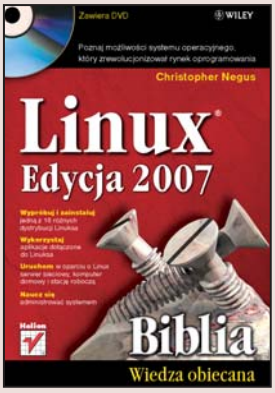
Linux. Biblia. Edycja 2007



Autor: Christopher Negus
Tłumaczenie: Robert Górczyński
ISBN: 978-83-246-1172-0
Tytuł oryginału: Linux Bible 2007 Edition: Boot up Ubuntu, Fedora, KNOPPIX, Debian, SUSE, and 11 Other Distributions (Bible)
Format: B5, stron: 968
oprawa twarda
Zawiera CD-ROM, DVD

Poznaj możliwości systemu operacyjnego, który zrewolucjonizował rynek oprogramowania

- Wypróbuj i zainstaluj jedną z 16 różnych dystrybucji Linuksa
- Wykorzystaj aplikacje dołączone do Linuksa
- Uruchom w oparciu o Linux serwer sieciowy, komputer domowy i stację roboczą
- Naucz się administrować systemem

Linux stanowi fenomen na współczesnym rynku oprogramowania. Prosty system operacyjny wzorowany na Uniksie, napisany w ramach zajęć laboratoryjnych przez studenta Linusa Thorvaldsa i udostępniony przez autora bezpłatnie w sieci, w krótkim czasie zyskał uznanie setek tysięcy użytkowników, stając się ikoną ruchu open-source i zdobywając ugruntowaną pozycję tam, gdzie wcześniej królowały drogie komercyjne systemy operacyjne. W sieci znajdziemy kilkanaście wersji Linuksa, zwanych dystrybucjami, a oferta oprogramowania dla tego systemu powiększa się z każdym dniem, dzięki czemu na korzystanie z niego decyduje się coraz więcej prestiżowych firm i przedsiębiorstw.

„Linux. Biblia. Edycja 2007" to książka, którą musisz przeczytać, gdy postanowisz rozpocząć przygodę z Linuksem. Znajdziesz w niej omówienie dystrybucji tego systemu, dowiesz się, jak go zainstalować, optymalnie skonfigurować, a przede wszystkim, jak korzystać z niego i dołączonego oprogramowania. Nauczysz się pracować z konsolą tekstową i środowiskiem graficznym, administrować systemem i łączyć się z internetem. Ponadto zobaczysz, jak konfigurować serwer WWW, bazy danych, druku i poczty elektronicznej. Przeczytasz także o narzędziach programistycznych dostępnych dla Linuksa.

- Historia Linuksa
- Praca z powłoką tekstową
- Korzystanie z środowiska graficznego KDE i GNOME
- Administrowanie systemem i kontami użytkowników
- Połączenie z siecią lokalną i internetem
- Zabezpieczanie Linuksa przed atakami hakerów
- Wybór i instalacja dystrybucji systemu Linux
- Praca z pakietem OpenOffice.org
- Korzystanie z poczty elektronicznej i WWW
- Konfigurowanie serwera Apache
- Uruchamianie serwera poczty, bazy danych, druku i plików

Programowanie w systemie Linux
Wybierz odpowiednią dla siebie dystrybucję Linuksa!

Wydawnictwo Helion
ul. Kościuszki 1c
44-100 Gliwice
tel. 032 230 98 63
e-mail: helion@helion.pl

Spis treści

Rozdział 6.
Bezpieczeństwo systemu Linux

W tym rozdziale:

- Lista kontrolna bezpieczeństwa Linuksa.
- Używanie zabezpieczenia w postaci hasła.
- Monitorowanie plików dzienników zdarzeń.
- Bezpieczna komunikacja za pomocą narzędzi powłoki.
- Zrozumienie techniki ataków.
- Zabezpieczanie serwerów za pomocą certyfikatów.
- Używanie narzędzi bezpieczeństwa systemu Linux.

Od początku istnienia sieci niektórzy użytkownicy próbowali włamywać się do systemów innych użytkowników. Wraz z rozwojem internetu oraz dostępu szerokopasmowego ten problem przybrał tylko na sile. Niezabezpieczony komputer domowy może zostać wykorzystany jako potężny przekaźnik wiadomości pocztowych, stanowić miejsce wymiany nielegalnych danych, narażać użytkownika na wyciek jego poufnych informacji lub stać się źródłem innych, podobnie nieprzyjemnych zdarzeń.

Dawno temu przeprowadzenie ataku sieciowego wymagało od atakującego pewnego wysiłku oraz umiejętności. W chwili obecnej zautomatyzowane narzędzia mogą być wykorzystywane nawet przez zupełnie początkujących użytkowników do próby złamania systemu połączonego z siecią w zastraszająco krótkim czasie. Oprócz tego robaki sieciowe otrzymały możliwość zmiany dużej liczby niezabezpieczonych systemów w armię „zombie", używanych do olbrzymich, skoordynowanych ataków typu Distributed Denial of Service (DDOS).

Dlaczego należy przejmować się kwestiami bezpieczeństwa? Według organizacji Internet Storm Center (*http://isc.sans.org*) przeciętnie już po upływie16 minut komputer podłączony do internetu staje się obiektem pewnego rodzaju ataku. Zabezpieczenie każdego systemu komputerowego nie jest szalenie trudne, ale wymaga po prostu zdrowego rozsądku oraz stosowania się do dobrych nawyków dotyczących bezpieczeństwa.

W większości przypadków dobre nawyki dotyczące ustalania i ochrony haseł, monitorowania plików dzienników zdarzeń oraz utworzenia dobrych reguł zapory sieciowej pozwalają na skuteczną ochronę przed atakami. Czasami konieczne staje się podjęcie bardziej aktywnych działań w odpowiedzi na włamania.

Wiele zadań powiązanych z zabezpieczaniem systemu Linux jest wspólnych zarówno dla komputerów biurkowych, jak i serwerów. Ponieważ jednak serwery pozwalają klientom z zewnątrz na pewny poziom dostępu do systemu, wymagają specjalnych środków bezpieczeństwa.

W rozdziale zostaną przedstawione ogólne zadania zabezpieczania systemów Linux oraz techniki bezpieczeństwa stosowane w komputerach biurkowych i serwerach. Następnie zostaną omówione niektóre narzędzia możliwe do użycia z poziomu systemów Linux uruchamianych bezpośrednio z nośnika służące do rozwiązywania problemów z komputerem oraz siecią.

Lista kontrolna bezpieczeństwa Linuksa

Podczas gdy większość systemów opartych na Linuksie zawiera wszystkie narzędzia niezbędne do zabezpieczenia komputera, to jeśli użytkownik będzie nierozważny, ktoś może (i prawdopodobnie spróbuje) włamać się do systemu, przejąć go i ukraść dane. Nie wolno zapominać, że żadne środki bezpieczeństwa nie dają 100 procent pewności. Dysponując fizycznym dostępem do komputera lub nieograniczoną ilością wolnego czasu na próby włamania, doświadczony i zdeterminowany cracker będzie mógł włamać się do dowolnego komputera.

Istnieje jednak wiele zabezpieczeń, które zwiększają poziom bezpieczeństwa systemu Linux. Przedstawiona poniżej lista prezentuje szeroki zakres funkcji wpływających na bezpieczeństwo komputera biurowego lub serwera.

- **Kontrola fizycznego dostępu**. Umieszczenie komputera w zamykanym na klucz pomieszczeniu jest całkiem dobrym pomysłem, zwłaszcza jeśli zawiera on bardzo ważne dane. Można ograniczyć krąg osób, które posiadają fizyczny dostęp do komputera, poprzez włączenie haseł w BIOS-ie (uniemożliwiając w ogóle uruchomienie komputera) oraz programie rozruchowym GRUB lub LILO. W BIOS-ie można także ograniczyć liczbę urządzeń uruchamiających komputer.
- **Dodanie użytkowników oraz haseł**. Utworzenie oddzielnych kont użytkowników (każde obowiązkowo z silnym hasłem) jest pierwszą linią obrony w ochronie danych. Użytkownicy są nawzajem przed sobą chronieni, jak również przez osobami z zewnątrz, które chciałyby przejąć kontrolę nad jednym z kont użytkownika. Ustalenie grup kont może rozszerzyć koncepcję własności na wielu użytkowników. Informacje dotyczące ustawiania kont użytkowników zostały przedstawione w rozdziale 4., a dodatkowe zostaną zaprezentowane w podrozdziale „Używanie zabezpieczenia w postaci hasła”, znajdującym się w dalszej części rozdziału.
- **Ustawianie uprawnień do odczytu, zapisu oraz uruchamiania**. Każdy element systemu Linux (włączając w to katalogi, aplikacje oraz urządzenia) może zostać ograniczony prawami do odczytu, zapisu i uruchamiania w stosunku do właściciela i grupy, jak i pozostałych użytkowników. W ten sposób można na przykład zezwolić

użytkownikom na uruchamianie polecenia lub otwieranie pliku, ale bez prawa jego modyfikacji. Informacje dotyczące ustawiania uprawnień do plików i katalogów zostały przedstawione w rozdziale drugim.

- **Ochrona użytkownika root**. W standardowych systemach Linux użytkownik root (a także inne konta administracyjne, takie jak apache) posiada specjalne uprawnienia do używania i modyfikacji systemu Linux. Należy chronić hasło użytkownika root oraz nie używać tego konta bez potrzeby. Pozostawiona otwarta powłoka lub środowisko graficzne użytkownika root mogą stać się celem ataku. Uruchamianie graficznych narzędzi administracyjnych jako zwykły użytkownik (i podawanie hasła użytkownika root, gdy zajdzie taka potrzeba) oraz wykonywanie poleceń administracyjnych za pomocą polecenia `sudo` może zredukować narażanie konta superużytkownika na atak. Informacje dotyczące obsługi konta użytkownika root zostały przedstawione w rozdziale czwartym.
- **Używanie zaufanego oprogramowania**. Mimo że oprogramowanie open source jest dostarczane bez gwarancji, to używanie dystrybucji Linuksa z tradycjami (takich jak na przykład Fedora, Debian lub SUSE) daje większe szanse uniknięcia niebezpiecznego oprogramowania. Składy oprogramowania, z których pobierane są pakiety lub uaktualnienia, także powinny być dokładnie zbadane. Z kolei używanie kluczy publicznych GPG pomoże zagwarantować, że instalowane oprogramowanie pochodzi od zaufanego dostawcy. Przed otworzeniem pobieranych plików w aplikacji Linuksa zawsze należy upewnić się, skąd pochodzą dane pliki. Jeżeli pobierane są pełne obrazy ISO, warto sprawdzić integralność pliku za pomocą sum kontrolnych MD5 oraz SHA1 dostarczanych przez twórców.
- **Pobieranie uaktualnień oprogramowania**. W każdej głównej dystrybucji Linuksa (włączając w to między innymi Debiana, SUSE, Gentoo i Red Hat) znajdują się narzędzia do pobierania i instalacji uaktualnień oprogramowania, ponieważ w pakietach oprogramowania odnajdywane są słabe punkty oraz błędy. Należy się upewnić, że uaktualnienia są pobierane, zwłaszcza jeśli Linux działa jako serwer. Wymienione narzędzia to między innymi `apt`, `yum` i `emerge`.
- **Używanie bezpiecznych aplikacji**. Nawet jeśli oprogramowanie wydaje się działać bez zarzutu, pewne aplikacje oferują większą ochronę przed atakami niż inne. Jeżeli na przykład użytkownik chce się zdalnie zalogować do komputera za pomocą internetu, użycie bezpiecznej powłoki (ssh) jest znacznie bezpieczniejszym rozwiązaniem niż skorzystanie z usługi *rlogin* lub *telnet* (które przekazują hasła w postaci zwykłego tekstu). Ponadto niektóre usługi uważane jako niebezpieczne, gdy zostaną udostępnione w internecie (na przykład Samba lub NFS), mogą być używane znacznie bezpieczniej w internecie za pomocą tuneli VPN (na przykład IPSec lub CIPE).
- **Używanie restrykcyjnie ustawionych zapór sieciowych**. Podstawowym zadaniem zapory sieciowej jest akceptacja przychodzących z sieci żądań usług, których używanie jest dozwolone, oraz odrzucanie pozostałych żądań (bazując głównie na żądanych numerach portów). System biurkowy powinien odrzucać żądania przychodzące do większości portów. Z kolei serwer powinien dopuszczać żądania dla ustalonego i nadzorowanego zestawu portów. Więcej informacji dotyczących ustawiania zapory sieciowej za pomocą `iptables` zostanie przedstawionych w rozdziale 18.

