną wartością dla uprawnień — 750). Na szczęście ustawienie mniej restrykcyjne można łatwo zmienić. Ustawia się to w pliku */etc/adduser.conf*, w okolicy linii 56:

```
DIR_MODE=750
```

Jak wspomniałem wcześniej, w Ubuntu 20.04 będzie tam wartość 755. Aby założyć blokadę, po prostu zmień tę wartość na 750.

Chociaż adduser jest przydatny przy codziennym tworzeniu kont użytkowników, to nie oferuje takiej elastyczności jak useradd i nie nadaje się do użycia w skryptach powłoki. Czymś, co można zrobić z adduser, a nie można z useradd, jest automatyczne szyfrowanie katalogu domowego użytkownika podczas tworzenia konta. Aby to zadziałało, musisz najpierw zainstalować pakiet *ecryptfs-utils*. Aby utworzyć konto z zaszyfrowanym katalogiem domowym dla Cleopatry, wpisz poniższe polecenia:

```
donnie@ubuntu-steemnode:~$ sudo apt install ecryptfs-utils
donnie@ubuntu-steemnode:~$ sudo adduser --encrypt-home cleopatra
[sudo] password for donnie: // hasło dla donnie:
Adding user 'cleopatra' ...
Adding new group 'cleopatra' (1004) ...
Adding new user 'cleopatra' (1004) with group 'cleopatra' ...
Creating home directory '/home/cleopatra' ...
Setting up encryption ...
**********************************************************************
YOU SHOULD RECORD YOUR MOUNT PASSPHRASE AND STORE IT IN A SAFE LOCATION.
ecryptfs-unwrap-passphrase ~/.ecryptfs/wrapped-passphrase
THIS WILL BE REQUIRED IF YOU NEED TO RECOVER YOUR DATA AT A LATER TIME.
**********************************************************************
Done configuring.
Copying files from '/etc/skel' ...
Enter new UNIX password: // Wprowadź nowe hasło UNIX:
```

```
Retype new UNIX password: // Wpisz ponownie nowe hasło UNIX:
passwd: password updated successfully
Changing the user information for cleopatra
Enter the new value, or press ENTER for the default // Wprowadź nową wartość
lub naciśnij ENTER, aby wyświetlić domyślną
Full Name []: Cleopatra Tabby Cat
Room Number []: 1
Work Phone []: 555-5556
Home Phone []: 555-5555
Other []:
Is the information correct? [Y/n] Y
donnie@ubuntu-steemnode:~$
```

Przy pierwszym logowaniu Cleopatra będzie musiała uruchomić polecenie `ecryptfs-unwrap-passphrase`, o którym mowa w poprzednim komunikacie. Następnie dobrze by było, gdyby zapisała swoje hasło i przechowywała je w bezpiecznym miejscu:

```
cleopatra@ubuntu-steemnode:~$ ecryptfs-unwrap-passphrase
Passphrase:
d2a6cf0c3e7e46fd856286c74ab7a412
cleopatra@ubuntu-steemnode:~$
```

Jeśli chodzi o szyfrowanie, będzie ono omówione bardziej szczegółowo w rozdziale poświęconym szyfrowaniu.

Ćwiczenie: szyfrowanie katalogu domowego za pomocą adduser

W tym ćwiczeniu będziesz pracować z narzędziem `adduser` na maszynie wirtualnej Ubuntu 22.04:

1. Zainstaluj pakiet *ecryptfs-utils*:

   ```
   sudo apt install ecryptfs-utils
   ```

2. Utwórz dla Cleopatry konto użytkownika z zaszyfrowanym katalogiem domowym, a następnie wyświetl wyniki:

   ```
   sudo adduser --encrypt-home cleopatra
   ls -l /home
   ```

3. Zaloguj się jako Cleopatra i uruchom polecenie `ecryptfs-unwrap-passphrase`:

   ```
   su - cleopatra
   ecryptfs-unwrap-passphrase
   exit
   ```

Zwróć uwagę, że niektóre informacje, o które prosi `adduser`, są opcjonalne: dla tych elementów możesz po prostu nacisnąć klawisz *Enter*.

Wymuszenie korzystania z silnych haseł

Może nie pomyślałbyś, że tak niewinnie brzmiący temat, jak kryteria dla silnego **hasła** (ang. *password*), może być kontrowersyjny, ale niestety taki jest. Zapewne stale słyszałeś:

- Twórz hasła o określonej minimalnej długości.
- Twórz hasła składające się z kombinacji wielkich i małych liter, cyfr i znaków specjalnych.
- Upewnij się, że hasła nie zawierają żadnych słów, które można znaleźć w słowniku lub które są oparte na danych osobowych użytkowników.
- Zmuszaj użytkowników do regularnej zmiany haseł.

Ale korzystając z ulubionej wyszukiwarki, zobaczysz, że zdania ekspertów są podzielone, jeśli chodzi o szczegóły tych zaleceń. Na przykład znajdziesz dyskusje, czy hasła powinny być zmieniane co 30, 60 lub 90 dni, czy wszystkie cztery typy znaków muszą znajdować się w haśle, a nawet takie, które dotyczą minimalnej długości hasła. Najbardziej interesujące jest to, że źródłem wszystkich kontrowersji jest jedna osoba — ta, która wymyśliła poprzednie kryteria. Teraz osoba ta mówi, że to wszystko bzdury, i żałuje, że je wymyśliła. Teraz uważa, że powinniśmy używać **haseł będących zdaniami** (ang. *passphrase*) — są one długie, ale łatwe do zapamiętania. Mówi również, że powinny być zmieniane tylko wtedy, gdy zostały ujawnione.

Uwaga

Bill Burr, były inżynier National Institute of Standards and Technology (NIST), który stworzył przedstawione przeze mnie wcześniej kryteria silnego hasła, udostępnił swoje przemyślenia na temat tego, dlaczego teraz zmienia swoją opinię. Jeśli chcesz się dowiedzieć więcej, odwiedź *https://www.pcmag.com/news/355496/you-might-not-need-complex-alphanumeric-passwords-after-all*.

Od czasu opublikowania oryginalnego wydania tej książki NIST zgodził się z Billem Burrem. Instytut zmienił swoje kryteria implementacji haseł, aby dostosować je do zaleceń Burra. Możesz o tym przeczytać na stronie: *https://www.riskcontrolstrategies.com/2018/01/08/new-nist-guidelines-wrong/*.

Niezależnie od tego wszystkiego wiele organizacji nadal jest przywiązanych do idei używania złożonych haseł, które regularnie wygasają, będziesz więc musiał przestrzegać tych zasad, jeśli nie uda Ci się ich przekonać, że można inaczej. Dodatkowo, jeśli używasz tradycyjnych haseł, powinny być one wystarczająco silne, aby oprzeć się wszelkiego rodzaju atakom na hasła. Zobaczmy, jak wyegzekwować używanie silnych kryteriów dla haseł w systemie Linux.

Wskazówka

Muszę się przyznać, że nigdy wcześniej nie pomyślałem, aby spróbować utworzyć hasło w postaci zdania (ang. *passphrase*) zamiast hasła (ang. *password*) w systemie Linux. Wypróbowałem więc takie podejście na mojej maszynie wirtualnej CentOS, aby sprawdzić, czy to zadziała.

Utworzyłem konto dla Maggie, mojej czarno-białej kotki-pingwinka. Jako hasło wpisałem „Lubię inne kotki". Możesz pomyśleć: „Masakra! To nie spełnia żadnych kryteriów złożoności i używa słów, które można łatwo znaleźć w słowniku. Jak to może być bezpieczne?". Ale faktem jest, że jest to zdanie złożone z odrębnych słów oddzielonych spacjami, co czyni je bezpiecznym i bardzo trudnym do złamania.

W prawdziwym życiu nigdy nie stworzyłbym hasła wyrażającego moją sympatię do kotów, ponieważ nietrudno się dowiedzieć, że jestem kociarzem. Wybrałbym raczej hasło dotyczące jakiejś mało znanej części mojego życia, o której nikt poza mną by nie wiedział. W każdym razie używanie zdań w miejsce tradycyjnych haseł ma dwie zalety: są one trudniejsze do złamania niż tradycyjne hasła, a jednocześnie są łatwiejsze do zapamiętania przez użytkowników. Jednak by zwiększyć trudność ich odgadnięcia, nie twórz (w przypadku zdań, które mają pełnić rolę hasła dostępu) zdań dotyczących takich faktów z Twojego życia, o których wszyscy wiedzą.

Instalowanie i konfigurowanie pwquality

Będziemy używać modułu *pwquality* do uwierzytelniania za pomocą Pluggable Authentication Module (PAM). Jest to nowsza technologia, która zastąpiła stary moduł *cracklib*. W każdym systemie Red Hat 7 (lub nowszym), a także w SUSE i OpenSUSE *pwquality* jest instalowany domyślnie, nawet jeśli wybierzesz instalację minimalną. Jeśli przejdziesz do katalogu */etc/pam.d/*, możesz wykonać operację `grep`, aby sprawdzić, czy pliki konfiguracyjne PAM są już odpowiednio skonfigurowane. `retry=3` oznacza, że użytkownik będzie mógł tylko trzy razy spróbować podać prawidłowe hasło podczas logowania do systemu:

```
[donnie@localhost pam.d]$ grep 'pwquality' *
password-auth:password requisite pam_pwquality.so try_first_pass
local_users_only retry=3 authtok_type=
password-auth-ac:password requisite pam_pwquality.so try_first_pass
local_users_only retry=3 authtok_type=
system-auth:password requisite pam_pwquality.so try_first_pass
local_users_only retry=3 authtok_type=
system-auth-ac:password requisite pam_pwquality.so try_first_pass
local_users_only retry=3 authtok_type=
[donnie@localhost pam.d]$
```

Na Debianie i Ubuntu musisz samodzielnie doinstalować *pwquality*, używając następującego polecenia:

```
sudo apt install libpam-pwquality
```

Reszta procedury jest taka sama dla wszystkich systemów operacyjnych, z którymi tu pracujemy, i polega wyłącznie na edycji pliku */etc/security/pwquality.conf*. Kiedy otworzysz ten plik w edytorze tekstu, zobaczysz, że wszystko jest zakomentowane, co oznacza, że początkowo w systemie nie mają zastosowania żadne kryteria dotyczące złożoności hasła. Zobaczysz również, że opcje są bardzo dobrze udokumentowane: znajdziesz komentarz wyjaśniający przy każdym ustawieniu.