- **Włączenie tylko potrzebnych usług**. Aby oferować usługi w systemie Linux (takie jak strony WWW, serwery plików lub poczty), procesy demonów nadsłuchują określone numery portów. Nie należy włączać usług, które nie są potrzebne.

Program, który działa w tle i obsługuje żądania dostępu do usług (takich jak sendmail), nosi nazwę *demona*. Zazwyczaj demony są uruchamiane automatycznie podczas startu systemu i działają aż do jego zamknięcia. Demony mogą być również uruchamiane ręcznie, gdy zaistnieje taka potrzeba. Do tego celu służy `xinetd`, czyli specjalny demon, który nasłuchuje na wielu portach, a następnie uruchamia żądane procesy.

- **Ograniczenie dostępu do usług**. Poprzez zezwolenie na dostęp tylko do określonego komputera, domeny lub interfejsu sieciowego można ograniczyć dostęp do usług. Na przykład komputer z interfejsami zarówno do internetu, jak i sieci lokalnej może umożliwić dostęp do usług takich jak NFS tylko komputerom sieci LAN i nie oferować tych usług komputerom w internecie. Usługi mogą posiadać ograniczenia dostępu we własnych plikach konfiguracyjnych lub za pomocą osłony TCP/IP (opisanej w dalszej części rozdziału).
- **Sprawdzanie systemu**. Linux posiada dziesiątki narzędzi przeznaczonych do sprawdzania bezpieczeństwa systemu. Po zainstalowaniu systemu Linux użytkownik może sprawdzić dostęp do portów systemu za pomocą narzędzia *nmap* lub obserwować ruch sieciowy za pomocą narzędzia *Ethereal*. Pełny obraz bezpieczeństwa systemu dopełniają narzędzia takie jak *Nessus*. Narzędzia bezpieczeństwa umieszczone na płytach CD i DVD dołączonych do książki zostaną omówione w dalszej części rozdziału.
- **Monitorowanie systemu**. W systemie Linux można zarejestrować prawie każdy rodzaj aktywności. Systemowe pliki dzienników zdarzeń za pomocą demonów `syslogd` oraz `klogd` mogą zostać skonfigurowane tak, aby w maksymalny lub minimalny sposób śledzić aktywność systemu. Z kolei narzędzia takie jak *logwatch* przekazują komunikaty o potencjalnych problemach bezpośrednio na konto e-mail administratora. Funkcje rejestrowania zdarzeń zostaną szczegółowo opisane w dalszej części rozdziału.
- **Używanie SELinux**. SELinux jest wyjątkowo bogatym w możliwości (i skomplikowanym) narzędziem do zarządzania dostępem do niemal każdego komponentu systemu Linux. Narzędzie stanowi rozwiązanie kwestii określanej mianem „po uzyskaniu uprawnień superużytkownika staję się odpowiedzialny także za jego mankamenty" systemów Linux i Unix funkcjonujących w środowiskach o znaczeniu krytycznym. Systemy Red Hat oferują użyteczny, ale ograniczony zbiór reguł SELinux, domyślnie włączony w dystrybucji Fedora. Inne dystrybucje Linuksa również pracują nad implementacją SELinux.

Wyszukiwanie zasobów dotyczących bezpieczeństwa danej dystrybucji

Większość dystrybucji Linuksa posiada zasoby poświęcone zagadnieniom pomocy w zabezpieczaniu systemu oraz zawierające informacje związane z bezpieczeństwem danej wersji Linuksa. Poniżej przedstawiono kilka zasobów internetowych, które skupiają się na bezpieczeństwie różnych dystrybucji systemu Linux.

- **Bezpieczeństwo Red Hat Enterprise Linux i Fedora Core** — witryna dotycząca bezpieczeństwa Red Hat (*http://www.redhat.com/security*) zawiera omówienie kwestii bezpieczeństwa RHEL (które zwykle mają zastosowanie również dla dystrybucji Fedora Core). W witrynie można przejrzeć dostępne uaktualnienia i dowiedzieć się więcej na ich temat. Użytkownik może także uzyskać informacje na temat szkoleń z zakresu bezpieczeństwa oraz skonsultować się z Red Hat Inc. Kwestie bezpieczeństwa dotyczące Fedory są dostępne na stronie Wiki (*http://fedoraproject.org/wiki/Security/Features*).

  Z kolei podręcznik bezpieczeństwa Red Hat Enterprise Linux 4 zawiera szczegółowe omówienie bezpieczeństwa Linuksa dla dystrybucji Red Hat. Wymieniony podręcznik jest dostępny w internecie na stronie

  http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/security-guide/.

- **Bezpieczeństwo Debiana** — strona zawierająca informacje dotyczące bezpieczeństwa Debiana (*http://www.debian.org/security*) stanowi punkt centralny wyszukiwania porad na temat bezpieczeństwa, odpowiedzi na najczęściej zadawane pytania oraz odnośników do dokumentów poświęconych zagadnieniom bezpieczeństwa. Podręcznik omawiający kwestie bezpieczeństwa w dystrybucji Debian znajduje się na stronie

  http://www.debian.org/doc/manuals/securing-debian-howto.

- **Bezpieczeństwo Ubuntu** — dokumenty oraz narzędzia związane z bezpieczeństwem Ubuntu znajdują się na stronie

  https://help.ubuntu.com/community/Security.

- **Bezpieczeństwo Gentoo** — na stronie *http://www.gentoo.org/security* znajdują się narzędzia, ogłoszenia oraz odnośniki do stron poświęconych bezpieczeństwu i dokumentacji projektu związanej z zabezpieczaniem systemów Gentoo. Podręcznik bezpieczeństwa Gentoo znajduje się na stronie

  http://www.gentoo.org/doc/en/security.

- **Bezpieczeństwo Slackware** — w celu zapewnienia bezpieczeństwa dystrybucji Slackware warto zapoznać się z poradami odnośnie bezpieczeństwa Slackware (*http://www.slackware.com/security*). Użytkownik może również zapisać się na listę dyskusyjną dotyczącą bezpieczeństwa w Slackware (*http://www.slackware.com/lists*).

- **Bezpieczeństwo SUSE** — pomoc techniczna w zakresie bezpieczeństwa dystrybucji SUSE jest dostarczana przez firmę Novell. Różne zagadnienia dotyczące bezpieczeństwa SUSE zostały przedstawione na stronie

  http://www.novel.com/linux/security/securitysupport.html.

Wyszukiwanie ogólnych zasobów dotyczących bezpieczeństwa

Istnieje wiele zasobów sieciowych dotyczących bezpieczeństwa komputerowego, które zawierają także informacje użyteczne dla administratorów systemu Linux. Poniżej przedstawiono listę witryn, które warto sprawdzić:

- **CERT** (*http://www.cert.org*) — centrum koordynacyjne CERT zajmuje się bezpieczeństwem komputerów. Na stronie głównej centrum znajdują się informacje dotyczące najnowszych słabych punktów. Witryna posiada również zbiór artykułów o praktykach bezpieczeństwa (*http://www/cert.org/nav/articles_reports.html*). Na witrynie znajdują się także zalecenia odnośnie kroków, które należy podjąć, jeśli komputer stał się celem udanego ataku (*http://www.cert.org/tech_tips/win-UNIX-system_compromise.html*).
- **SecurityFocus** (*http://www.securityfocus.com*) — oprócz nowości oraz ogólnych informacji dotyczących bezpieczeństwa komputerowego, SecurityFocus oferuje również kilka zasobów przeznaczonych dla Linuksa. Użytkownik może zaprenumerować tygodniowy newsletter *Linux Security News*.
- **LinuxSecurity** (*http://www.linuxsecurity.com*) — witryna zawiera wiele artykułów oraz funkcji związanych z bezpieczeństwem systemu Linux. Wskazuje również doradców bezpieczeństwa dla ponad dziesięciu dystrybucji Linuksa.

Bezpieczne korzystanie z Linuksa

Zabezpieczenie i zapewnianie bezpieczeństwa systemom Linux oznacza nie tylko podjęcie dobrych decyzji w trakcie początkowej konfiguracji systemu, ale wiąże się również ze sposobem jego dalszego używania. Niezależnie od tego, czy system Linux jest używany jako komputer biurkowy, czy jako serwer, zawsze bardzo ważne są dobre praktyki związane z hasłami, używaniem bezpiecznych aplikacji oraz monitorowaniem plików dzienników zdarzeń.

Ustawienie bezpiecznej zapory sieciowej (jak to zostanie opisane w rozdziale 18.) jest krytycznym krokiem podczas zabezpieczania systemu Linux. Istnieje jeszcze kilka innych środków, które należy zastosować w Linuksie. W podrozdziale zostaną przedstawione niektóre dobre praktyki dotyczące używania haseł, śledzenia aktywności systemu poprzez obserwację plików dzienników zdarzeń oraz komunikacji z innymi systemami za pomocą bezpiecznej powłoki (ssh).

Używanie zabezpieczenia w postaci hasła

Zabezpieczanie hasłem jest podstawowym sposobem zapewnienia bezpieczeństwa każdego nowoczesnego systemu operacyjnego i jednocześnie najczęściej atakowaną funkcją bezpieczeństwa. Naturalne jest, że użytkownik stara się wybrać hasło najłatwiejsze do zapamiętania, ale często oznacza to, że tak wybrane hasło jest łatwe do odgadnięcia. Crackerzy doskonale wiedzą, że w dowolnym systemie posiadającym więcej niż tylko kilku użytkowników przynajmniej jeden z nich zastosuje łatwe do odgadnięcia hasło.

Poprzez użycie metody „brutalnej siły" do próby zalogowania się na każdym koncie systemu i wypróbowanie na każdym z nich najczęściej stosowanych haseł wytrwały cracker ma dużą szansę osiągnięcia celu. Należy pamiętać, że cracker zautomatyzuje taki atak, więc niewykluczone są tysiące prób zalogowania. Oczywiste staje się, że wybór dobrego hasła jest pierwszym i najważniejszym krokiem podczas zabezpieczania systemu.

Poniżej przedstawiono listę elementów, których należy unikać podczas ustalania hasła:

- Nie należy używać żadnych odmian loginu lub pełnego imienia i nazwiska. Nawet jeśli zostanie zróżnicowana wielkość liter, dołączony znak interpunkcyjny lub liczba bądź zapis wspak, to takie hasło wciąż pozostaje łatwe do odgadnięcia.
- Nie należy używać słowa słownikowego, nawet jeśli zostanie uzupełnione o cyfry lub znaki interpunkcyjne.
- Nie należy używać jakichkolwiek poprawnych nazw.
- Nie należy używać kolejnych liter lub cyfr na klawiaturze (na przykład „qwerty" lub „asdfg").

Wybór dobrego hasła

Dobrym sposobem wyboru silnego hasła jest użycie pierwszej litery każdego słowa z łatwego do zapamiętania zdania. Takie hasło staje się jeszcze lepsze po dodaniu cyfr, znaków interpunkcyjnych i zróżnicowaniu wielkości liter. Wybrane zdanie powinno posiadać znaczenie tylko dla użytkownika i nie powinno być publicznie dostępne (dlatego też wybór zdania ze swojej strony internetowej jest wyjątkowo złym pomysłem). W tabeli 6.1 zostały przedstawione przykłady silnych haseł oraz podpowiedzi ułatwiających zapamiętanie tych haseł.

Tabela 6.1. *Propozycje dobrych haseł*

Hasło	Jak je łatwo zapamiętać?
`Mzsmj7l!`	Mój zardzewiały samochód ma już 7 lat!
`2stZp1jl`	2 słonie to ZŁY pomysł, 1 jest lepszy
`CtMp?Omgzp`	Czy to MÓJ płaszcz? Oddaj mi go z powrotem

Powyższe hasła przypominają bełkot, ale w rzeczywistości są stosunkowo łatwe do zapamiętania. Jak widać, nacisk został położony na słowa, które w haśle są przedstawiane za pomocą wielkich liter. Ustawienie hasła odbywa się za pomocą polecenia `passwd`. Po wydaniu polecenia `passwd` z poziomu powłoki użytkownik będzie mógł zmienić hasło. W pierwszej kolejności nastąpi wyświetlenie pytania o dotychczasowe hasło. Aby zmobilizować użytkownika do zapamiętania hasła oraz uniemożliwić jego poznanie innym użytkownikom, którzy mogliby ewentualnie „spoglądać mu przez ramię", wpisywane hasło nie będzie wyświetlane na ekranie.

Zakładając, że dotychczasowe hasło zostanie podane prawidłowo, następnym krokiem polecenia `passwd` będzie pytanie o nowe hasło. W trakcie wpisywania nowego hasła polecenie `passwd` używa biblioteki `cracklib` w celu określenia, czy podawane hasło jest *dobre*, czy *złe*.

Użytkownik, który nie jest użytkownikiem root, zostanie poproszony o podanie innego hasła, jeśli wprowadzone nie zostanie uznane za dobre.

Użytkownik root jest jedynym użytkownikiem, który może użyć *złego* hasła. Po zaakceptowaniu hasła przez bibliotekę `cracklib` polecenie `passwd` prosi o ponowne podanie hasła, aby upewnić się, że nie popełniono pomyłki (którą trudno wychwycić, gdy podawane hasło nie jest widoczne na ekranie). Kiedy użytkownik działa jako root, może zmienić hasło dowolnego użytkownika, podając nazwę użytkownika jako parametr polecenia `passwd`, na przykład:

```
# passwd janek
Changing password for user janek.
New UNIX password: ********
Retype new UNIX password: ********
passwd: all authentication tokens updated successfully.
```

W powyższym przykładzie wykonania polecenia `passwd` dla użytkownika `janek` trzeba dwukrotnie podać nowe hasło. W tym przypadku nie jest wyświetlane pytanie o podanie dotychczasowego hasła. W ten sposób root może wyzerować hasło użytkownika, gdy użytkownik je zapomni (a zdarza się to zbyt często).

Korzystanie z pliku haseł shadow

We wczesnych wersjach systemu Unix wszystkie informacje dotyczące kont użytkowników oraz ich haseł były przechowywane w pliku możliwym do odczytania przez wszystkich użytkowników (chociaż tylko root miał uprawnienia modyfikacji tego pliku). Takie rozwiązanie nie było problemem, ponieważ hasła były zaszyfrowane. Użytym algorytmem szyfrowania był *algorytm trapdoor*, oznaczający, że niezakodowane hasło było kodowane do postaci ciągu znaków, ale ten ciąg znaków nie mógł być przekształcony z powrotem na niezakodowane hasło. Innymi słowy, algorytm szyfrowania *trapdoor* był algorytmem jednokierunkowym.

W jaki więc sposób system sprawdzał poprawność tak zaszyfrowanych haseł? W trakcie logowania system szyfrował hasło podawane przez użytkownika, a następnie porównywał otrzymany zaszyfrowany ciąg tekstowy z ciągiem tekstowym przechowywanym w pliku. Użytkownik uzyskiwał dostęp tylko wtedy, gdy oba ciągi były identyczne. Jeżeli użytkownik zapytałby administratora systemu, jak brzmi hasło do jego konta, w odpowiedzi mógłby usłyszeć „Nie wiem”. Wyjaśnienie jest proste — administrator rzeczywiście nie zna hasła i ma dostęp tylko do jego zaszyfrowanej postaci. Niezaszyfrowane hasło istnieje tylko w chwili jego wpisywania przez użytkownika.

Łamanie zaszyfrowanych haseł

Możliwość zobaczenia zaszyfrowanych haseł przez użytkowników była jednak problemem. Chociaż odszyfrowanie hasła zaszyfrowanego algorytmem *trapdoor* może być trudne (o ile nawet niemożliwe), to bardzo łatwo jest zaszyfrować olbrzymią liczbę zgadywanych haseł i porównać je z zaszyfrowanymi hasłami umieszczonymi w pliku. Ze względu na ogrom danych jest to znacznie efektywniejsze niż rzeczywiste próby logowania z użyciem loginu i hasła. Jeżeli cracker będzie w stanie uzyskać kopię pliku z hasłami, ma znacznie większe szanse na włamanie do systemu.

Na szczęście Linux oraz wszystkie nowoczesne systemy Unix domyślnie obsługują plik haseł *shadow*. Jest on specjalną wersją pliku *passwd*, którą może odczytać tylko root. Plik zawiera zaszyfrowane informacje o hasłach, więc mogą one zostać wyrzucone ze zwykłego pliku *passwd*, do którego dostęp mają wszyscy użytkownicy systemu. Linux obsługuje zarówno starszy pojedynczy plik z hasłami, jak i nowszy plik haseł *shadow*. Zawsze należy stosować plik haseł *shadow* (używany domyślnie).

Sprawdzanie pliku haseł shadow

Plik haseł nosi nazwę *passwd* i jest umieszczony w katalogu */etc*. Z kolei plik haseł *shadow* również znajduje się w katalogu */etc*. Jeżeli w systemie nie ma pliku */etc/shadow*, system Linux prawdopodobnie przechowuje hasła w pliku */etc/passwd*. Można to zweryfikować za pomocą polecenia `less`:

```
# less /etc/passwd
```

Po wydaniu powyższego polecenia na ekranie powinny zostać wyświetlone dane podobne do przedstawionych poniżej:

```
root:DkkS6Uke799fQ:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:/bin/sh
.
.
.
maria:KpRUp2ozmY5TA:500:100:Maria Nowak:/home/maria:/bin/bash
janek:OsXrzvKnQaksI:501:100:Janek Kowalski:/home/janek:/bin/bash
jadwiga:ptNoiueYEjwX.:502:100:Jadwiga Malinowska:/home/jadwiga:/bin/bash
bartek:Ju2vY7AOX6Kzw:503:100:Bartek Matusiak:/home/bartek:/bin/bash
```

Każdy wiersz odpowiada pojedynczemu kontu użytkownika systemu Linux i jest utworzony z siedmiu pól rozdzielonych średnikiem (`:`). Patrząc od lewej do prawej strony, wymienione pola oznaczają nazwę logowania, zaszyfrowane hasło, identyfikator użytkownika, identyfikator grupy, opis, katalog domowy oraz domyślną powłokę. W wierszu pierwszym widać, że wpis dotyczy konta root, które posiada zaszyfrowane hasło `DkkS6Uke799fQ`. Wiersz informuje również, że identyfikator użytkownika i grupy to zero, katalog domowy superużytkownika to */root*, natomiast powłoką domyślną tego konta jest */bin/bash*.

Wszystkie wymienione wartości są standardowe dla konta root, ale zobaczenie zaszyfrowanego hasła powinno spowodować wywołanie sygnału alarmowego u użytkownika systemu, ponieważ stanowi potwierdzenie, że system nie używa pliku haseł o nazwie *shadow*. W takiej sytuacji należy natychmiast skonwertować plik *passwd*, tak aby do przechowywania haseł był używany plik */etc/shadow*. Konwersję wykonuje się za pomocą polecenia `pwconv`. W tym celu należy po prostu zalogować się jako użytkownik root (lub użyć polecenia `su`, aby uzyskać jego uprawnienia), a następnie wydać polecenie `ppwconv`. Nie powoduje ono wyświetlenia na ekranie żadnych danych wyjściowych, ale po jego wykonaniu system będzie posiadał plik */etc/shadow*, natomiast plik */etc/passwd* powinien przedstawiać się następująco:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:/bin/sh
```