Możesz ustawić kryteria złożoności hasła, jak tylko chcesz, musisz po prostu odkomentować odpowiednie wiersze i ustawić odpowiednie wartości. Przyjrzyjmy się tylko jednemu ustawieniu:

```
# Minimum acceptable size for the new password (plus one if
# credits are not disabled which is the default). (See pam_cracklib manual.)
# Cannot be set to lower value than 6.
# minlen = 8
```

Ustawienie minimalnej długości działa, wykorzystując system punktów (ang. *credit*). Oznacza to, że przy pojawieniu się każdego kolejnego typu znaków w haśle minimalna wymagana długość hasła zostanie zmniejszona o jeden znak. Na przykład ustaw `minlen` na wartość `19` i spróbuj ustawić dla Kasi hasło `turkeylips`:

```
minlen = 19

[donnie@localhost ~]$ sudo passwd kasia
 Changing password for user kasia.
 New password:
 BAD PASSWORD: The password is shorter than 17 characters
 Retype new password:
 [donnie@localhost ~]$
```

Ponieważ małe litery w `turkeylips` liczą się jako jeden typ znaków, więc musimy mieć tylko 18 znaków zamiast 19. Jeśli spróbujemy ponownie, tym razem z `TurkeyLips`, otrzymamy:

```
[donnie@localhost ~]$ sudo passwd kasia
 Changing password for user kasia.
 New password:
 BAD PASSWORD: The password is shorter than 17 characters
 Retype new password:
 [donnie@localhost ~]$
```

Tym razem duża litera *T* oraz *L* liczą się jako drugi typ znaków, więc musimy mieć tylko 17 znaków w haśle.

Tuż pod linią `minlen` zobaczysz linię z punktami. Załóżmy, że nie chcesz, aby małe litery były wliczane do punktacji. Znajdź tę linię:

```
# lcredit = 1
```

Usuń komentarz i zmień `1` na `0`:

```
lcredit = 0
```

Następnie spróbuj przypisać Kasi hasło `turkeylips`:

```
[donnie@localhost ~]$ sudo passwd kasia
 Changing password for user kasia.
 New password:
 BAD PASSWORD: The password is shorter than 19 characters
 Retype new password:
 [donnie@localhost ~]$
```

Tym razem *pwquality* naprawdę żąda hasła mającego 19 znaków. Jeśli ustawimy wartość punktów na wyższą niż 1, otrzymamy więcej punktów za większą liczbę znaków tego samego typu.

Wartości punktów mogą być również liczbami ujemnymi, wtedy oznaczają wymaganą liczbę znaków określonego typu w haśle. Na przykład możemy mieć coś takiego:

```
dcredit = -3
```

To oznacza, że wymagamy co najmniej trzech cyfr w haśle. Jednak korzystanie z tej funkcji jest naprawdę złym pomysłem, ponieważ ktoś, kto przeprowadza atak na hasło, szybko znajdzie wzorzec, którego wymagasz, co pomoże atakującemu precyzyjniej ukierunkować atak. Jeśli chcesz wymagać, aby hasło miało wiele typów znaków, lepiej użyć parametru `minclass`:

 # minclass = 3

Jest on wstępnie ustawiony na wartość 3, co oznacza, że wymagane są znaki trzech różnych typów. Aby użyć tego ustawienia, wystarczy usunąć symbol komentarza.

Reszta parametrów w *pwquality.conf* działa prawie tak samo, a przy każdym znajdziesz dobrze napisany komentarz wyjaśniający, za co odpowiada.

Wskazówka

Jeśli użyjesz uprawnień `sudo` do ustawienia hasła dla innej osoby, system będzie protestował, jeśli utworzysz hasło, które nie spełnia kryteriów złożoności, ale będziesz mógł to zrobić. Jeśli zwykły użytkownik próbowałby zmienić swoje własne hasło bez uprawnień `sudo`, system nie pozwoli na ustawienie hasła, które nie spełnia kryteriów złożoności.

Ćwiczenie: konfigurowanie kryteriów złożoności hasła

W tym ćwiczeniu możesz użyć maszyny wirtualnej CentOS, AlmaLinux lub Ubuntu, w zależności od potrzeb. Jedyną różnicą jest to, że ani dla CentOS, ani dla AlmaLinux nie wykonujesz pierwszego kroku:

1. Zainstaluj pakiet *libpam-pwquality,* tylko jeśli używasz w tym ćwiczeniu Ubuntu:

   ```
   sudo apt install libpam-pwquality
   ```

2. Otwórz plik */etc/security/pwquality.conf* w preferowanym edytorze tekstu. Usuń symbol komentarza przed linią `minlen` i zmień wartość na `19`. Powinien on teraz wyglądać następująco:

   ```
   minlen = 19
   ```

3. Zapisz plik i zamknij edytor.
4. Utwórz konto użytkownika Tomek i spróbuj przypisać mu hasła `turkeylips`, `TurkeyLips` i `Turkey93Lips`. Zwróć uwagę na zmiany w każdym komunikacie ostrzegawczym.
5. W pliku *pwquality.conf* oznacz jako komentarz linię `minlen`. Usuń komentarz z linii `minclass` i linii `maxclassrepeat`. Zmień wartość `maxclassrepeat` na `5`. Linie powinny teraz wyglądać następująco:
6. Zapisz plik i zamknij edytor tekstu.

   ```
   minclass = 3
   maxclassrepeat = 5
   ```

7. Spróbuj przypisać różne hasła, które nie spełniają kryteriów złożoności ustawionych dla konta Tomek, i przyjrzyj się rezultatom.

Uwaga

W pliku */etc/login.defs* na komputerze z systemem CentOS 7 zobaczysz linię `PASS_MIN_LEN 5`.

Podobno ma to na celu ustawienie minimalnej długości hasła, ale w rzeczywistości *pwquality* ją zastępuje. Możesz więc ustawić tam dowolną wartość i nie będzie to miało na nic wpływu. (Zauważ, że parametr `PASS_MIN_LEN` nie jest już obsługiwany w dystrybucjach typu RHEL 8/9).

Ustawianie i używanie mechanizmu wygasania haseł i kont

Zapewne nigdy nie chciałbyś, aby nieużywane konta użytkowników były aktywne. Zdarzało się, że administrator konfigurował konta użytkowników do tymczasowego użytku, na przykład na potrzeby konferencji, a następnie po prostu o nich zapominał, mimo że konta nie były już potrzebne.

Innym przykładem może być sytuacja, w której Twoja firma zatrudnia pracowników kontraktowych i ich umowa wygasa w określonym dniu. Takie konta nie powinny być aktywne i dostępne po zakończeniu wykonywania zadań na rzecz firmy pracowników tymczasowych — stanowiłoby to ogromny problem w zakresie bezpieczeństwa. W takich przypadkach potrzebny jest sposób na zadbanie o to, by nie zapomnieć o tymczasowych kontach użytkowników, gdy nie będą już potrzebne. Jeśli Twój pracodawca wyznaje powszechną zasadę, że użytkownicy powinni regularnie zmieniać swoje hasła, będziesz również chciał zadbać, by to zrobili.

Dane wygaśnięcia hasła i dane wygaśnięcia konta to dwie różne rzeczy. Można ustawić je razem lub każde z osobna. Gdy komuś wygaśnie hasło, może je sam zmienić i wszystko będzie w porządku. Jeśli czyjeś konto wygaśnie, tylko osoba z odpowiednimi uprawnieniami administracyjnymi może je odblokować.

Na początek zerknij, kiedy wygasa Twoje konto. Zauważ, że nie będziesz musiał użyć `sudo`, aby dostać się do własnych danych, ale nadal będziesz musiał podać swoją nazwę użytkownika:

```
donnie@packt:~$ chage -l donnie
 [sudo] password for donnie:
 Last password change : Oct 03, 2017
 Password expires : never
 Password inactive : never
 Account expires : never
 Minimum number of days between password change : 0
 Maximum number of days between password change : 99999
 Number of days of warning before password expires : 7
 donnie@packt:~$
```

Możesz tutaj zobaczyć, że nie ustawiono żadnego terminu wygaśnięcia. Wszystko, co tutaj widzisz, jest ustawione zgodnie z domyślnymi wartościami systemu. Oto zestawienie tego, co widzisz (nie uwzględniłem oczywistych elementów):

- **`Password inactive`** (inercja w zmianie hasła). Jeśli ustawiona tu wartość byłaby dodatnia, miałbym tyle dni na zmianę hasła, które wygasło, zanim system zablokowałby moje konto.
- **`Minimum number of days between password change`** (minimalna liczba dni pomiędzy kolejnymi zmianami hasła). Ponieważ ta wartość jest ustawiona na `0`, mogę zmieniać hasło tak często, jak chcę. Jeśli wartość ta byłaby dodatnia, musiałbym odczekać określoną liczbę dni po zmianie hasła, zanim mógłbym zmienić je ponownie.
- **`Maximum number of days between password change`** (maksymalna liczba dni pomiędzy kolejnymi zmianami hasła). Tu jest domyślna wartość `99999`, co oznacza, że moje hasło nigdy nie wygaśnie.
- **`Number of days of warning before password expires`** (na ile dni przed wygaśnięciem hasła pojawi się ostrzeżenie). Domyślna wartość to `7`, ale jest to raczej bez znaczenia, gdy hasło nigdy nie wygasa.

Uwaga

Za pomocą narzędzia `chage` możesz ustawić dane wygaśnięcia hasła i konta dla innych użytkowników lub użyć opcji `-l`, aby wyświetlić dane wygaśnięcia. Każdy nieuprzywilejowany użytkownik może użyć `chage -l` bez `sudo`, aby zobaczyć swoje dane. Aby ustawić dane lub wyświetlić dane innej osoby, potrzebujesz `sudo`. Przyjrzymy się bliżej `chage` nieco później.

Zanim przyjrzymy się, jak zmienić ustawienia wygasania hasła, zobaczmy, gdzie przechowywane są ustawienia domyślne. Najpierw przyjrzymy się plikowi */etc/login.defs*. Oto trzy odpowiadające za to linie:

```
PASS_MAX_DAYS 99999
PASS_MIN_DAYS 0
PASS_WARN_AGE 7
```

Możesz edytować te wartości, aby dopasować je do potrzeb swojej organizacji. Na przykład zmiana `PASS_MAX_DAYS` na wartość `30` spowoduje, że wszystkie nowe hasła użytkowników od tego momentu będą ważne przez 30 dni (ustawienie domyślnych danych wygaśnięcia hasła w *login.defs* działa we wszystkich używanych przez nas dystrybucjach Linuksa).

Konfigurowanie domyślnych danych wygasania haseł przy użyciu pliku useradd (tylko dla rodziny systemów Red Hat)

W pliku */etc/default/useradd* znajdują się pozostałe domyślne ustawienia. W tym przypadku będziemy pracować na maszynie AlmaLinux 9.