```
.
.
.
maria:x:500:100:Maria Nowak:/home/maria:/bin/bash
janek:x:501:100:Janek Kowalski:/home/janek:/bin/bash
jadwiga:x.:502:100:Jadwiga Malinowska:/home/jadwiga:/bin/bash
bartek:x:503:100:Bartek Matusiak:/home/bartek:/bin/bash
```

Zaszyfrowane dane zostały zastąpione znakiem `x`, a hasła przeniesione do pliku */etc/shadow*.

Istnieje również narzędzie *Konfiguracja uwierzytelniania* (dostępne w dystrybucjach Fedora oraz RHEL), którego można użyć do zarządzania hasłami oraz innymi uwierzytelniającymi danymi systemu. Wymienione narzędzie posiada także funkcje, które pozwalają na pracę z hasłami MD5, uwierzytelnianiem LDAP lub Kerberos 5. Narzędzie jest dostępne w menu *System/Administracja/Uwierzytelnianie*.

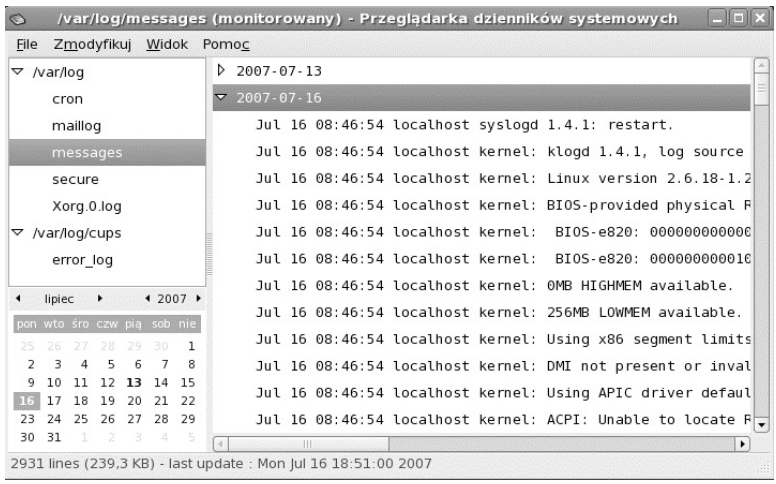
Podczas pracy z hasłami grup można użyć narzędzia `grpconv`, które skonwertuje hasła z pliku */etc/groups* na hasła w pliku haseł */etc/gshadow*. Jeżeli plik *passwd* lub *groups* zostanie zmieniony lub uszkodzony (uniemożliwiając zalogowanie się na konto), można użyć narzędzi odpowiednio `pwunconv` oraz `grpunconv` do odwrócenia procesu konwersji haseł.

W chwili obecnej w systemie używany jest plik haseł *shadow*, a użytkownicy ustawiają silne hasła. Można więc stwierdzić, że wykonano pierwsze kroki we właściwym kierunku podczas zabezpieczania systemu. Czytelnik prawdopodobnie zauważył, że bezpieczeństwo nie jest tylko jednorazowym zadaniem. Bezpieczeństwo to ciągły proces dotyczący zarówno podejmowanych działań, jak i używanych programów. Warto czytać dalej i dowiedzieć się więcej na ten temat.

Korzystanie z plików dzienników zdarzeń

Jeżeli zostaną zastosowane dobre praktyki dotyczące zapory sieciowej (będą opisane w rozdziale 18.), system będzie dobrze przygotowany na osłabianie większości ataków crackerów bądź zapobieganie im. Aby zapora sieciowa mogła zatrzymać intruza, musi być wstanie rozpoznać atak, gdy taki ma miejsce. Zrozumienie różnych (i licznych) plików dzienników zdarzeń, w których system Linux zapisuje ważne zdarzenia, jest krytycznym elementem takiego zadania. Pliki dzienników zdarzeń systemu Linux są umieszczone w katalogu */var/log*.

Większość systemów Linux używa narzędzi przeglądania plików dzienników zdarzeń dostarczonych wraz ze środowiskiem graficznym (na przykład GNOME), albo w postaci poleceń wykonywanych z poziomu okna terminalu. Systemy używające GNOME bardzo często posiadają narzędzie *Przeglądarka dzienników systemowych* (polecenie `gnome-system-log`), które można wykorzystać do przeglądania i przeszukiwania krytycznych plików dzienników zdarzeń systemu z poziomu GUI. W celu otworzenia okna narzędzia *Przeglądarka dzienników systemowych* z górnego panelu Fedory należy wybrać *System/Administracja/Dziennik systemowy*. Na rysunku 6.1 pokazano przykładowe okno narzędzia *Przeglądarka dzienników systemowych*.



Rysunek 6.1. *Wyświetlanie plików dzienników zdarzeń za pomocą narzędzia Przeglądarka dzienników systemowych*

Aby wyświetlić określony plik dziennika zdarzeń, należy kliknąć w lewej kolumnie jego nazwę. Wyświetlenie komunikatów z określonego dnia i godziny jest możliwe dzięki kalendarzowi umieszczonemu w lewym dolnym rogu okna.

W tabeli 6.2 wymieniono pliki dzienników zdarzeń wyświetlane w oknie narzędzia *Przeglądarka dzienników systemowych* oraz inne interesujące pliki znajdujące się w katalogu */var/log* (w celu otworzenia pliku dziennika zdarzeń nieznajdującego się w lewej kolumnie trzeba wybrać opcję *Otwórz* z menu *Plik*). Wiele wymienionych w tabeli plików jest dostarczanych z większością systemów Linux i są one dostępne tylko dla użytkownika root. Ponadto niektóre systemy Linux mogą używać innej nazwy pliku bądź katalogu (na przykład zamiast */etc/httpd* w niektórych systemach jest katalog */etc/apache*).

Tabela 6.2. *Pliki dzienników zdarzeń umieszczone w katalogu /var/log*

Nazwa systemowego dziennika zdarzeń	Nazwa pliku	Opis
Boot Log	*boot.log*	Zawiera komunikaty wskazujące uruchomione i zamknięte usługi systemowe oraz (ewentualnie) te, których uruchomienie lub zatrzymanie zakończyło się niepowodzeniem. Najnowsze komunikaty są umieszczone na końcu pliku.
Cron log	*cron*	Zawiera komunikaty demona `crond`, który okresowo uruchamia wykonywanie zadań, takich jak tworzenie kopii zapasowych bądź rotacja plików dzienników zdarzeń.
Kernel Startup Log	*dmesg*	Zapis komunikatów wyświetlanych przez jądro w trakcie uruchamiania systemu.
FTP Log	*xferlog*	Zawiera informacje o plikach transferowanych za pomocą usługi FTP.
Apache Access Log	*httpd/access_log*	Plik zawiera żądania względem serwera Apache.
Apache Error Log	*httpd/error_log*	Plik zawiera błędy, które wystąpiły w klientach próbujących uzyskać dane z serwera Apache.

Tabela 6.2. *Pliki dzienników zdarzeń umieszczone w katalogu /var/log — ciąg dalszy*

Nazwa systemowego dziennika zdarzeń	Nazwa pliku	Opis
Mail Log	*maillog*	Zawiera informacje o adresach, z których i do których zostały wysłane wiadomości e-mail. Plik jest użyteczny w trakcie wykrywania spamu.
MySQL Server Log	*mysqld.log*	Plik zawiera informacje powiązane z aktywnością serwera bazy danych MySQL (`mysqld`).
News Log	*spooler*	Katalog zawierający dzienniki komunikatów z serwera list dyskusyjnych, o ile taki jest używany.
RPM Packages	*rpmpkgs*	Plik zawiera listę pakietów RPM zainstalowanych w systemie. (Dla systemów, które nie bazują na pakietach RPM, należy szukać katalogu *debian-installer* lub *packages* zawierającego listę zainstalowanych pakietów).
Security Log	*secure*	Zapis daty, godziny oraz czasu trwania prób zalogowania i sesji.
System Log	*messages*	Plik dziennika zdarzeń ogólnego przeznaczenia, w którym są zapisywane komunikaty wielu programów.
Update Agent Log	*up2date*	Plik zawiera komunikaty wynikające z działań Red Hat Update Agent.
X.Org X11 Log	*Xorg.0.log*	Komunikaty wygenerowane przez serwer X systemu X.Org.
[a]	*gdm/:0.log*	Zawiera komunikaty powiązane z ekranem logowania (*GNOME Display Manager*).
[a]	*samba/log.smbd*	Komunikaty demona serwera Samba (`smbd`).
[a]	*squid/access.log*	Plik zawiera komunikaty związane z serwerem proxy i buforowania.
[a]	*vsftpd.log*	Plik zawiera komunikaty związane z trybem transferu używanym przez demona `vsFTPd` (serwer FTP).
[a]	*sendmail*	Komunikaty błędów zapisane przez demona `sendmail`.
[a]	*uucp*	Komunikaty stanu pochodzące z demona protokołu *Unix to Unix Copy Protocol*.

Litera [a] oznacza plik dziennika zdarzeń, który nie jest pokazywany w oknie narzędzia *Przeglądarka dzienników systemowych*. Dostęp do tego pliku jest możliwy bezpośrednio z katalogu */var/log*.

Ponieważ wymienione pliki dzienników zdarzeń mają postać zwykłych plików tekstowych, można je wyświetlić również za pomocą dowolnego edytora tekstowego (takiego jak vi czy gedit) lub polecenia stronicującego (na przykład `less`).

Rola demona syslogd

Większość plików w katalogu */var/log* jest obsługiwana przez usługę `syslogd`. Demon `syslogd` oznacza System Logging Daemon. Jego zadaniem jest przyjmowanie komunikatów z różnych programów i zapisywanie ich w odpowiednich plikach dzienników zdarzeń. Jest to znacznie lepsze rozwiązanie niż zapis plików dzienników zdarzeń bezpośrednio przez każdy program, gdyż demon umożliwia centralne zarządzanie obsługą plików dzienników zdarzeń. Demon `syslogd` może zostać skonfigurowany tak, aby zapisywał pliki dzienników zdarzeń o różnym stopniu szczegółowości. Demon może również ignorować wszystkie komunikaty poza krytycznymi bądź też zapisywać bardzo skrupulatnie wszystkie informacje.

Demon `syslogd` można nawet przyjmować komunikaty z innych komputerów w sieci. Jest to szczególnie użyteczna funkcja, gdyż umożliwia centralizację zarządzania i pobierania plików dzienników zdarzeń z wielu komputerów w sieci. Takie rozwiązanie niesie ze sobą pewne korzyści związane z bezpieczeństwem.

Jeżeli dowolny komputer z sieci zostanie złamany, wówczas cracker nie będzie mógł usunąć bądź zmodyfikować plików dzienników zdarzeń, ponieważ będą one przechowywane na innym komputerze. Należy jednak pamiętać, że domyślnie pliki dzienników zdarzeń nie są szyfrowane. Podsłuchując ruch w sieci lokalnej, można wykryć przekazywanie tych komunikatów w trakcie ich transferu między komputerami. Ponadto, mimo że cracker nie będzie mógł zmodyfikować starych wpisów, może wpłynąć w taki sposób na system, aby nowym komunikatom nie można było ufać.

Stosunkowo często stosowaną praktyką jest ustalanie komputera przeznaczonego specjalnie do zapisu plików dzienników zdarzeń z innych komputerów w sieci. Ponieważ taki system działa bez innych uruchomionych usług, włamanie do niego jest bardzo mało prawdopodobne. Dzięki temu cracker praktycznie nie ma możliwości wymazania śladów swojej obecności i działalności, ale to nie oznacza, że wszystkie wpisy w dziennikach zdarzeń powstałe po włamaniu będą prawidłowe.

Przekierowanie komunikatów zdarzeń do serwera zdarzeń za pomocą syslogd

W celu przekierowania plików dzienników zdarzeń do demona `syslogd` innego komputera należy przeprowadzić kilka modyfikacji w pliku konfiguracyjnym lokalnego demona `syslogd` (plik */etc/syslog.conf*). Po uzyskaniu uprawnień użytkownika root (za pomocą polecenia `su`) trzeba otworzyć w dowolnym edytorze tekstowym, takim jak vi, plik */etc/syslog.conf*. Plik będzie podobny do przedstawionego poniżej:

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;news.none;authpriv.none;cron.none  /var/log/messages
```

```
# The authpriv file has restricted access.
authpriv.*                                  /var/log/secure

# Log all the mail messages in one place.
mail.*                                      /var/log/maillog

# Log cron stuff
cron.*                                      /var/log/cron

# Everybody gets emergency messages
*.emerg                                     *

# Save news errors of level crit and higher in a special file.
uucp,news.crit                              /var/log/spooler

# Save boot messages also to boot.log
local7.*                                    /var/log/boot.log

#
# INN
#
news.=crit                                  /var/log/news/news.crit
news.=err                                   /var/log/news/news.err
news.notice                                 /var/log/news/news.notice
```

Wiersze rozpoczynające się od znaku # są komentarzami. Pozostałe wiersze zawierają dwie kolumny informacji. Lewa kolumna stanowi rozdzieloną przecinkami (spacje tutaj nie działają) listę rodzajów i priorytetów komunikatów. Z kolei prawa kolumna zawiera plik dziennika zdarzeń, w którym te komunikaty powinny być zapisywane.

W celu wysłania komunikatów do innego komputera (o nazwie `loghost`) zamiast do pliku należy nazwę pliku zastąpić znakiem `@` i nazwą komputera. Na przykład przekierowanie danych wyjściowych wysyłanych zwykle do plików dzienników zdarzeń `messages`, `secure` i `maillog` wymaga w powyższym pliku wprowadzenia następujących zmian:

```
# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;news.none;authpriv.none;cron.none  @loghost

# The authpriv file has restricted access.
authpriv.*                                  @loghost

# Log all the mail messages in one place.
mail.*                                      @loghost
```

Po wprowadzeniu powyższych zmian komunikaty będą wysyłane do demona `syslogd` działającego na komputerze o nazwie `loghost`. Nazwa `loghost` nie jest dowolną nazwą. Zwykle tworzy się nazwę komputera i udostępnia jako alias rzeczywistego systemu działającego jako `loghost`. W ten sposób, jeżeli kiedykolwiek zajdzie potrzeba przeniesienia serwera plików dzienników zdarzeń na inny komputer, wystarczy zmienić tylko alias. Nie będzie konieczna ponowna edycja pliku *syslog.conf* na każdym komputerze.

Po stronie serwera `loghost` komputer musi posiadać uruchomionego demona `syslogd` wraz z opcją `-r`, która powoduje oczekiwanie na komunikaty zdarzeń przekazywane z innych komputerów w sieci. W systemie Fedora Core oznacza to dodanie opcji `-r` do zmiennej

SYSLOGD_OPTIONS w pliku */etc/sysconfig/syslog* i ponowne uruchomienie usługi syslog (service syslog restart). Ponadto serwer loghost musi posiadać port UDP 514 dostępny dla demona syslogd (trzeba sprawdzić plik */etc/services*), a więc należy utworzyć odpowiednią regułę w zaporze sieciowej.

Zrozumienie komunikatów pliku dziennika zdarzeń

Ponieważ w pliku dziennika zdarzeń messages informacje zapisuje wiele programów i usług, bardzo ważne jest zrozumienie formatu tego pliku. Dzięki analizie tego pliku można wcześnie uzyskać ostrzeżenie o problemach w systemie. Każdy wiersz pliku stanowi pojedynczy komunikat zapisany przez program lub usługę. Poniżej przedstawiono fragment rzeczywistego pliku dziennika zdarzeń messages:

```
Feb 25 11:04:32 toys network: Bringing up loopback interface:  succeeded
Feb 25 11:04:35 toys network: Bringing up interface eth0:  succeeded
Feb 25 13:01:14 toys vsftpd(pam_unix)[10565]: authentication failure;
logname= uid=0 euid=0 tty= ruser= rhost=10.0.0.5  user= krzysztof
Feb 25 14:44:24 toys su(pam_unix)[11439]: session opened for
user root by krzysztof(uid=500)
```

Odczyt pliku jest bardzo prosty, o ile wiadomo, na co zwrócić uwagę. Każdy komunikat jest podzielony na pięć głównych części. Patrząc od lewej do prawej strony, są to:

- data i godzina zarejestrowania komunikatu,
- nazwa komputera, z którego pochodzi dany komunikat,
- nazwa programu lub usługi, której dotyczy dany komunikat,
- numer procesu (umieszczony w nawiasach kwadratowych) programu wysyłającego komunikat,
- rzeczywista treść komunikatu.

Spójrzmy ponownie na powyższy fragment pliku. Pierwsze dwa wiersze informują o ponownym uruchomieniu sieci. Kolejny wiersz wskazuje, że użytkownik krzysztof próbował uzyskać dostęp do serwera FTP (próba zakończona niepowodzeniem) z komputera o adresie 10.0.0.5 (użytkownik podał nieprawidłowe hasło i uwierzytelnienie nie powiodło się). Ostatni wiersz wskazuje, że użytkownik krzysztof użył polecenia su, aby uzyskać uprawnienia superużytkownika.

Okazjonalne przeglądanie plików messages i secure umożliwia wychwycenie próby włamania, nim zakończy się ona powodzeniem. Jeżeli plik wskazuje na nadmierną liczbę prób połączenia z określoną usługą, zwłaszcza jeśli żądania nadchodzą z internetu, może oznaczać to próbę ataku.

Używanie narzędzi bezpiecznej powłoki

Narzędzia bezpiecznej powłoki (ssh) to zbiór aplikacji typu klient – serwer, które pozwalają na podstawową komunikację (zdalne logowanie, zdalne kopiowanie, zdalne wykonywanie itd.) między zdalnymi komputerami a danym systemem Linux. Ponieważ

komunikacja między serwerem (zazwyczaj procesem demona sshd) i klientami (takimi jak *ssh*, *scp* i *sftp*) jest szyfrowana, to te narzędzia są niewątpliwie bezpieczniejsze niż podobne, ale starsze narzędzia Uniksa, takie jak *rsh*, *rcp* i *rlogin*.

Większość systemów Linux zawiera klienty bezpiecznej powłoki, a niektóre posiadają również serwer sshd. Jeśli na przykład używana jest dystrybucja Fedora lub Red Hat Enterprise Linux, oprogramowanie ssh składa się z następujących pakietów: *openssh*, *openssh-clients* i *openssh-serwer*.

Uruchamianie usługi ssh

Systemy Linux, które są dostarczane z zainstalowaną usługą ssh, bardzo często są skonfigurowane tak, aby uruchamiać ją automatycznie. W dystrybucjach Fedora i RHEL demon sshd jest uruchamiany za pomocą skryptu startowego */etc/init.d/sshd*. Aby upewnić się, że usługa została skonfigurowana do automatycznego uruchamiania w Fedorze, RHEL lub innym systemie Linux bazującym na pakietach RPM, należy wydać następujące polecenie (jako użytkownika root):

```
# chkconfig --list sshd
sshd      0:off   1:off   2:on   3:on   4:on   5:on   6:off
```

Powyższe polecenie wskazuje, że usługa ssh została skonfigurowana do uruchomienia na 2., 3., 4. lub 5. poziomie systemu (zwykły stan uruchamiania systemu to poziom 5.) i jest wyłączona na pozostałych. Jeżeli usługa ssh jest wyłączona, jej uruchomienie na dowolnym poziomie działania następuje po wydaniu polecenia (jako użytkownik root):

```
# chkconfig sshd on
```

Powyższe polecenie włącza usługą ssh, kiedy system działa na 2., 3., 4. lub 5. poziomie. W celu natychmiastowego uruchomienia usługi należy wydać polecenie:

```
# service sshd start
```

Inne dystrybucje Linuksa mogą po prostu uruchamiać demona sshd z pliku o nazwie podobnej do *rc.sshd*, znajdującego się w katalogu */etc/rc.d*. W każdym przypadku określenie, czy demon sshd jest uruchomiony, następuje po wydaniu polecenia:

```
$ ps ax | grep sshd
1996 ?        Ss       0:00     /usr/sbin/sshd
```

Po wydaniu powyższego polecenia wyraźnie widać, że demon sshd jest uruchomiony. Jeżeli demon sshd działa, a zapora sieciowa pozwala na korzystanie z usługi bezpiecznej powłoki (na otwartym porcie TCP o numerze 22), użytkownik powinien móc używać poleceń klienta ssh do uzyskiwania dostępu do systemu. (Wszelka dalsza konfiguracja ograniczająca możliwości demona sshd jest przeprowadzana w pliku */etc/ssh/sshd_config*).

Używanie poleceń ssh, sftp i scp

Polecenia, których można używać z usługą ssh, to ssh, sftp oraz scp. Zdalni użytkownicy mogą wykorzystać polecenie ssh w celu bezpiecznego zalogowania się do systemu lub zdalnego wykonania polecenia systemu. Z kolei polecenie scp pozwala zdalnym użytkownikom

na kopiowanie plików z oraz do systemu. Natomiast polecenie `sftp` zapewnia bezpieczny dostęp do witryn FTP za pomocą usługi ssh (w przypadku witryn, które oferują dostęp ssh do zawartości FTP).

Podobnie jak w przypadku zwykłej usługi zdalnej powłoki, także bezpieczna powłoka sprawdza plik */etc/hosts.equiv* oraz w katalogu domowym użytkownika plik *.rhost* w celu określenia, czy zezwolić na połączenie. Sprawdzane są również pliki przeznaczone specjalnie dla usługi ssh, czyli */etc/shosts.equiv* i *.shosts*. Używanie plików *shosts.equiv* oraz *.shosts* jest zalecanym rozwiązaniem, ponieważ unika się w ten sposób nadania dostępu od usług nieszyfrowanej zdalnej powłoki. Pliki */etc/shost.equiv* i *.shosts* posiadają takie same funkcje jak pliki *hosts.equiv* i *.rhosts*, tak więc stosują się do nich takie same reguły.

Teraz można już przetestować usługę ssh. Z poziomu innego komputera z zainstalowanym ssh (lub nawet z tego samego komputera, jeśli inny nie jest dostępny) należy wydać polecenie `ssh`, po którym znajdzie się spacja oraz nazwa systemu docelowego. Na przykład w celu nawiązania połączenia z systemem `ratbert.glaci.com` należy wydać polecenie:

```
# ssh ratbert.glaci.com
```

Jeżeli będzie to pierwsze zalogowanie do danego systemu za pomocą polecenia `ssh`, system wyświetli komunikat potwierdzający chęć nawiązania połączenia. Po wyświetleniu poniższego komunikatu należy wpisać `yes` i nacisnąć klawisz *Enter*:

```
The authenticity of host 'ratbert.glaci.com (199.170.177.18)' can't be
established.
RSA key fingerprint is xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx.
Are you sure you want to continue connecting (yes/no)?
```

Następnie system powinien zapytać o nazwę użytkownika oraz hasło. Nawiązane połączenie będzie funkcjonowało tak jak zwykłe zdalne połączenie (innymi słowy, użytkownik może rozpocząć wydawanie poleceń powłoki). Jedyną różnicą jest fakt, że podczas przesyłania informacji przez sieć są one szyfrowane. Użytkownik powinien mieć również możliwość używania polecenia `ssh` do uruchamiania zdalnych poleceń powłoki na zdalnym systemie.

Polecenie `scp` jest podobne do polecenia `rcp` służącego do kopiowania plików między systemami Linux. Poniżej przedstawiono przykład użycia polecenia `scp` do skopiowania pliku o nazwie *notatka* z katalogu domowego użytkownika `janek` do katalogu */tmp* komputera *klon*:

```
$ scp /home/janek/notatka klon:/tmp
janek@klon's password: ********
notatka       100%|****************|  153   0:00
```

Jeżeli będzie wymagane podanie hasła, należy podać hasło danego użytkownika. Po zaakceptowaniu podanego hasła zdalny system poinformuje o pomyślnym skopiowaniu pliku.