Uwaga

Pomimo że w Ubuntu również znajdziesz plik `useradd`, to bez względu na to, co zrobisz, i tak z niego nie skorzystasz, bo Ubuntu go nie odczytuje. Więc ten opis dotyczy tylko systemów typu Red Hat.

```
# useradd defaults file
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

Linia `EXPIRE=` pozwala ustawić domyślną datę wygaśnięcia dla nowych kont użytkowników, domyślnie nie jest nic ustawione. `INACTIVE=-1` oznacza, że konta użytkowników nie będą automatycznie blokowane po wygaśnięciu ich haseł. Jeśli ustawimy tu wartość będącą liczbą dodatnią, wówczas każdy nowy użytkownik będzie miał tyle dni na zmianę hasła, które wygasło, zanim konto zostanie zablokowane. Aby zmienić domyślne ustawienia w pliku `useradd`, możesz ręcznie edytować plik lub użyć `useradd -D` z odpowiednim przełącznikiem opcji dla elementu, który chcesz zmienić. Na przykład, aby ustawić domyślną datę wygaśnięcia na 31 grudnia 2025 r., polecenie wyglądałoby następująco:

```
sudo useradd -D -e 2025-12-31
```

Aby zobaczyć nową konfigurację, możesz otworzyć plik `useradd` lub po prostu wykonać `sudo useradd -D`:

```
[donnie@localhost ~]$ sudo useradd -D
 GROUP=100
 HOME=/home
 INACTIVE=-1
 EXPIRE=2023-12-31
 SHELL=/bin/bash
 SKEL=/etc/skel
 CREATE_MAIL_SPOOL=yes
 [donnie@localhost ~]$
```

To ustawienie powoduje, że wszystkie nowe konta użytkowników, które zostaną utworzone, będą miały tę samą datę wygaśnięcia. Możesz zrobić to samo z ustawieniem `INACTIVE` lub `SHELL`:

```
sudo useradd -D -f 5
sudo useradd -D -s /bin/zsh

[donnie@localhost ~]$ sudo useradd -D
 GROUP=100
 HOME=/home
 INACTIVE=5
 EXPIRE=2019-12-31
 SHELL=/bin/zsh
 SKEL=/etc/skel
 CREATE_MAIL_SPOOL=yes
[donnie@localhost ~]$
```

Teraz wszystkie nowe konta użytkowników, które zostaną utworzone, będą miały ustawioną powłokę Zsh jako powłokę domyślną, a użytkownicy będą musieli w ciągu pięciu dni zmienić hasła, które wygasły; w innym przypadku ich konta zostaną automatycznie zablokowane.

Uwaga

`useradd` nie ma żadnych mechanizmów kontroli bezpieczeństwa pozwalających na upewnienie się, że domyślna powłoka, którą przypisałeś, jest zainstalowana w systemie. W naszym przypadku, mimo że Zsh nie jest zainstalowany, to `useradd` i tak pozwoli Ci tworzyć konta z Zsh jako domyślną powłoką.

Na ile więc przydatna jest ta funkcja konfiguracji w `useradd` w administracji systemem? Prawdopodobnie nie bardzo, chyba że musisz utworzyć jednocześnie całą grupę kont użytkowników z tymi samymi ustawieniami. Ale nawet jeśli zaszłaby taka potrzeba, to doświadczony administrator po prostu zautomatyzuje ten proces za pomocą skryptu powłoki, zamiast bawić się z tym plikiem konfiguracyjnym.

Ustawianie danych związanych z wygasaniem dla poszczególnych kont za pomocą useradd i usermod

Możliwe, że metodę ustawiania domyślnych wartości wygasania hasła w pliku *login.defs* uznasz za przydatną, w przeciwieństwie do wykonywania tego samego zadania za pomocą zmiany ustawień w pliku konfiguracyjnym `useradd`. Jakie są szanse, że będziesz chciał, by konta wszystkich użytkowników miały tę sama datę wygaśnięcia konta? Ustawienie danych wygaśnięcia hasła w *login.defs* jest bardziej przydatne, ponieważ po prostu mówisz, że chcesz, aby nowe hasła wygasały w ciągu określonej liczby dni, a nie aby wszystkie wygasały w określonym dniu.

Najprawdopodobniej najlepiej jest ustawić dane wygaśnięcia konta dla każdego konta osobno, wiedząc, jak długo dane konta mają być aktywne. Możesz to zrobić na trzy sposoby:

- Użyj `useradd` z odpowiednimi przełącznikami opcji, aby ustawić dane wygaśnięcia podczas tworzenia kont. (Jeśli chcesz utworzyć za jednym zamachem całą grupę kont mających mieć tę samą datę wygaśnięcia, możesz zautomatyzować ten proces za pomocą skryptu powłoki).
- W przypadku już istniejących kont użyj `usermod`. (W `usermod` świetne jest to, że używane są te same przełączniki w opcjach co w `useradd`).
- W przypadku już istniejących kont możesz też użyć `chage`, aby zmodyfikować dane wygaśnięcia. (Używany tu jest zupełnie inny zestaw przełączników dla opcji).

Możesz użyć `useradd` i `usermod` do ustawienia, kiedy ma wygasać konto, ale nie kiedy ma wygasać hasło. Dwa przełączniki opcji, które wpływają na dane wygaśnięcia konta, są następujące:

- `-e` — użyj tej opcji, aby ustawić datę wygaśnięcia konta w postaci RRRR-MM-DD.
- `-f` — użyj tej opcji, aby ustawić liczbę dni po wygaśnięciu hasła użytkownika, po których jego konto ma zostać zablokowane.

Załóżmy, że chcesz utworzyć konto dla Charliego, które wygaśnie pod koniec 2025 roku. Na maszynie typu Red Hat możesz wpisać następujące polecenie:

```
sudo useradd -e 2025-12-31 charlie
```

Na maszynie innej niż Red Hat musisz dodać przełączniki opcji, które utworzą katalog domowy i przypiszą prawidłową domyślną powłokę:

```
sudo useradd -m -d /home/charlie -s /bin/bash -e 2025-12-31 charlie
```

Użyj `chage -l`, aby zweryfikować wprowadzone dane:

```
donnie@ubuntu-steemnode:~$ sudo chage -l charlie
 Last password change : Oct 06, 2017
 Password expires : never
 Password inactive : never
 Account expires : Dec 31, 2025
 Minimum number of days between password change : 0
 Maximum number of days between password change : 99999
 Number of days of warning before password expires : 7
 donnie@ubuntu-steemnode:~$
```

A co, jeśli okaże się, że Charlie ma przedłużony kontrakt i musisz zmienić datę wygaśnięcia jego konta na koniec stycznia 2026 roku? Wtedy użyjesz `usermod` (w ten sam sposób na każdej dystrybucji Linuksa):

```
sudo usermod -e 2026-01-31 charlie
```

Ponownie sprawdź, czy wszystko jest poprawne, za pomocą `chage -l`:

```
donnie@ubuntu-steemnode:~$ sudo chage -l charlie
 Last password change : Oct 06, 2017
 Password expires : never
 Password inactive : never
 Account expires : Jan 31, 2026
 Minimum number of days between password change : 0
```

```
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
donnie@ubuntu-steemnode:~$
```

Opcjonalnie możesz ustawić liczbę dni, po których konto z wygasłym hasłem zostanie zablokowane:

```
sudo usermod -f 5 charlie
```

Ale gdybyś zrobił to teraz, nie zobaczyłbyś żadnej różnicy w wyjściu `chage -l`, ponieważ nadal nie ustawiliśmy, kiedy ma wygasać hasło Charliego.

Ustawianie opcji wygasania poszczególnych kont za pomocą chage

`chage` używa się tylko do modyfikowania ustawień już istniejących kont. Możesz go używać do ustawiania, kiedy ma wygasnąć konto lub hasło. Poniżej zamieściłem przełączniki dla odpowiednich opcji:

Opcja	Wyjaśnienie
`-d`	Jeśli użyjesz opcji `-d 0` na czyimś koncie, zmusi to użytkownika do zmiany hasła przy następnym logowaniu.
`-E`	Jest to odpowiednik małej litery `-e` w `useradd` lub `usermod`. Ustawia datę wygaśnięcia konta użytkownika.
`-I`	Jest to odpowiednik `-f` w `useradd` lub `usermod`. Ustawia liczbę dni, po których konto z wygasłym hasłem zostanie zablokowane.
`-m`	Ustawia to minimalną liczbę dni pomiędzy zmianami hasła. Innymi słowy, jeśli Charlie zmieni dzisiaj swoje hasło, opcja `-m 5` zmusi go do odczekania pięciu dni, zanim będzie mógł ponownie zmienić hasło.
`-M`	Ustawia maksymalną liczbę dni przed wygaśnięciem hasła. (Pamiętaj jednak, że jeśli Charlie ostatnio zmienił swoje hasło 89 dni temu, użycie opcji `-m 90` dla jego konta spowoduje, że jego hasło wygaśnie jutro, a nie za 90 dni).
`-W`	W tej opcji podajesz, na ile dni wcześniej pojawi się komunikat ostrzegający o wygaśnięciu hasła.

Możesz ustawić tylko jeden z tych elementów danych lub możesz ustawić je wszystkie naraz. W rzeczywistości, aby nie frustrować Cię kolejnymi przykładami dla każdego pojedynczego elementu, który można ustawić, ustawmy je wszystkie naraz, z wyjątkiem `-d 0`, a następnie zobaczmy, co z tego wyjdzie:

```
donnie@ubuntu-steemnode:~$ sudo chage -E 2026-02-28 -I 4 -m 3 -M 90 -W 4 charlie

donnie@ubuntu-steemnode:~$ sudo chage -l charlie
 Last password change : Oct 06, 2019
 Password expires : Jan 04, 2026
 Password inactive : Jan 08, 2026
 Account expires : Feb 28, 2026
```

```
 Minimum number of days between password change : 3
 Maximum number of days between password change : 90
 Number of days of warning before password expires : 4
donnie@ubuntu-steemnode:~$
```

Wszystkie dane dotyczące wygasania zostały ustawione.

W naszym ostatnim przykładzie załóżmy, że właśnie utworzyłeś nowe konto dla Szymona i chcesz zmusić go do zmiany hasła przy pierwszym logowaniu. Możesz to zrobić na dwa sposoby. Tak czy inaczej, zrobisz to po początkowym ustawieniu hasła, używając jednego z tych dwóch poleceń:

```
sudo chage -d 0 szymon
```

lub

```
sudo passwd -e szymon
donnie@ubuntu-steemnode:~$ sudo chage -l szymon
Last password change : password must be changed
Password expires : password must be changed
Password inactive : password must be changed
Account expires : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
donnie@ubuntu-steemnode:~$
```

Przejdźmy do ćwiczenia.

Ćwiczenie: ustawianie wygasania konta i hasła

W tym ćwiczeniu utworzysz kilka nowych kont użytkowników, ustawisz dane wygaśnięcia i wyświetlisz wyniki. Możesz wykonać to ćwiczenie na dowolnej maszynie wirtualnej. Jedyną różnicą będą polecenia `useradd`:

1. Na maszynie wirtualnej z systemem CentOS lub AlmaLinux utwórz konto użytkownika Szymon z datą wygaśnięcia 30 czerwca 2025 r. i wyświetl wyniki:

   ```
   sudo useradd -e 2025-06-30 szymon
   sudo chage -l szymon
   ```

2. W przypadku Ubuntu uruchom następujące polecenia:

   ```
   sudo useradd -m -d /home/szymon -s /bin/bash -e 2025-06-30 szymon
   sudo chage -l szymon
   ```

3. Użyj `usermod`, aby zmienić datę wygaśnięcia konta Szymona na 31 lipca 2025 roku:

   ```
   sudo usermod -e 2025-07-31 szymon
   sudo chage -l szymon
   ```

4. Przypisz hasło do konta Szymona, a następnie zmuś go do zmiany hasła przy pierwszym logowaniu. Zaloguj się jako Szymon, zmień jego hasło, a następnie zaloguj się na swoje konto:

```
sudo passwd szymon
sudo passwd -e szymon
sudo chage -l szymon
su - szymon
exit
```

5. Użyj `chage`, aby ustawić pięciodniowy okres blokady zmiany hasła, 90-dniowy okres wygaśnięcia hasła, dwudniowy okres nieaktywności i wyświetlenie ostrzeżenia na pięć dni przed wygaśnięciem:

```
sudo chage -m 5 -M 90 -I 2 -W 5 szymon
sudo chage -l szymon
```

6. Zachowaj to konto, ponieważ będziesz go używać w ćwiczeniu w następnym podrozdziale.

Następnie zobaczmy, jak zapobiegać atakom siłowym (ang. *brute-force*).

Zapobieganie atakom siłowym na hasła

Co zaskakujące, jest to kolejny temat, który wywołuje pewne kontrowersje. To znaczy nikt nie zaprzecza mądrości automatycznego blokowania kont użytkowników, które są atakowane. Kontrowersyjna część dotyczy liczby nieudanych prób logowania, na które powinniśmy zezwolić przed zablokowaniem konta.

W epoce kamienia łupanego, co było dawno temu, w czasach gdy miałem jeszcze pełno włosów na głowie, wczesne systemy operacyjne Unix pozwalały użytkownikom na tworzenie haseł składających się z maksymalnie ośmiu małych liter. Tak więc w tamtych czasach wczesny człowiek mógł metodą prób i błędów w końcu trafić, jakie jest czyjeś hasło, po prostu siadając przy klawiaturze i wpisując losowe hasła. To właśnie wtedy zaczęła się filozofia blokowania kont użytkowników po zaledwie trzech nieudanych próbach logowania. W dzisiejszych czasach przy stosowaniu silnych haseł (ang. *password*) lub jeszcze lepiej silnych zdań pełniących rolę haseł (ang. *passphrase*) ustawianie blokady po trzech nieudanych próbach logowania spowoduje trzy rzeczy:

- Będzie niepotrzebnie frustrować użytkowników.
- Spowoduje dodatkową pracę dla pracowników działu pomocy technicznej.
- Jeśli konto naprawdę jest atakowane, zostanie zablokowane, zanim zdążysz zareagować, i nie da Ci szans na zebranie informacji o atakującym.

Ustawienie wartości blokady na coś bardziej realistycznego, na przykład 100 nieudanych prób logowania, nadal zapewni bezpieczeństwo na przyzwoitym poziomie, ale jednocześnie da Ci wystarczająco dużo czasu na zebranie informacji o atakujących. A dodatkowo, co równie ważne, nie spowoduje nadmiernej frustracji użytkowników i obciążenia pracowników działu pomocy technicznej.

W każdym razie niezależnie od tego, ile nieudanych prób logowania jest dozwolonych w Twojej organizacji, i tak musisz wiedzieć, jak to wszystko skonfigurować. W systemach typu RHEL 7 i Ubuntu 18.04 zrobisz to, konfigurując `pam_tally2`. W systemach typu RHEL 8/9 i Ubuntu 20.04/22.04 zamiast tego skonfigurujesz moduł `pam_faillock`. Zobaczmy więc, jak to się robi.

Konfigurowanie modułu pam_tally2 PAM w systemie CentOS 7

Aby ta magia zadziałała, będziemy polegać na naszym dobrym przyjacielu, PAM. Moduł `pam_tally2` jest już zainstalowany w CentOS 7, ale nie jest skonfigurowany. Zaczniemy od edycji pliku */etc/pam.d/login*. Łatwo jest dowiedzieć się, jak go skonfigurować, ponieważ na dole strony podręcznika `pam_tally2` jest przykład:

```
EXAMPLES
 Add the following line to /etc/pam.d/login to lock the account after 4 failed
logins. Root account will be locked as well. The accounts will be
automatically unlocked after 20 minutes. The module does not have to be called
in the account phase because the login calls pam_setcred(3) correctly.
/* Dodaj następującą linię do /etc/pam.d/login, aby zablokować konto po 4 nieudanych
logowaniach. Konto root również zostanie zablokowane. Konta zostaną automatycznie
odblokowane po 20 minutach. Moduł nie musi być wywoływany w kontekście konta, ponieważ
logowanie poprawnie wywołuje pam_setcred(3).*/
 auth required pam_securetty.so
 auth required pam_tally2.so deny=4 even_deny_root
unlock_time=1200
 auth required pam_env.so
 auth required pam_unix.so
 auth required pam_nologin.so
 account required pam_unix.so
 password required pam_unix.so
 session required pam_limits.so
 session required pam_unix.so
 session required pam_lastlog.so nowtmp
 session optional pam_mail.so standard
```

Wskazówka

Jeśli pracujesz z serwerem działającym w trybie tekstowym, będziesz musiał skonfigurować tylko plik */etc/pam.d/login*. Ale jeśli pracujesz z maszyną, na której działa środowisko graficzne, musisz również skonfigurować pliki */etc/pam.d/password.auth* i */etc/pam.d/system.auth*. Jak to zrobić, zobaczysz, gdy przejdziesz do ćwiczeń praktycznych.

W drugim wierszu przykładu widzimy, że `pam_tally2` jest ustawiony następująco:

- `deny=4`. Oznacza to, że atakowane konto użytkownika zostanie zablokowane po zaledwie czterech nieudanych próbach logowania.
- `even_deny_root`. Oznacza to, że nawet konto użytkownika `root` zostanie zablokowane, jeśli zostanie zaatakowane.
- `unlock_time=1200`. Konto zostanie automatycznie odblokowane po 1200 sekundach (20 minutach).

Jeśli teraz spojrzysz na rzeczywisty plik logowania na swojej maszynie wirtualnej, zobaczysz, że nie wygląda on dokładnie tak, jak ten przykładowy plik logowania, który znajduje się na stronie podręcznika. Nic nie szkodzi, zrobimy tak, że będzie działać.

Po skonfigurowaniu pliku logowania i nieudanym logowaniu zobaczysz nowy plik utworzony w folderze */var/log*. Informacje z tego pliku wyświetlisz za pomocą narzędzia `pam_tally2`. Możesz również użyć `pam_tally2` do ręcznego odblokowania zablokowanego konta, jeśli nie chcesz czekać na upływ limitu czasu:

```
donnie@centos7:~$ sudo pam_tally2
 Login Failures Latest failure From
 charlie 5 10/07/17 16:38:19
donnie@centos7:~$ sudo pam_tally2 --user=charlie --reset
 Login Failures Latest failure From
 charlie 5 10/07/17 16:38:19
donnie@centos7:~$ sudo pam_tally2
donnie@centos7:~$
```

Zauważ, że po zresetowaniu konta Charliego nie otrzymałem żadnych wyników po powtórzeniu zapytania.

Ćwiczenie: konfiguracja pam_tally2 w systemie CentOS 7

Konfiguracja `pam_tally2` jest bardzo prosta, ponieważ wymaga tylko dodania jednej linii do plików */etc/pam.d/login*, */etc/pam.d/password.auth* i */etc/pam.d/system.auth*. Aby to sobie ułatwić, możesz po prostu skopiować i wkleić tę linię z przykładu znajdującego się na stronie `man pam_tally2`. Pomimo tego, co napisałem wcześniej o zwiększeniu liczby nieudanych logowań do 100, na razie zostawimy tę liczbę ustawioną na `4`, ponieważ wiem, że nie chciałbyś musieć wykonywać 100 nieudanych logowań, aby sprawdzić działanie ustawienia w praktyce.

1. Na maszynie wirtualnej CentOS 7 otwórz do edycji plik */etc/pam.d/login*. Poszukaj linii, która wywołuje moduł *pam_securetty*, powinno to być około linii 2. Pod tą linią wstaw następującą:

   ```
   auth required pam_tally2.so deny=4 even_deny_root unlock_time=1200
   ```

2. Zapisz plik i zamknij edytor.
3. Umieść tę samą linię na górze plików */etc/pam.d/password.auth* i */etc/pam.d/system.auth*, tuż nad pierwszą wymaganą linią `auth`. (Komentarz na górze tych plików informuje, aby nie edytować ich ręcznie, ponieważ uruchomienie `authconfig` zniszczy zmiany. Niestety, musisz edytować je ręcznie, ponieważ `authconfig` nie skonfiguruje tego za Ciebie).
4. W tym kroku musisz wylogować się ze swojego konta, ponieważ `pam_tally2` nie działa z `su`. Wyloguj się więc i celowo używając niewłaściwego hasła, spróbuj zalogować się na konto `szymon`, które utworzyłeś w poprzednim ćwiczeniu. Kontynuuj, aż zobaczysz komunikat informujący, że konto zostało zablokowane. Zwróć uwagę, że gdy wartość `deny` jest ustawiona na `4`, zablokowanie konta Szymona będzie wymagało pięciu nieudanych prób logowania.
5. Zaloguj się ponownie na swoje konto. Uruchom następujące polecenie i zapisz dane wyjściowe:

   ```
   sudo pam_tally2
   ```

6. W tym kroku będziesz symulować, że jesteś pracownikiem działu pomocy technicznej, a Szymon właśnie zadzwonił z prośbą o odblokowanie jego konta. Najpierw musiałbyś sprawdzić, czy naprawdę rozmawiasz z Szymonem, a następnie wprowadziłbyś następujące dwa polecenia:

   ```
   sudo pam_tally2 --user=szymon --reset
   sudo pam_tally2
   ```

7. Teraz gdy już wiesz, jak to działa, otwórz plik */etc/pam.d/login* do edycji. Zmień parametr `deny=` z `4` na `100` i zapisz plik. (Sprawi to, że Twoja konfiguracja będzie nieco bardziej realistyczna, odpowiadając nowoczesnej filozofii bezpieczeństwa).

Teraz przyjrzyjmy się konfiguracji `pam_faillock` na maszynach AlmaLinux.

Konfigurowanie pam_faillock na AlmaLinux 8/9

Moduł `pam_faillock` jest zainstalowany w każdej dystrybucji Linuksa typu RHEL 8 lub RHEL 9. Ponieważ podstawowe koncepcje `pam_faillock` są prawie identyczne jak w przypadku `pam_tally2`, zrezygnuję ze wstępnych wyjaśnień i przejdziemy od razu do praktycznego działania.

Ćwiczenie: konfiguracja pam_faillock na AlmaLinux 8 lub AlmaLinux 9

Chociaż możesz włączyć i skonfigurować `pam_faillock` ręcznie, edytując pliki konfiguracyjne PAM, dystrybucje RHEL zapewniają łatwiejszą metodę, która nazywa się **`authselect`**. (Zauważ, że robimy to dokładnie tak samo dla maszyny z trybem tekstowym, jak i z interfejsem graficznym).

1. Na maszynie wirtualnej AlmaLinux 8 lub AlmaLinux 9 wyświetl dostępne profile `authselect`:

   ```
   [donnie@localhost ~]$ sudo authselect list
   - minimal    Local users only for minimal installations
   - sssd       Enable SSSD for system authentication (also for local users only)
   - winbind    Enable winbind for system authentication
   [donnie@localhost ~]$
   ```

2. Jak na razie mamy do czynienia tylko z użytkownikami lokalnymi. Użyjemy więc profilu `minimal`. Wyświetl funkcje tego profilu w następujący sposób:

   ```
   [donnie@localhost ~]$ sudo authselect list-features minimal
   . . .
   . . .
   with-faillock
   . . .
   . . .
   [donnie@localhost ~]$
   ```

3. Zauważ, że jest tu wiele funkcji, ale interesuje nas tylko funkcja `with-faillock`.

4. Włącz profil `minimal` poleceniem jak poniżej:

   ```
   sudo authselect select minimal --force
   ```

5. Po włączeniu profilu można włączyć moduł *pam_faillock* w następujący sposób:

   ```
   sudo authselect enable-feature with-faillock
   ```

6. W katalogu */etc/security/* otwórz plik *faillock.conf* w ulubionym edytorze tekstu. Poszukaj następujących czterech linii:

   ```
   # silent
   # deny = 3
   # unlock_time = 600
   # even_deny_root
   ```

7. Usuń symbole komentarza z przodu wszystkich czterech linii i zapisz plik.

8. Utwórz konto użytkownika dla Vicky:

   ```
   sudo useradd vicky
   sudo passwd vicky
   ```

9. Otwórz inny terminal i spróbuj się zalogować trzykrotnie jako Vicky — ale tak, by logowanie się nie udało. Następnie wyświetl wyniki we własnym terminalu, jak poniżej:

   ```
   [donnie@localhost ~]$ sudo faillock
   donnie:
   When                Type  Source
   Valid
   vicky:
   When                Type  Source
   Valid
   2022-10-12 15:54:35 RHOST 192.168.0.16
   V[I]
   2022-10-12 15:54:42 RHOST 192.168.0.16
   V
   2022-10-12 15:54:46 RHOST 192.168.0.16
   V
   [donnie@localhost ~]$
   ```

10. Następnie przed upływem 10 minut blokady spróbuj zalogować się jako Vicky, tym razem przy użyciu poprawnego hasła.

11. Po upływie 10 minut ponownie spróbuj się zalogować jako Vicky, używając poprawnego hasła.

12. Wyloguj się z konta użytkownika Vicky, a następnie spróbuj się ponownie trzykrotnie zalogować, podając niewłaściwe hasło. Tym razem zresetuj konto użytkownika przed upływem czasu w następujący sposób:

    ```
    sudo faillock --reset --user vicky
    ```

I to wszystko, jeśli chodzi o to ćwiczenie.

Na Ubuntu robi się to nieco inaczej. W następnym podrozdziale pokażę Ci jak.

Konfiguracja pam_faillock dla Ubuntu 20.04 i Ubuntu 22.04

Niestety, narzędzie `authselect` nie jest dostępne dla Ubuntu, więc trzeba ręcznie edytować pliki konfiguracyjne PAM. Oto krok po kroku procedura, według której należy postępować.

Ćwiczenie: konfiguracja pam_faillock dla Ubuntu 20.04 i Ubuntu 22.04

1. Otwórz plik */etc/pam.d/common-auth* w swoim ulubionym edytorze tekstu. W górnej części pliku wstaw następujące dwie linie:

```
auth    required    pam_faillock.so preauth silent

auth    required    pam_faillock.so authfail
```

2. Otwórz plik */etc/pam.d/common-account* w edytorze tekstu. Na dole pliku dodaj wiersz:

```
account required    pam_faillock.so
```

3. Skonfiguruj plik */etc/security/faillock.conf* w taki sam sposób, jak pokazałem w kroku 5. poprzedniego ćwiczenia dla AlmaLinux.
4. Przetestuj konfigurację zgodnie z krokami od 6. do 8. poprzedniego ćwiczenia dla AlmaLinux.
5. I to już wszystko. Teraz pokażę Ci, jak ręcznie zablokować konto użytkownika.

Blokowanie kont użytkowników

Właśnie zobaczyłeś, w jaki sposób można skonfigurować, by Linux automatycznie blokował atakowane konta użytkowników. Będziesz jednak też miał do czynienia z sytuacjami, w których będziesz chciał ręcznie zablokować konta użytkowników. Ma to miejsce na przykład:

- Gdy użytkownik wyjeżdża na wakacje, a Ty chcesz mieć pewność, że nikt nie będzie się bawił jego kontem podczas jego nieobecności.
- Gdy wobec użytkownika prowadzone jest dochodzenie w związku z podejrzaną działalnością.
- Gdy użytkownik opuszcza firmę.

Jeśli chodzi o ostatni punkt, możesz zadawać sobie pytanie: „dlaczego nie możemy po prostu usunąć konta osoby, która już tu nie pracuje?”. Z pewnością możesz to zrobić. Zanim jednak to zrobisz, musisz zapoznać się z lokalnymi przepisami, aby upewnić się, że nie wpakujesz się w poważne kłopoty. Na przykład w Stanach Zjednoczonych obowiązuje ustawa Sarbanes-Oxley, która nakłada ograniczenia na to, jakie pliki mogą usuwać ze swoich komputerów spółki notowane na giełdzie. Jeśli usuniesz konto użytkownika wraz z jego katalogiem domowym i pocztą, możesz naruszyć ustawę Sarbanes-Oxley lub inne prawo obowiązujące w Twoim kraju.

Tak czy inaczej, istnieją dwa narzędzia, których możesz użyć do tymczasowego zablokowania konta użytkownika:

- `Usermod`,
- `passwd`.

Uwaga

Pomimo pozornej sprzeczności z tym, co wcześniej napisałem, w pewnym momencie będziesz musiał usunąć nieaktywne konta użytkowników. Powodem jest to, że złośliwi osobnicy mogą próbować używać nieaktywnego konta do załatwiania swoich brudnych interesów, zwłaszcza jeśli to nieaktywne konto miało jakiekolwiek uprawnienia administracyjne. Kiedy jednak zdecydujesz się na usunięcie konta, upewnij się, że jest to zgodnie z lokalnymi przepisami i zasadami obowiązującymi w firmie. Najlepiej jest upewnić się, że Twoja organizacja ma pisemne wytyczne dotyczące usuwania nieaktywnych kont użytkowników w swoich procedurach zarządzania zmianami.

Używanie usermod do blokowania konta użytkownika

Załóżmy, że Katarzyna poszła na urlop macierzyński i nie będzie jej przez kilka tygodni. Możemy zablokować jej konto, wydając następujące polecenie:

```
sudo usermod -L katarzyna
```

Kiedy spojrzysz na wpis dotyczący jej konta w pliku */etc/shadow*, zobaczysz teraz wykrzyknik przed skrótem (ang. *hash*) jej hasła, co pokazałem poniżej:

```
katarzyna:!$6$uA5ecH1A$MZ6q5U.cyY2SRSJezVOOOAudP.
ckXXndBNsXUdMI1vPO8aFmlLXcbGV25K5HSSaCv4RlDilwzlXq/hKvXRkpB/:17446:0:99999:7:::
```

Ten wykrzyknik uniemożliwia systemowi odczytanie skrótu hasła, co skutecznie blokuje dostęp do systemu.

Aby odblokować jej konto, wydaj następujące polecenie:

```
sudo usermod -U katarzyna
```

Możesz sprawdzić, że teraz wykrzyknik został usunięty, dzięki czemu Katarzyna może się już zalogować na swoje konto.

Używanie passwd do blokowania kont użytkowników

Możesz również zablokować konto Katarzyna następującym poleceniem:

```
sudo passwd -l katarzyna
```

Polecenie to robi to samo co `usermod -L`, ale w nieco inny sposób. Po pierwsze `passwd -l` daje Ci informację zwrotną o tym, co się dzieje, podczas gdy `usermod -L` nie daje żadnej informacji zwrotnej. W Ubuntu informacja zwrotna wygląda następująco:

```
donnie@ubuntu-steemnode:~$ sudo passwd -l katarzyna
 passwd: password expiry information changed.
donnie@ubuntu-steemnode:~$
```

W systemach CentOS lub AlmaLinux informacja zwrotna wygląda tak:

```
[donnie@localhost ~]$ sudo passwd -l katarzyna
 Locking password for user katarzyna.
 passwd: Success
[donnie@localhost ~]$
```

Ponadto na komputerze z systemem CentOS lub AlmaLinux zobaczysz, że `passwd -l` umieszcza dwa wykrzykniki przed skrótem hasła (zamiast tylko jednego). Tak czy inaczej, efekt jest taki sam.

Aby odblokować konto Katarzyna, wydaj polecenie:

```
sudo passwd -u katarzyna
```

Uwaga

W wersjach Red Hat lub CentOS wcześniejszych niż 7 `usermod -U` usuwał tylko jeden z wykrzykników, które `passwd -l` umieszczał przed skrótem hasła w pliku `shadow`, co powodowało, że konto było nadal zablokowane. To nic wielkiego, ponieważ ponowne uruchomienie `usermod -U` usuwało drugi wykrzyknik.

Od czasu wprowadzenia dystrybucji typu RHEL 7 zostało to naprawione. `Passwd -l` nadal umieszcza dwa wykrzykniki w pliku `shadow`, ale `usermod -U` teraz usuwa je oba. (Szkoda, bo popsuło mi to bardzo dobre ćwiczenie praktyczne, które lubiłem pokazywać swoim studentom).

Blokowanie konta użytkownika root

Chmura to obecnie wielki biznes i obecnie dość powszechne jest wynajmowanie wirtualnego serwera prywatnego od firm takich jak Rackspace, DigitalOcean lub Microsoft Azure. Mogą one służyć różnym celom:

- Zamiast wynajmować usługę hostingową, możesz uruchomić własną stronę internetową, na której zainstalujesz własne oprogramowanie serwerowe.
- Możesz skonfigurować aplikację internetową, do której dostęp będą miały inne osoby.
- Niedawno widziałem filmik na YouTube, na kanale związanym z kopaniem kryptowalut, na którym pokazano, jak skonfigurować główny węzeł Proof of Stake na wynajętym wirtualnym serwerze prywatnym.

Jedną wspólną cechą większości tych usług w chmurze jest to, że kiedy po raz pierwszy konfigurujesz swoje konto, a dostawca konfiguruje dla Ciebie maszynę wirtualną, jesteś

proszony o zalogowanie się na konto użytkownika *root*. (Dzieje się tak nawet w przypadku Ubuntu, mimo że konto *root* jest domyślnie wyłączone, jeśli zainstalujesz Ubuntu lokalnie).

Wiem, że są ludzie, którzy po prostu logują się na główne konto serwerów uruchomionych w chmurze bez zastanowienia, ale jest to naprawdę fatalny pomysł. Są botnety, takie jak botnet Hail Mary, nieustannie skanujące internet w poszukiwaniu serwerów, których port Secure Shell jest dostępny z sieci. Gdy botnet znajdzie taki serwer, przeprowadza siłowy atak na konto użytkownika *root* tego serwera. W ten sposób botnetom czasami udaje się włamać, zwłaszcza jeśli konto *root* jest zabezpieczone słabym hasłem.

Więc pierwszą rzeczą, którą powinieneś zrobić podczas konfigurowania serwera uruchomionego w środowisku chmury obliczeniowej, jest utworzenie dla siebie konta normalnego użytkownika i skonfigurowanie go z pełnymi uprawnieniami przez `sudo`. Następnie powinieneś wylogować się z konta użytkownika *root*, zalogować się na swoje nowe konto i wydać następujące polecenie:

```
sudo passwd -l root
```

Naprawdę, po co ryzykować, że Twoje konto *root* zostanie przez kogoś przejęte?

Konfigurowanie wiadomości z ostrzeżeniem

Zaręczam, że ostatnia rzecz, jakiej możesz chcieć, to ustawione wyświetlanie po zalogowaniu wiadomości (ang. *security banner*) o treści w stylu „Witamy w naszej sieci". Mówię o tym, ponieważ kilka lat temu uczestniczyłem w prowadzonym przez mentora kursie SANS związanym z obsługą incydentów. Instruktor opowiedział nam następującą historię: firma podała osobę podejrzaną o włamanie się do sieci do sądu, ale sprawa została oddalona. Powód? Domniemany intruz powiedział: „Cóż, zobaczyłem wiadomość z napisem »Witamy w sieci«, więc pomyślałem, że naprawdę jestem tam mile widziany". Tak, to podobno wystarczyło, by sąd sprawę oddalił.

Kilka lat później opowiedziałem tę historię studentom na jednych z moich zajęć z administrowania Linuksem. Jeden ze studentów powiedział: „To jest bez sensu. Wszyscy mamy wycieraczki przy drzwiach wejściowych z napisem »Witamy«, ale to nie znaczy, że włamywacze są mile widziani". Muszę przyznać, że miał rację, i teraz zastanawiam się nad prawdziwością tej historii.

W każdym razie dla spokoju sumienia powinieneś skonfigurować wiadomość powitalną przy logowaniu, jasno informującą, że tylko autoryzowani użytkownicy powinni uzyskiwać dostęp do systemu.

Korzystanie z pliku motd

Plik */etc/motd* będzie prezentował wiadomość z komunikatem każdej osobie, która zaloguje się do systemu przez Secure Shell. Na Twoim komputerze z CentOS lub AlmaLinux pusty plik *motd* już tam jest. Na Twoim komputerze z Ubuntu pliku *motd* nie ma, ale jego utworzenie jest bardzo proste. Tak czy inaczej, otwórz plik w edytorze tekstu i utwórz wiadomość. Zapisz plik i przetestuj go, logując się zdalnie przez Secure Shell. Powinieneś zobaczyć coś takiego:

```
maggie@192.168.0.100's password:
Last login: Sat Oct 7 20:51:09 2017
Warning: Authorized Users Only!
All others will be prosecuted.
[maggie@localhost ~]$
```

Uwaga

motd to skrót od ang. *Message of the Day* (wiadomość dnia).

Ubuntu jest dostarczane z dynamicznym systemem MOTD, który wyświetla predefiniowane informacje od firmy Ubuntu i dynamicznie pobierane informacje o systemie operacyjnym. Kiedy utworzysz nowy plik *motd* w katalogu */etc*, cokolwiek, co w nim umieścisz, pojawi się na końcu dynamicznego wyjścia, tak jak poniżej:

```
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-48-generic x86_64)
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information as of Thu Oct 13 06:20:54 PM UTC 2022

 System load:  0.0               Processes:               103
 Usage of /:   47.8% of 9.75GB   Users logged in:         1
 Memory usage: 12%               IPv4 address for enp0s3: 192.168.0.11
 Swap usage:   0%

39 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Warning!!! Authorized users only!
Last login: Thu Oct 13 17:14:52 2022 from 192.168.0.16
```

Linia `Warning!!! Authorized users only!` (Uwaga! Dostęp tylko dla osób upoważnionych!) jest tym, co umieściłem w pliku */etc/motd*.

Korzystanie z pliku issue

Zawartość pliku *issue*, również znajdującego się w katalogu */etc*, odpowiada za zawartość komunikatu wyświetlanego na lokalnym terminalu, tuż nad monitem logowania. Domyślny plik *issue* zawiera tylko kod makra, który wyświetla informacje o maszynie. Oto przykład z maszyny z Ubuntu:

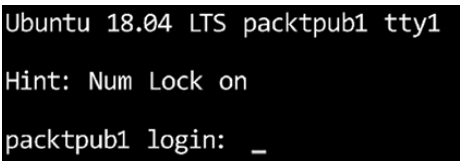
```
Ubuntu 22.04.1 LTS \n \l
```

Na maszynie typu Red Hat wygląda to tak:

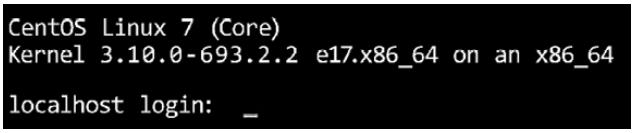
```
\S
Kernel \r on an \m
```

Na komputerze z Ubuntu komunikat wyglądałby mniej więcej tak jak na rysunku 3.1.

Na maszynie typu Red Hat wyglądałoby to mniej więcej tak jak na rysunku 3.2.

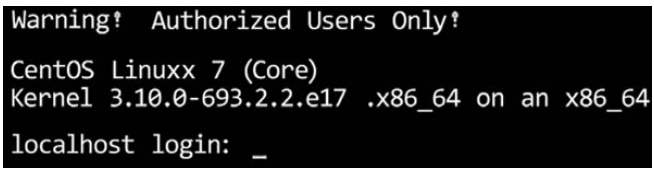


Rysunek 3.1. Komunikat z pliku issue



Rysunek 3.2. Domyślny komunikat z pliku issue w systemie CentOS

Możesz umieścić wiadomość z ostrzeżeniem (ang. *security message*) w pliku *issue* i będzie ono widoczne po ponownym uruchomieniu systemu (rysunek 3.3).



Rysunek 3.3. Zmodyfikowany komunikat z pliku issue w CentOS

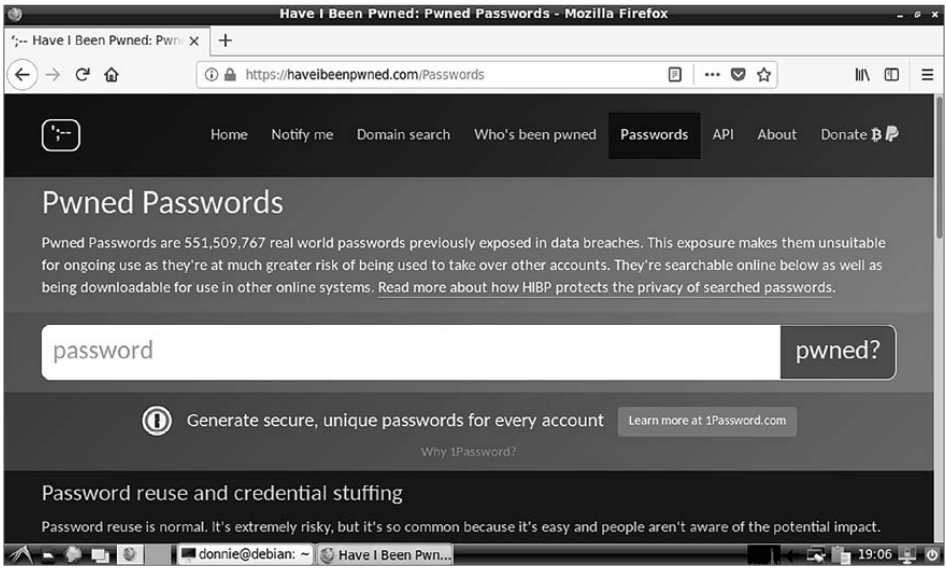
Czy w rzeczywistości umieszczanie wiadomości z ostrzeżeniem w pliku *issue* ma jakikolwiek sens? Jeśli Twoje serwery są odpowiednio zabezpieczone w serwerowni i dostęp do nich jest pod kontrolą, to prawdopodobnie nie. W przypadku komputerów stacjonarnych, które są ogólnie dostępne, jest to bardziej przydatne.

Korzystanie z pliku issue.net

Po prostu z niego nie korzystaj. Służy do logowania przez `telnet`, ale w dzisiejszych czasach włączony `telnet` to bardzo poważna wpadka. Jednak z jakiegoś niezrozumiałego dla mnie powodu plik *issue.net* nadal znajduje się w katalogu */etc*.

Wykrywanie ujawnionych haseł

Tak, moi drodzy, źli ludzie naprawdę dysponują obszernymi słownikami haseł, które albo są powszechnie używane, albo zostały ujawnione (ang. *compromised*). Jednym z najskuteczniejszych sposobów przeprowadzenia siłowego ataku na hasła jest skorzystanie z tych słowników do przeprowadzenia ataku słownikowego. Polega on na tym, że narzędzie do łamania haseł odczytuje hasła z określonego słownika i wypróbowuje każde z nich, dopóki lista nie zostanie wyczerpana lub atak się nie powiedzie. Skąd więc możesz wiedzieć, czy Twoje hasło znajduje się na jednej z tych list? To proste. Po prostu skorzystaj z jednej z usług online, które sprawdzą Twoje hasło. Jedną z popularnych stron jest Have I Been Pwned? (rysunek 3.4).



Rysunek 3.4. Strona Have I Been Pwned

Uwaga

Możesz przejść do Have I Been Pwned?, korzystając z adresu *https://haveibeenpwned.com*.

Wszystko, co musisz zrobić, by sprawdzić, czy Twoje hasło znajduje się na listach ujawnionych haseł, to je wpisać, a usługa pokaże, czy znajduje się ono w bazie. Zastanów się jednak. Czy naprawdę chcesz wysłać swoje używane w praktyce hasło na czyjąś stronę internetową? Tak myślałem, że nie. Zamiast tego po prostu wyślij wartość skrótu hasła. Albo jeszcze lepiej, wyślij tylko taki fragment skrótu (ang. *hash*), aby witryna mogła znaleźć hasło w swojej bazie danych, ale nie tyle, aby mogła dowiedzieć się, jakie jest całe Twoje hasło. Można to zrobić za pomocą **interfejsu API** aplikacji Have I Been Pwned.

Aby pokazać Ci, jak to zrobić, pokażę prosty przykład z użyciem `curl` (wraz z API), aby wyświetlić listę skrótów haseł, w których występuje `21BD1` jako część wartości skrótu. Ćwiczenie możesz wykonać na dowolnej maszynie wirtualnej. Ja zrobię to tylko na stacji roboczej Fedora, której obecnie używam, pisząc ten tekst. Po prostu uruchom następujące polecenie:

```
curl https://api.pwnedpasswords.com/range/21BD1
```

Przesłanych zostanie wiele danych, więc pokażę tu tylko kilka pierwszych wierszy:

```
[donnie@fedora-teaching ~]$ curl https://api.pwnedpasswords.com/range/21BD1
 0018A45C4D1DEF81644B54AB7F969B88D65:1
 00D4F6E8FA6EECAD2A3AA415EEC418D38EC:2
 011053FD0102E94D6AE2F8B83D76FAF94F6:1
 012A7CA357541F0AC487871FEEC1891C49C:2
 0136E006E24E7D152139815FB0FC6A50B15:3
 01A85766CD276B17DE6DA022AA3CADAC3CE:3
```

```
024067E46835A540D6454DF5D1764F6AA63:3
02551CADE5DDB7F0819C22BFBAAC6705182:1
025B243055753383B479EF34B44B562701D:2
02A56D549B5929D7CD58EEFA97BFA3DDDB3:8
02F1C470B30D5DDFF9E914B90D35AB7A38F:3
03052B53A891BDEA802D11691B9748C12DC:6
. . .
. . .
```

Przekierujmy to do `wc -l`, przydatnego narzędzia do zliczania, aby zobaczyć, ile jest pasujących wyników:

```
[donnie@fedora-teaching ~]$ curl https://api.pwnedpasswords.com/range/21BD1 | wc -l
 % Total % Received % Xferd Average Speed Time Time Time Current
 Dload Upload Total Spent Left Speed
 100 20592 0 20592 0 0 197k 0 --:--:-- --:--:-- --:--:-- 199k
 526
[donnie@fedora-teaching ~]$
```

Wygląda na to, że jest 526 dopasowań. Ale to nie jest zbyt użyteczne, więc podrasujmy to trochę. Zrobimy to, tworząc skrypt powłoki *pwnedpasswords.sh*, który wygląda następująco:

```
#!/bin/bash
candidate_password=$1
echo "Candidate password: $candidate_password"
full_hash=$(echo -n $candidate_password | sha1sum | awk '{print substr($1, 0,
32)}')
prefix=$(echo $full_hash | awk '{print substr($1, 0, 5)}')
suffix=$(echo $full_hash | awk '{print substr($1, 6, 26)}')
if curl https://api.pwnedpasswords.com/range/$prefix | grep -i $suffix;
  then echo "Candidate password is compromised";
  else echo "Candidate password is OK for use";
fi
```

Dobra, nie mogę Cię w tej chwili przemienić w guru skryptów powłoki, ale oto uproszczone wyjaśnienie:

- `candidate_password=$1`. Powoduje, że możesz wprowadzić hasło, które chcesz sprawdzić podczas wywoływania skryptu.
- `full_hash=`, `prefix=`, `suffix=`. Te linie odpowiadają za obliczenie wartości skrótu SHA1 hasła, a następnie wyodrębniają tylko te części, które chcesz wysłać do usługi sprawdzania haseł.
- `if curl`. Zapętlamy kod, korzystając z `if..then..else`, co umożliwia wysyłanie wybranych fragmentów skrótu hasła do usługi sprawdzającej, a następnie informuje nas, czy hasło zostało ujawnione.

Po zapisaniu pliku ustaw dla niego uprawnienia do wykonywania dla użytkownika w następujący sposób:

```
chmod u+x pwnedpasswords.sh
```

Sprawdźmy teraz, czy `TurkeyLips`, moje ulubione hasło wszech czasów, zostało ujawnione:

```
[donnie@fedora-teaching ~]$ ./pwnedpasswords.sh TurkeyLips
 Candidate password: TurkeyLips
 % Total % Received % Xferd Average Speed Time Time Time Current
 Dload Upload Total Spent Left Speed
 0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:--
09FDEDF4CA44D6B432645D6C1D3A8D4A16BD:2
 100 21483 0 21483 0 0 107k 0 --:--:-- --:--:-- --:--:-- 107k
 Candidate password is compromised
[donnie@fedora-teaching ~]$
```

O rany, hasło zostało ujawnione! No to w takim razie nie chcę używać go jako hasła w rzeczywiście działającym systemie.

Teraz spróbujmy jeszcze raz, ale z losową dwucyfrową liczbą na końcu:

```
[donnie@fedora-teaching ~]$ ./pwnedpasswords.sh TurkeyLips98
Candidate password: TurkeyLips98
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 20790 0 20790 0 0 110k 0 --:--:-- --:--:-- --:--:-- 110k
Candidate password is OK for use
[donnie@fedora-teaching ~]$
```

Cóż, teraz dowiaduję się, że to hasło jest w porządku. Nadal jednak prawdopodobnie nie chcesz używać tak prostej permutacji hasła, o którym wiadomo, że zostało ujawnione.

Uwaga

Chciałbym przypisać sobie autorstwo skryptu powłoki, który tu przedstawiłem, ale nie mogę. To dzieło mojego kumpla, Leo Dorrendorfa z byłej firmy VDOO zajmującej się bezpieczeństwem internetu rzeczy, która została przejęta przez JFrog. (Powieliłem ten skrypt tutaj za jego uprzejmą zgodą).

Jeśli jesteś zainteresowany rozwiązaniami bezpieczeństwa dla swoich urządzeń internetu rzeczy, możesz je sprawdzić tutaj:

https://jfrog.com/security-and-compliance/?vr=1/

Żeby nie było niedomówień: firma VDOO/JFrog była jednym z moich klientów.

Po tym wszystkim muszę Ci przypomnieć, że zdanie jako hasło jest nadal lepsze niż samo hasło. Zdanie w roli hasła jest nie tylko trudniejsze do złamania, ale także znacznie mniej prawdopodobne jest, że znajdzie się na liście ujawnionych danych uwierzytelniających.

Ćwiczenie: wykrywanie ujawnionych haseł

W tym ćwiczeniu użyjesz interfejsu API z *pwnedpasswords*, aby sprawdzić własne hasła:

1. Użyj `curl`, aby sprawdzić, ile jest haseł z ciągiem `21BD1` w ich skrótach haseł:

   ```
   curl https://api.pwnedpasswords.com/range/21BD1
   ```

2. W katalogu domowym dowolnej maszyny wirtualnej z systemem Linux utwórz skrypt *pwnpassword.sh* o następującej zawartości:

```
#!/bin/bash
candidate_password=$1
echo "Candidate password: $candidate_password"

full_hash=$(echo -n $candidate_password | sha1sum | awk '{print
substr($1, 0, 32)}')
prefix=$(echo $full_hash | awk '{print substr($1, 0, 5)}')
suffix=$(echo $full_hash | awk '{print substr($1, 6, 26)}')

if curl https://api.pwnedpasswords.com/range/$prefix | grep -i $suffix;
        then echo "Candidate password is compromised";
        else echo "Candidate password is OK for use";
fi
```

3. Ustaw uprawnienia wykonywania skryptu:

```
chmod u+x pwnedpasswords.sh
```

4. Uruchom skrypt, podając `TurkeyLips` jako hasło:

```
./pwnedpasswords.sh TurkeyLips
```

5. Powtórz krok 4. tyle razy, ile będziesz chciał, używając za każdym razem innego hasła.

Techniki zarządzania użytkownikami, które pokazałem do tej pory, świetnie sprawdzają się przy niewielkiej liczbie komputerów. Ale co, jeśli pracujesz w dużym przedsiębiorstwie? Przyjrzymy się temu w kolejnym podrozdziale.

Scentralizowane zarządzanie użytkownikami

W środowisku korporacyjnym często będziesz mieć setki lub tysiące użytkowników i komputerów, którymi musisz zarządzać. Siłą rzeczy logowanie się do każdego serwera dzienników lub stacji roboczej każdego użytkownika w celu wykonania procedur, które właśnie poznałeś, byłoby raczej rzeczą niewykonalną (co nie znaczy, że masz tego nie umieć). Potrzebujesz jednak też sposobu na zarządzanie komputerami i użytkownikami z jednej centralnej lokalizacji. Brak miejsca nie pozwala mi na szczegółowe przedstawienie różnych metod osiągnięcia tego celu, więc na razie musisz zadowolić się ogólnym przeglądem.

Microsoft Active Directory

Nie jestem wielkim fanem ani systemu Windows, ani firmy Microsoft. Ale jeśli chodzi o Active Directory, to muszę wyrazić swoje uznanie, bo ono się należy. Jest to całkiem zgrabny produkt, który znacznie upraszcza zarządzanie bardzo dużymi sieciami korporacyjnymi. I tak, możliwe jest dodanie komputerów Unix/Linux i ich użytkowników do domeny Active Directory.

Uwaga

Skrywam pewien mroczny sekret i mam nadzieję, że mnie za to nie znienawidzisz. Zanim zająłem się Linuksem, uzyskałem certyfikat MCSE dla Windows Server 2003. W większości przypadków moi klienci pracują wyłącznie na komputerach z Linuksem, ale od czasu do czasu muszę wykorzystać moje umiejętności MCSE. Kilka lat temu były klient potrzebował mojej pomocy do skonfigurowania opartego na Linuksie serwera Nagios jako części domeny Windows Server 2008, tak aby jego użytkownicy byli uwierzytelniani przez Active Directory. Zajęło mi to trochę czasu, ale w końcu się udało, a klient był zadowolony.

O ile nie występujesz w wielu rolach — tak jak ja czasami muszę to robić — to jako administrator Linuksa prawdopodobnie nie będziesz musiał uczyć się, jak korzystać z Active Directory. Najprawdopodobniej po prostu powiesz administratorom Windows Server, czego potrzebujesz, i oni się tym zajmą.

Wiem, że chcesz zobaczyć, co możemy zrobić, dysponując serwerem z Linuksem. A więc zaczynamy.

Samba w systemie Linux

Samba jest demonem systemu Unix/Linux, który może służyć trzem celom:

- Jego głównym celem jest udostępnianie katalogów z serwera Unix/Linux stacjom roboczym z systemem Windows. Katalogi są wyświetlane w eksploratorze plików systemu Windows tak, jakby były udostępnione z innych maszyn z systemem Windows.
- Można go również skonfigurować jako sieciowy serwer druku.
- Może być również skonfigurowany jako kontroler domeny Windows.

Możesz zainstalować Sambę w wersji 3 na linuksowym serwerze i skonfigurować demona tak, aby działał jako kontroler domeny Windows NT w starym stylu. Jest to dość skomplikowana procedura i zajmuje trochę czasu. Gdy jednak to zrobisz, możesz dołączyć do domeny zarówno maszyny z systemem Linux, jak i Windows oraz używać zwykłych narzędzi do zarządzania użytkownikami i grupami w systemie Windows.

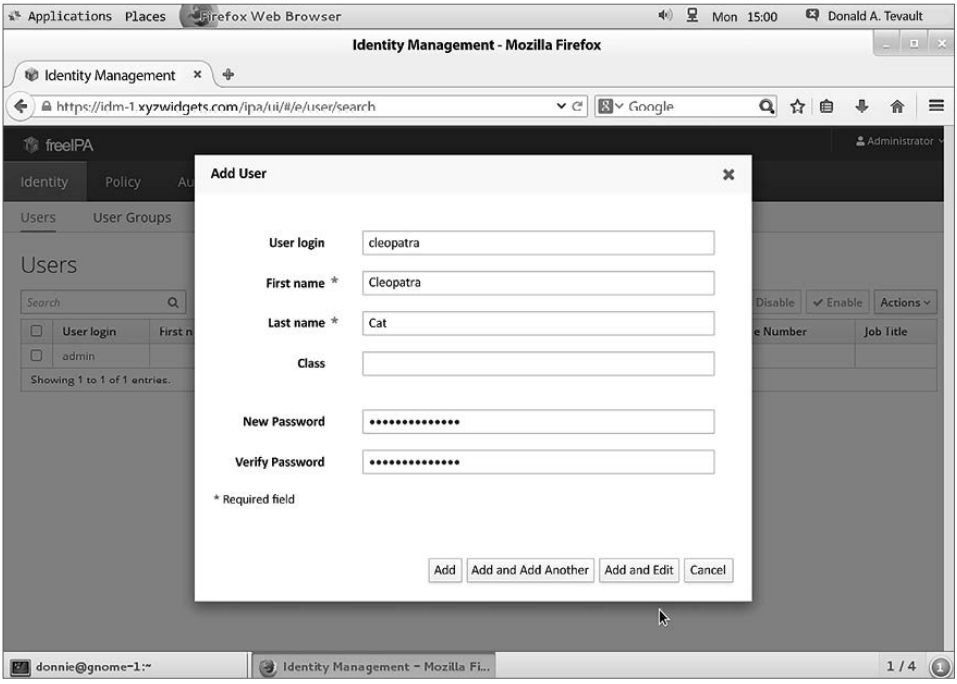
Jednym ze świętych Graali społeczności Linuksa było wymyślenie, jak emulować Active Directory na linuksowym serwerze. Urzeczywistniło się to zaledwie kilka lat temu, wraz z wprowadzeniem Samby w wersji 4. Ale jej konfiguracja jest bardzo złożoną procedurą i prawdopodobnie nie jest czymś, co sprawi Ci frajdę. Być może powinniśmy więc szukać czegoś jeszcze lepszego.

FreeIPA — zarządzanie tożsamością na dystrybucjach typu RHEL

Kilka lat temu firma Red Hat wprowadziła FreeIPA jako zestaw pakietów dla Fedory. Dlaczego Fedory? Ponieważ chciano poddać go dokładnym testom na Fedorze przed udostępnieniem go w rzeczywistych sieciach produkcyjnych. Teraz jest też dostępny dla systemów od RHEL 7 do RHEL 9 i wszystkich opartych na nich dystrybucjach, w tym CentOS i AlmaLinux. Akronim *IPA* oznacza:

- tożsamość (ang. ***I**dentity*),
- zasady (ang. ***P**olicy*),
- kontrolę (ang. ***A**udit*).

Jest to coś w rodzaju odpowiedzi świata linuksowego na Active Directory Microsoftu, ale wciąż nie jest to kompletne rozwiązanie. Robi kilka fajnych rzeczy, ale wciąż jest w trakcie rozwoju. Najfajniejsze jest to, że instalacja i konfiguracja jest prosta. Wszystko, czego naprawdę potrzebujesz, to zainstalować pakiety z normalnych repozytoriów, otworzyć odpowiednie porty zapory sieciowej, a następnie uruchomić skrypt instalacyjny. A potem już możesz rozpocząć dodawanie użytkowników i komputerów do nowej domeny za pośrednictwem interfejsu internetowego FreeIPA. Na rysunku 3.5 możesz zobaczyć, jak dodaję konto dla Cleopatry, mojej szaro-białej kotki w prążki.



Rysunek 3.5. Dodawanie użytkownika przez FreeIPA

Chociaż do domeny FreeIPA możesz dodać maszyny z systemem Windows, nie jest to zalecane. Począwszy jednak od RHEL/CentOS 7.1, możesz używać FreeIPA do tworzenia relacji zaufania między domenami (ang. *cross-domain trusts*) z domeną Active Directory.

Uwaga

Oficjalna nazwa tego programu to FreeIPA. Ale z jakiegoś dziwnego powodu ludzie z Red Hat odmawiają podania tej nazwy w swojej dokumentacji. Zawsze odnoszą się do niego jako do „zarządzania tożsamością" (ang. *Identity Management*) lub w skrócie — *IdM*.

To już wszystko, jeśli chodzi o tematykę zarządzania użytkownikami. Poniżej znajdziesz podsumowanie, a następnie możesz przejść do następnego rozdziału.

Podsumowanie

W tym rozdziale zobaczyłeś, jak zablokować katalogi domowe użytkowników, jak egzekwować zasady silnych haseł oraz jak egzekwować zasady wygasania kont i haseł. Następnie omówiłem sposoby zapobiegania siłowym atakom na hasła, ręcznego blokowania kont użytkowników, konfigurowania ostrzeżeń i sprawdzania, czy hasła nie zostały ujawnione. Wszystko zakończyłem krótkim przeglądem scentralizowanych systemów zarządzania użytkownikami.

W następnym rozdziale pokażę Ci, jak pracować z różnymi narzędziami związanymi z zaporą sieciową. Do zobaczenia.

Pytania

1. W którym pliku skonfigurowałbyś kryteria złożoności dla haseł?
2. Jakie powinno być ustawienie `UMASK` w pliku */etc/login.defs* podczas korzystania z narzędzia `useradd` na maszynie typu RHEL 7?
3. Podczas korzystania z narzędzia `adduser` na komputerze z Ubuntu 20.04 w jaki sposób skonfigurowałbyś plik */etc/adduser.conf*, aby katalogi domowe nowych użytkowników uniemożliwiały innym użytkownikom dostęp do nich?
4. Jaką zmianę wprowadził ostatnio NIST w swoich zalecanych zasadach dotyczących haseł?
5. Których trzech z poniższych narzędzi możesz użyć do ustawienia danych wygaśnięcia konta użytkownika?
   a) `useradd`
   b) `adduser`
   c) `usermod`
   d) `chage`

6. Dlaczego może być bardziej wskazane zablokowanie konta użytkownika byłego pracownika zamiast usunięcia?
   a) Łatwiej jest zablokować konto niż je usunąć.
   b) Usunięcie konta trwa zbyt długo.
   c) Nie da się usunąć konta użytkownika.
   d) Usunięcie konta użytkownika wraz z jego plikami i pocztą może wpędzić Cię w kłopoty z prawem.
7. Właśnie utworzyłeś konto użytkownika dla Szymona i chcesz zmusić go do zmiany hasła przy pierwszym logowaniu. Które dwa z poniższych poleceń to zrobią?
   a) `sudo chage -d 0 szymon`
   b) `sudo passwd -d 0 szymon`
   c) `sudo chage -e szymon`
   d) `sudo passwd -e szymon`
8. Który z poniższych elementów powoduje, że narzędzie `adduser` ma przewagę nad tradycyjnym narzędziem `adduser`?
   a) `adduser` może być używany w skryptach powłoki.
   b) `adduser` jest dostępny dla wszystkich dystrybucji Linuksa.
   c) `adduser` ma opcję, która pozwala zaszyfrować katalog domowy użytkownika podczas tworzenia konta użytkownika.
   d) `adduser` jest również dostępny dla systemów Unix i BSD.
9. Jak nazywa się moduł PAM w najnowszych dystrybucjach Linuksa, którego możesz użyć do wymuszenia silnych haseł?
   a) *cracklib*
   b) *passwords*
   c) *Secure*
   d) *pwquality*

Lektura uzupełniająca

- *You might not need complex, alphanumeric passwords after all* („Być może nie potrzebujesz złożonych, alfanumerycznych haseł”): *https://www.pcmag.com/news/355496/you-might-not-need-complex-alphanumeric-passwords-after-all*.
- *The new NIST Guidelines — We had it all wrong before* („Nowe wytyczne NIST — wcześniej się myliliśmy”): *https://www.riskcontrolstrategies.com/2018/01/08/new-nist-guidelines-wrong/*.
- Zarządzanie użytkownikami systemu Linux: *https://www.youtube.com/playlist?list=PL6IQ3nFZzWfpy2gISpCppFk3UQVGf_x7G*.

- Strona główna projektu FreeIPA: *https://www.freeipa.org/page/Main_Page*.
- Dokumentacja RHEL 9 (przewiń w dół do sekcji *Identity Management* [zarządzanie tożsamością]): *https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9*.
- Dokumentacja RHEL 8 (przewiń w dół do sekcji *Identity Management* [zarządzanie tożsamością]): *https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/*.
- Dokumentacja RHEL 7 (przewiń w dół do sekcji *Identity Management* [zarządzanie tożsamością]): *https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/*.
- *pam_faillock: Lock user account aft er X failed attempts* („Zablokuj konto użytkownika po X nieudanych próbach”): *https://www.golinuxcloud.com/pam-faillock-lock-user-account-linux/*.

Odpowiedzi

1. */etc/security/pwquality.conf*
2. `077`
3. Zmień wartość `DIR_MODE=` na `DIR_MODE=750`
4. Porzucili swoją starą filozofię dotyczącą złożoności haseł i ich wygasania.
5. a, c i d
6. d
7. a i d
8. c
9. d

Twój Linux Twoją twierdzą!

Systemy linuksowe uchodzą za bezpieczne i odporne na cyberataki. Jednak nawet najbezpieczniejszy system może paść ich ofiarą, jeśli jego administrator nie zastosuje odpowiednich środków zabezpieczających. Cyberprzestępcy wciąż opracowują i testują nowe metody ataków, dlatego też specjaliści do spraw bezpieczeństwa nieustannie muszą doskonalić swoje umiejętności.

Oto kompleksowe omówienie sposobów wdrażania najnowszych dostępnych środków zabezpieczających systemy linuksowe. Z książki dowiesz się, jak skonfigurować laboratorium do ćwiczeń praktycznych, tworzyć konta użytkowników z odpowiednimi poziomami uprawnień, chronić dane dzięki uprawnieniom i szyfrowaniu, a także skonfigurować zaporę sieciową przy użyciu najnowszych technologii. Nauczysz się też automatyzować takie czynności jak monitorowanie systemu za pomocą **auditd** i utwardzanie (*hardening*) konfiguracji jądra Linux. Poznasz również sposoby ochrony przed złośliwym oprogramowaniem i skanowania systemów pod kątem luk w zabezpieczeniach. Znajdziesz tu ponadto podpowiedź, jak używać Security Onion do skonfigurowania systemu wykrywania włamań.

W książce między innymi:

- zapobieganie naruszeniom bezpieczeństwa systemów Linux
- dodatkowe funkcje i możliwości systemu Linux
- ochrona przed nieautoryzowanym dostępem
- konfiguracja uprawnień do plików i katalogów
- utwardzanie usługi Secure Shell
- szablony zabezpieczeń i monitorowanie

Donald A. Tevault od 2006 roku pracuje z systemami Linux. Posiada certyfikaty *Level 3 Security* i *GIAC Incident Handler*. Jest profesjonalnym wykładowcą, prowadził zajęcia dla pasjonatów Linuksa z całego świata. Brał także udział w badaniach nad bezpieczeństwem Linuksa na potrzeby rozwoju IoT.

Helion
helion.pl
HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl
KOD KORZYŚCI
Sięgnij po więcej!
ISBN 978-83-289-0292-3
9 788328 902923
Cena: 129,00 zł
packt