Polecenie `sftp` rozpoczyna interaktywną sesję FTP z serwerem FTP obsługującym połączenia ssh. Wiele osób przykładających dużą wagę do bezpieczeństwa wybiera klientów `sftp` zamiast `ftp`, ponieważ zapewniają one znacznie większe bezpieczeństwo połączenia między użytkownikiem i zdalnym komputerem. Oto przykład takiego połączenia:

```
$ sftp ftp.handsonhistory.com
Connecting to ftp.handsonhistory.com
janek@ftp.handsonhistory.com's password: ********
sftp>
```

Od tej chwili można rozpocząć interaktywną sesję FTP. Użytkownik może korzystać z poleceń `get` i `put` na plikach, podobnie jak w przypadku dowolnego klienta FTP, ale z komfortem wypływającym z wiedzy, że praca odbywa się poprzez bezpieczne połączenie.

Polecenie `sftp`, podobnie jak polecenia `ssh` i `scp`, wymaga, aby usługa ssh była uruchomiona na serwerze. Jeżeli nie można nawiązać połączenia z serwerem FTP za pomocą polecenia `sftp`, usługa ssh może nie być dostępna.

Używanie poleceń ssh, scp i sftp bez haseł

W przypadku komputerów wykorzystywanych przez użytkownika (zwłaszcza tych, które znajdują się za zaporą sieciową w sieci lokalnej) dużym ułatwieniem jest możliwość skonfigurowania tych maszyn w taki sposób, aby do logowania nie było potrzebne hasło. Przedstawiona poniżej procedura pokaże, w jaki sposób to zrobić.

Kolejne kroki procedury prowadzą użytkownika do ustawienia możliwości uwierzytelniania z poziomu jednego komputera na drugim bez użycia hasła. W omówionym przykładzie użytkownik lokalny to `chester` znajdujący się na komputerze o nazwie `host1`. Zdalny użytkownik to również `chester`, ale na komputerze o nazwie `host2`.

1. Zaloguj się w komputerze lokalnym (w omawianym przykładzie użytkownik to `chester` na komputerze `host1`).

Polecenie przedstawione w drugim kroku należy wydać na komputerze lokalnym tylko raz. Polecenia generującego klucz nie wolno wydawać ponownie, chyba że klucz zostanie utracony. W trakcie konfiguracji kolejnych zdalnych serwerów trzeba od razu przejść do kroku trzeciego.

2. W celu wygenerowania klucza ssh wydaj następujące polecenia:

```
$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key
(/home/chester/.ssh/id_dsa): <Enter>
Enter passphrase (empty for no passphrase): <Enter>
Enter same passphrase again: <Enter>
Your identification has been saved in
/home/chester/.ssh/id_dsa.
Your public key has been saved in
/home/chester/.ssh/id_dsa.pub.
The key fingerprint is:
3b:c0:2f:63:a5:65:70:b7:4b:f0:2a:c4:18:24:47:69 chester@host1
```

Jak przedstawiono powyżej, trzeba nacisnąć klawisz *Enter*, aby zaakceptować nazwę pliku, w którym będzie przechowywany klucz. Następnie dwukrotne naciśnięcie klawisza *Enter* spowoduje akceptację pustego hasła. (Jeżeli hasło zostanie podane, użytkownik będzie o nie pytany, a więc nastąpi wykluczenie możliwości logowania bez użycia hasła).

3. Użytkownik musi zabezpieczyć prawa dostępu do kluczy uwierzytelniających poprzez ograniczenie uprawnień do katalogu domowego, katalogu *.ssh* oraz plików uwierzytelniających, jak przedstawiono poniżej:

```
$ chmod go-w $HOME
$ chmod 700 $HOME/.ssh
$ chmod go-rwx $HOME/.ssh/*
```

4. Wydaj poniższe polecenie, aby skopiować klucz do zdalnego serwera (nazwę `chester` należy zastąpić nazwą zdalnego użytkownika, a `host2` nazwą zdalnego komputera):

```
$ cd ~/.ssh
$ scp id_dsa.pub chester@host2:/tmp
chester@host2's password: *******
```

5. Wydaj poniższe polecenie, aby dodać klucz ssh do kluczy uwierzytelniających zdalnego użytkownika (kod powinien znajdować się jednym wierszu):

```
$ ssh chester@host2 'cat /tmp/id_dsa.pub >>
/home/chester/.ssh/authorized_keys2'
```

W poprzednich dwóch krokach pojawiło się pytanie o hasło. To zupełnie normalne.

Aby demon `sshd` zaakceptował utworzony przez użytkownika plik *authorized_keys2*, katalog domowy oraz ten plik muszą posiadać bezpieczne uprawnienia. W celu zabezpieczenia tego pliku oraz katalogu należy wydać polecenia:

```
$ ssh chester@host2 chmod go-w $HOME $HOME/.ssh
$ ssh chester@host2 chmod 600 $HOME/.ssh/authorized_keys2
```

6. Wydaj następujące polecenie, które usuwa klucz z katalogu tymczasowego:

```
$ ssh chester@host2 rm /tmp/id_dsa.pub
```

W 6. kroku nie powinno paść pytanie o hasło.

Bardzo ważne jest, aby pamiętać, że po przeprowadzeniu omówionej powyżej procedury wszystko będzie funkcjonowało niezależnie od zmian adresu IP komputera lokalnego. Adres IP nie ma nic wspólnego z tym rodzajem uwierzytelniania.

Zabezpieczanie serwerów Linux

Udostępnienie w publicznej sieci systemu Linux w charakterze serwera stanowi nowe wyzwanie związane z bezpieczeństwem. Zamiast odrzucania wszystkich żądań z zewnątrz od komputera oczekuje się odpowiedzi na żądania dla udostępnianych usług (na przykład WWW, FTP lub poczty elektronicznej) poprzez dostarczenie informacji lub uruchomienie skryptów na danych.

Na temat zabezpieczania serwerów napisano całe książki. Wiele firm, których funkcjonowanie zależy od usług internetowych, zatrudnia w pełnym wymiarze godzin administratorów, którzy czuwają nad bezpieczeństwem serwerów. Z tego powodu niniejszy

podrozdział należy traktować jako ogólny opis niektórych rodzajów ataków oraz niektórych narzędzi służących do zabezpieczania serwerów Linuksa.

Nadzór dostępu do usług za pomocą osłon TCP

Całkowite wyłączenie nieużywanej usługi jest dobrym rozwiązaniem, ale które usługi będą faktycznie potrzebne? W jaki sposób można selektywnie nadawać i odbierać prawa dostępu do tych usług? W przypadku systemów Linux, które zawierają obsługę osłony TCP, to pliki */etc/hosts.allow* oraz */etc/hosts.deny* określają, kiedy dane połączenie zostaje dopuszczone do danej usługi lub odrzucone, na przykład *rlogin*, *rsh*, *telnet*, *finger* i *talk*.

Większość systemów Linux implementujących osłony TCP wprowadza je dla zestawu usług, które są monitorowane przez pojedynczy proces, nazywany Internet Super Server. W dystrybucjach Fedora i RHEL wymieniony serwer to demon `xinetd`, podczas gdy w innych systemach (takich jak Debian) używany jest demon `inetd`. Kiedy następuje żądanie usługi korzystającej z osłon TCP, sprawdzane są pliki *hosts.allow* oraz *hosts.deny* w celu wyszukania wpisu odpowiadającemu adresowi IP komputera nawiązującego połączenie. Sprawdzenie następuje podczas próby połączenia:

- Jeśli dany adres znajduje się w pliku *hosts.allow*, połączenie zostaje zaakceptowane, a plik *hosts.deny* nie jest sprawdzany.
- Jeżeli adres znajduje się w pliku *hosts.deny*, połączenie jest odrzucane.
- Jeżeli adres nie znajduje się w żadnym pliku, połączenie zostaje zaakceptowane.

Należy pamiętać, że kolejność sprawdzania jest ważna. Nie można na przykład zabronić dostępu komputerowi w pliku *hosts.deny*, jeżeli dany komputer uzyskuje dostęp za pomocą pliku *hosts.allow*.

Umieszczanie każdego adresu, który może próbować nawiązać połączenie z serwerem, nie jest niezbędne (a nawet jest niemożliwe). Pliki *hosts.allow* i *hosts.deny* umożliwiają nadzór nad pewnym zestawem adresów. Do określenia wszystkich możliwych adresów stosuje się słowo kluczowe `ALL`. Użytkownik może także ograniczyć wpisy w tych plikach, aby miały zastosowanie tylko do określonych usług sieciowych. Przykład typowych plików *hosts.allow* oraz *hosts.deny* znajduje się poniżej. Oto przykładowy plik */etc/hosts.allow*:

```
#
# hosts.allow  This file describes the names of the hosts are
#          allowed to use the local INET services, as decided
#          by the '/usr/sbin/tcpd' server.
#

cups-lpd: 199.170.177.
in.telnetd: 199.170.177., .linuxtoys.net
vsftpd: ALL
```

Oto przykładowy plik */etc/hosts.deny*:

```
#
# hosts.deny This file describes names of the hosts which are
#          *not* allowed to use the local INET services, as
```

```
#          decided by the '/usr/sbin/tcpd' server.
#

ALL: ALL
```

Powyższy przykład przedstawia raczej restrykcyjną konfigurację. Dopuszczone są połączenia z określonych węzłów do usług *cups-lpd* oraz *telnet*, natomiast wszystkie pozostałe zostają odrzucone. Konfiguracja pozwala również wszystkim węzłom na dostęp do FTP (*vsftp*). Przeanalizujemy więc szczegółowo przedstawione pliki.

Jak zwykle wiersze rozpoczynające się znakiem # są komentarzem, a więc w trakcie przetwarzania pliku zostają zignorowane przez `xinetd` lub `inetd`. Każdy wiersz niebędący komentarzem składa się z rozdzielonej przecinkami listy demonów, znaku dwukropka (`:`), a następnie rozdzielonej przecinkami listy adresów klientów do sprawdzenia. W tym kontekście klientem jest każdy komputer, który próbuje uzyskać dostęp do usługi sieciowej serwera.

Wpis klienta może być liczbowym adresem IP (na przykład `199.170.177.25`) lub nazwą komputera (na przykład `jukebox.linuxtoys.net`), ale najczęściej jest kombinacją wraz ze znakiem wieloznacznym, określającą pewien zakres adresów. Wpis klienta może przyjąć cztery różne formy, które podręcznik pliku *hosts.allow* opisuje w następujący sposób:

- Ciąg znakowy rozpoczynający się od znaku kropki (`.`). Nazwa węzła zostanie dopasowana, jeżeli ostatnie komponenty jej nazwy odpowiadają określonemu wzorcowi. Na przykład wzorzec `.tue.nl` zostanie dopasowany do nazwy węzła `wzv.win.tue.nl`.
- Ciąg znakowy kończący się znakiem kropki (`.`). Adres węzła zostanie dopasowany, jeżeli jego pierwsze pola liczbowe będą odpowiadały danemu ciągu znaków. Na przykład wzorzec `131.155.` będzie dopasowany do adresu (prawie) każdego węzła sieci uniwersytetu w Eindhoven (`131.155.x.x`).
- Ciąg znakowy rozpoczynający się symbolem at (`@`) jest traktowany jako nazwa grupy sieciowej (ang. *netgroup*) NIS. Nazwa węzła zostanie dopasowana, jeżeli jest elementem wskazanej grupy sieciowej. Dopasowania grup sieciowych nie są obsługiwane dla nazw procesów demonów lub nazw użytkowników klientów.
- Wyrażenie w formie *`n.n.n.n/m.m.m.m`* jest interpretowane jako para *`net/maska`*. Adres węzła zostanie dopasowany, jeżeli *`net`* jest równe bitowemu *`and`* adresu oraz masce. Na przykład wzorzec `131.155.72.0/255.255.254.0` spowoduje dopasowanie każdego adresu w zakresie `131.155.72.0` do `131.155.73.255`.

Przykładowy plik *hosts.allow* zawiera dwa rodzaje specyfikacji klienta. Wpis `199.170.177.` będzie próbował dopasować dowolny adres IP, który rozpoczyna się podanym ciągiem znakowym, na przykład `199.170.177.25`. Z kolei wpis `.linuxtoys.net` spróbuje dopasować nazwy węzłów, takie jak `jukebox.linuxtoys.net` lub `picframe.linuxtoys.net`.

Przeanalizujmy zdarzenia, które będą miały miejsce, gdy węzeł o nazwie `jukebox.linuxtoys.net` (o adresie IP `199.170.179.18`) spróbuje nawiązać połączenie z serwerem za pomocą protokołu *telnet*. W omawianym przypadku dystrybucja Linuksa to Fedora, która używa demona `xinetd` do nasłuchu żądań usług powiązanych z osłonami TCP.

1. Demon `xinetd` otrzymuje żądanie nawiązania połączenia.
2. Demon `xinetd` rozpoczyna porównywanie adresu oraz nazwy `jukebox.linuxtoys.net` z regułami w pliku */etc/hosts.allow*. Przetwarzanie pliku rozpoczyna się od początku i posuwa się ku końcowi aż do znalezienia dopasowania wzorca. Zarówno demon (program obsługujący usługi sieciowe w systemie Fedora), jak i adres IP oraz nazwa klienta nawiązującego połączenie muszą zostać dopasowane do informacji znajdujących się w pliku *hosts.allow*. W omawianym przykładzie druga reguła powoduje dopasowanie żądania:

   ```
   in.telnetd: 199.170.177., .linuxtoys.net
   ```

3. Węzeł `jukebox` nie znajduje się w podzbiorze `199.170.177`, ale w domenie `linuxtoys.net`. Po odnalezieniu dopasowania demon `xinetd` przerywa przeszukiwanie pliku.

Co się jednak stanie, jeżeli `jukebox` spróbuje nawiązać połączenie za pomocą protokołu *CUPS-lpd*? W takim przypadku nie zostanie dopasowana żadna reguła z pliku *hosts.allow*. Jedyny wiersz odnoszący się do demona `lpd` nie dotyczy ani podsieci `199.170.179`, ani domeny `linuxtoys.net`. Demon `xinetd` rozpocznie więc sprawdzanie pliku *hosts.deny*. Wpis `ALL: ALL` powoduje dopasowanie wszystkiego, dlatego też demon `tcpd` odrzuci próbę nawiązania połączenia.

W pliku *hosts.allow* została również użyta flaga `ALL`. W tym przypadku demon `xinetd` został poinformowany, aby zezwalać na absolutnie wszystkie połączenia z usługą FTP. Takie rozwiązanie jest odpowiednie dla uruchomionego anonimowego serwera FTP, do którego każdy użytkownik internetu może uzyskać dostęp. Jeżeli użytkownik nie ma zamiaru uruchamiania anonimowego serwera FTP, prawdopodobnie nie należy używać flagi `ALL`.

Dobrą i praktyczną regułą jest utrzymywanie jak najbardziej restrykcyjnych reguł w plikach *hosts.allow* oraz *hosts.deny*, a następnie udostępnianie jedynie tych usług, które są absolutnie niezbędne. Ponadto dostęp powinny otrzymać tylko te systemy, którego go potrzebują. Używanie flagi `ALL` w celu nadania dostępu do określonej usługi może być znacznie łatwiejsze niż podawanie długiej listy podsieci lub domen, ale warto poświęcić kilka minut na odpowiednie ustawienie reguł bezpieczeństwa, zamiast potem spędzać wiele godzin na naprawianiu systemu po włamaniu.

W systemach Linux, które używają usługi `xinetd`, można jeszcze bardziej ograniczyć dostęp do usług za pomocą różnych opcji w pliku */etc/xinetd.conf*. Możliwe jest nawet ograniczenie dostępu do usług w określonych godzinach. Więcej informacji na temat tych opcji znajduje się podręczniku demona `xinetd` (wyświetlany po wydaniu z poziomu powłoki polecenia `man xinetd`).

Zrozumienie techniki ataków

Atak na system komputerowy przybiera różne formy, w zależności od celu i zasobów atakującego. Niektórzy atakujący chcą być jak najbardziej destrukcyjni, podczas gdy inni chcą przeniknąć do komputera, a następnie wykorzystać jego zasoby do własnych, często nikczemnych celów. Z kolei celem jeszcze innych mogą być dane finansowe lub szantaż. Poniżej przedstawiono trzy główne kategorie ataków:

- **Odmowa usługi** (DOS, ang. *Denial of Service*) — najłatwiejszym atakiem do przeprowadzenia jest DOS, czyli odmowa usługi. Podstawowym celem tego rodzaju ataku jest zakłócenie działalności zdalnej witryny poprzez przeciążenie jej nieistotnymi danymi. Ataki typu DOS mogą po prostu polegać na wysyłaniu w ciągu sekundy tysięcy żądań dostępu do strony. Taki rodzaj ataku jest bardzo łatwy do przeprowadzenia, ale również łatwo można się przed nim chronić. Po ustaleniu źródła ataku sprawę powinien rozwiązać zwykły telefon do ISP atakującego.
- **Rozproszony atak DOS** (ang. *Distributed Denial of Service*) — bardziej zaawansowane ataki DOS są nazywane rozproszonymi atakami typu DOS. Ataki typu DDOS są trudniejsze do przeprowadzenia i niemal niemożliwe do zatrzymania. W tej formie ataku atakujący przejmuje kontrolę nad setkami lub nawet tysiącami słabo zabezpieczonych komputerów. Następnie atakujący wykorzystuje je do wysyłania nieistotnych danych do pojedynczego węzła internetowego. Wynikiem takiego działania jest to, że siła pojedynczego atakującego zostaje zwielokrotniona tysiące razy. Zamiast ataku tylko z jednego kierunku, jak ma to miejsce w przypadku ataków DOS, atak nachodzi z tysięcy miejsc. Najlepszą ochroną przed atakiem DDOS jest kontakt z własnym ISP i próba dowiedzenia się, czy ISP może filtrować ruch na swoich routerach brzegowych.

  Wiele osób, które nie przywiązują większej wagi do kwestii bezpieczeństwa, używa wymówki „nie mam niczego na komputerze, co mogłoby zainteresować atakującego". Problem jednak w tym, że atakujący ma dużo powodów, aby wykorzystać komputer takiego użytkownika. Atakujący może zmienić taki komputer w agenta, który później będzie użyty podczas ataku DDOS. Niejednokrotnie zdarzało się już, że organy ścigania pojawiały się u użytkownika tak przejętego komputera i zadawały pytania dotyczące zagrożeń przez niego powodowanych. Poprzez ignorowanie kwestii bezpieczeństwa użytkownicy narażają się na dużą odpowiedzialność.
- **Atak intruza** — zdalne wykorzystanie zasobów atakowanego komputera. Atakujący musi wcześniej znaleźć lukę, którą może wykorzystać. Bez informacji, takich jak hasła bądź zaszyfrowane klucze, atakujący musi skanować atakowany komputer i przekonać się, jakie usługi oferuje. Istnieje pewne prawdopodobieństwo, że jedna z dostępnych usług jest słabo zabezpieczona i atakujący może wykorzystać jej znane słabości do własnych celów.

  Za najlepsze narzędzie do skanowania węzła pod kątem dostępnych usług uważa się *nmap* (warto zwrócić uwagę, że narzędzia *nmap* można użyć zarówno w dobrych, jak i złych celach). Kiedy atakujący zdobędzie listę usług działających na komputerze ofiary, musi znaleźć sposób wykorzystania jednej z nich do uzyskania uprawnień dostępu do systemu. Zwykle ten etap jest przeprowadzany za pomocą programu nazywanego *exploit*.

Podczas gdy ataki typu DOS są destrukcyjne, ataki intruzów przynoszą jeszcze więcej szkód. Powody tego są różne, ale wyniki zawsze takie same. Nieproszony gość wprowadza się do komputera i używa go w sposób, nad którym właściciel nie ma żadnej kontroli.

Ochrona przed atakami typu DOS

Jak wyjaśniono wcześniej, ataki typu DOS próbują złamać komputer lub przynajmniej obniżyć jego wydajność do zupełnie niewystarczającego poziomu. Istnieje kilka różnych sposobów użycia tego typu ataków. Często spotykanym działaniem jest próba przeciążenia zasobów systemu, takich jak ilość wolnej przestrzeni, lub połączenia z internetem. W podrozdziale zostaną zaprezentowane niektóre często spotykane rodzaje ataków oraz sposoby obrony przed nimi.

Mailbombing

Termin *mailbombing* oznacza masowe wysyłanie wiadomości e-mail do określonego użytkownika lub systemu aż do zapełnienia wolnej przestrzeni. Istnieje kilka sposobów obrony przed takim atakiem. Jednym z nich jest użycie narzędzia filtrującego pocztę elektroniczną o nazwie *Procmail*. Jeśli używanym agentem transportu poczty jest sendmail, można skonfigurować demona `sendmail`.

Blokowanie poczty za pomocą Procmail

Narzędzie filtrowania poczty elektronicznej *Procmail* jest instalowane domyślnie w systemach Fedora, RHEL oraz kilku innych dystrybucjach. *Procmail* jest ściśle zintegrowany z demonem `sendmail`, dlatego też może być używany do selektywnego blokowania lub filtrowania określonego rodzaju poczty elektronicznej. Więcej informacji na temat narzędzia *Procmail* znajduje się na witrynie *http://www.procmail.org*.

W celu włączenia narzędzia *Procmail* dla danego konta użytkownika należy w jego katalogu domowym utworzyć plik *.procmailrc*. Ten plik powinien mieć uprawnienia `0600` (czyli do odczytu tylko przez użytkownika i nikogo więcej). Następnie należy wpisać podane poniżej wiersze, zastępując słowo „evilmailer" rzeczywistym adresem e-mail, za którym kryje się atakujący:

```
# Usunięcie poczty elektronicznej od evilmailer.
:0
* ^From.*evilmailer
/dev/null
```

Reguła narzędzia *Procmail* powoduje wyszukanie wiersza `From` na początku każdej wiadomości e-mail i sprawdzenie, czy zawiera ciąg tekstowy `evilmailer`. Jeżeli ten ciąg tekstowy zostanie znaleziony, wiadomość jest wysyłana do urządzenia */dev/null* (czyli narzędzie efektywnie pozbywa się jej).

Blokowanie poczty za pomocą sendmail

Narzędzie *Procmail* sprawuje się całkiem dobrze, gdy tylko jeden użytkownik jest atakowany poprzez mailbombing. Jeśli jednak ofiarą tego typu ataku pada większa liczba użytkowników, prawdopodobnie należy skonfigurować demona `sendmail` tak, aby blokował całą pocztę pochodzącą od atakującego. Wspomniana konfiguracja polega na umieszczeniu adresu atakującego lub nazwy systemu w pliku *access* znajdującym się w katalogu */etc/mail*.

Każdy wiersz pliku *access* składa się z adresu e-mail, nazwy węzła, domeny lub adresu IP, a następnie znaku tabulatora, po którym znajduje się słowo kluczowe określające podejmowane działanie, gdy osoba wskazana w wierszu będzie przysyłała wiadomość. Możliwymi do zastosowania słowami kluczowymi są `OK`, `RELAY`, `REJECT`, `DISCARD` oraz `ERROR`. Użycie słowa kluczowego `REJECT` spowoduje odrzucenie wiadomości od nadawcy i wygenerowanie komunikatu błędu. Słowo kluczowe `DISCARD` spowoduje ciche pozbycie się wiadomości bez generowania dla nadawcy komunikatu błędu. Użytkownik może nawet zwrócić własny komunikat błędu, wykorzystując słowo kluczowe `ERROR`.

Plik */etc/mail/access* może przedstawiać się następująco:

```
# Check the /usr/share/doc/sendmail/README.cf file for a description
# of the format of this file. (search for access_db in that file)
# The /usr/share/doc/sendmail/README.cf is part of the sendmail-doc
# package.
#
# by default we allow relaying from localhost...
localhost.localdomain           RELAY
localhost                       RELAY
127.0.0.1                       RELAY
#
# Senders we want to Block
#
evilmailer@yahoo.com    REJECT
stimpy.glaci.com        REJECT
cyberpromo.com          DISCARD
199.170.176.99          ERROR:"550 Die Spammer Scum!"
199.170.177             ERROR:"550 Email Refused"
```

Jak w przypadku większości plików konfiguracyjnych systemu Linux, wiersze rozpoczynające się od znaku # są komentarzem. W zaprezentowanym pliku lista zablokowanych spamerów znajduje się na końcu. Warto zwrócić uwagę, że blokowane może odbywać się poprzez podanie pełnego adresu e-mail, pełnej nazwy węzła, samej domeny, adresu IP lub podsieci.

W celu zablokowania określonego adresu e-mail lub węzła, z którego pochodzi atak, należy zalogować się do systemu jako użytkownik root, przeprowadzić edycję pliku */etc/mail/*
↪*access* i dodać wiersz wraz ze słowem kluczowym `DISCARD` i adresem atakującego.

Po zapisaniu pliku i opuszczeniu edytora plik musi zostać skonwertowany do postaci zindeksowanej znakami hash bazy danych o nazwie *access.db*. Tak utworzona baza danych zostanie uaktualniona w trakcie kolejnego uruchomienia `sendmail`. W dystrybucji Fedora oraz innych systemach Red Hat natychmiastowa konwersja bazy danych następuje po wydaniu poleceń:

```
# cd /etc/mail
# make
```

`Sendmail` powinien odrzucać wiadomości e-mail pochodzące z dodanych adresów.

Przekazywanie spamu

Usługa poczty elektronicznej może zostać nadużyta, gdy system będzie wykorzystany do przekazywania spamu. Pojęcie *spam* oznacza niechcianą wiadomość e-mail, a samo zjawisko przesyłania spamu staje się coraz bardzie dokuczliwe. Termin *przekazywanie* odnosi się do funkcji serwera poczty elektronicznej, która wysyła pocztę otrzymaną z innego serwera. (W normalnych warunkach tylko użytkownicy z prawidłowymi kontami e-mail na serwerze mogą używać serwera poczty do przekazywania wiadomości. Serwer poczty skonfigurowany jako otwarty przekaźnik pozwoli każdemu za jego pomocą na przekazywanie wiadomości e-mail, dlatego jest to bardzo zła praktyka).

Bardzo często spamerzy wysyłają swoje irytujące wiadomości ze zwykłego konta typu dial-up. Potrzebują tylko pewnego rodzaju serwera poczty o dużej pojemności do zaakceptowania i buforowania takiej liczby wiadomości. Spamerzy dostarczają spam do serwera w postaci jednej olbrzymiej serii, a następnie wylogowują się, pozwalając serwerowi na wykonanie całej pracy dostarczenia tych wiadomości wielu ofiarom.

Oczywiście, żaden z odpowiedzialnych ISP nie uczestniczy w takim procederze, tak więc spamerzy muszą przejmować serwery innych ISP, aby wykonać tę brudną robotę. Dopuszczenie do sytuacji, w której serwer poczty zostanie przejęty przez spamera i wykorzystany do wysłania spamu, może mieć druzgocący efekt na system oraz reputację. Na szczęście otwarte przekazywanie poczty jest domyślnie wyłączone w systemach Fedora oraz Red Hat Enterprise Linux. Otwarte przekazywanie poczty jest więc jedną z kwestii bezpieczeństwa, o którą nie należy się martwić.

Użytkownik może pozwolić określonym węzłom lub domenom na przekazywanie poczty poprzez system po umieszczeniu tych nadawców w pliku */etc/mail/access* wraz ze słowem kluczowym `RELAY`. Domyślnie przekazywanie poczty jest możliwe tylko z poziomu komputera lokalnego.

Jednym z pakietów, którego używanie do filtrowania spamu na serwerze pocztowym warto rozważyć, jest *SpamAssassin*. Analizuje on tekst przychodzącej wiadomości i próbuje filtrować te wiadomości, które zostały określone jako spam. Dokładniejszy opis pakietu *SpamAssassin* znajdzie się w rozdziale 25.

Atak typu smurf

Termin *smurfing* odnosi się do określonego rodzaju ataku DOS, którego celem jest zalanie połączenia internetowego pakietami. Obrona przed takim atakiem może być bardzo trudna, ponieważ niełatwo jest wyśledzić atakującego. Poniżej objaśniono, w jaki sposób przebiega smurfing.

Atakujący korzysta z protokołu ICMP, czyli usługi, której celem jest sprawdzenie prędkości i dostępności połączeń sieciowych. Za pomocą polecenia `ping` użytkownik może wysłać pakiet sieciowy ze swojego komputera do innego komputera w internecie. Zdalny komputer rozpozna pakiet jako żądanie ICMP i odpowie również za pomocą pakietu. Następnie komputer nadawcy wyświetli komunikat wskazujący, że zdalny system funkcjonuje, oraz poinformuje, ile czasu zabrała mu odpowiedź.

Atak typu smurfing wykorzystuje zniekształcone żądanie ICMP, tak aby ukryć w sieci komputer nadawcy. Jest to możliwe poprzez wysyłanie żądań `ping` od nieświadomych tego użytkowników sieci, powodując tym samym, że odpowiedź jest duplikowana dziesiątki lub nawet setki razy. Adres docelowy polecenia `ping` jest ustalony jako cała podsieć w obrębie pojedynczego węzła. Adres zwrotny jest fałszowany tak, aby wskazywał adres nieświadomego użytkownika, a nie faktycznego nadawcy. Kiedy pakiet ICMP przybywa od nieświadomego pośrednika sieciowego, wtedy każdy węzeł podsieci odpowiada na żądanie `ping`! Co więcej, odpowiedź jest kierowana do nieświadomego użytkownika, a nie do atakującego. Jeżeli sieć nieświadomych pośredników składa się z setek komputerów, wówczas połączenie internetowe ofiary może być szybko zapchane.

Najlepszą reakcją w przypadku tego rodzaju ataku jest kontakt z organizacją użytą jako nieświadomy pośrednik i poinformowanie jej o nadużyciu. Zazwyczaj w celu zatrzymania ataków pośrednik musi tylko przekonfigurować router. Jeżeli organizacja wykorzystana w charakterze przekaźnika nie wykazuje woli współpracy, minimalizacja efektu ataku jest możliwa poprzez zablokowanie protokołu ICMP w routerze. W ten sposób duży ruch sieciowy będzie trzymany z dala od sieci wewnętrznej. Jeżeli uda się przekonać ISP do zablokowania pakietów ICMP skierowanych do zaatakowanej sieci, może to jeszcze bardziej pomóc. (Warto w tym miejscu zwrócić uwagę na toczącą się debatę, czy blokowanie pakietów ICMP jest dobrym, czy złym pomysłem, ponieważ usługi ICMP mogą być użyteczne w wielu zadaniach administracyjnych).

Ochrona przed rozproszonymi atakami typu DOS

Ataki typu DDOS są znacznie trudniejsze do zainicjowania i wyjątkowo trudne do zatrzymania. Atak DDOS rozpoczyna się od penetracji setek lub nawet tysięcy słabo zabezpieczonych komputerów. Następnie takie komputery są używane w ataku na pojedynczy węzeł wybrany na podstawie zachcianki atakującego.

Wraz z nadejściem ery modemów DSL oraz kablowych miliony użytkowników cieszy się dostępem do internetu, niemal bez ograniczeń w prędkości. W pośpiechu, aby jak najszybciej rozpocząć korzystanie z zasobów internetu, wielu takich użytkowników zaniedbuje nawet podstawowe zasady bezpieczeństwa. Ponieważ większość tych osób używa systemów operacyjnych Microsoft Windows, często bardzo szybko dochodzi do zarażenia ich robakiem bądź wirusem komputerowym. Kiedy komputer użytkownika zostanie zaatakowany, bardzo często robak lub wirus instaluje program powodujący ciche połączenie z autorem i poinformowanie go, że jest gotowy na *spełnianie rozkazów*.

Na życzenie pana zainfekowane komputery mogą być użyte w celu wysłania strumienia nieistotnych danych do wybranego węzła. Wspólnie z tysiącami innych zainfekowanych komputerów *script kiddie* osiąga moc pozwalającą rzucić na kolana niemal każdą witrynę w internecie.

Wykrycie ataku DDOS jest podobne do odkrycia ataku DOS i wiąże się z wystąpieniem jednego lub większej liczby następujących sygnałów:

- nieprzerwana, nasycona transmisja danych;
- brak zmniejszenia się ilości danych nawet w okresie poza szczytem;

- setki lub nawet tysiące jednoczesnych połączeń sieciowych;
- wyjątkowo słaba wydajność systemu.

W celu określenia, czy transmisja danych została nasycona, należy wydać polecenie `ping` do zewnętrznego węzła. Znacznie większe niż zwykle opóźnienie wskazuje na martwą bramkę dostępu. Zwykłe opóźnienie polecenia `ping` (a więc czas wymagany przez polecenie `ping` na otrzymanie odpowiedzi) przedstawia się podobnie do zaprezentowanego poniżej:

```
# ping www.example.com
PING www.example.com (192.0.34.166) from 10.0.0.11: 56(84) bytes of data
64 bytes from 192.0.34.166: icmp_seq=1 ttl=49 time=40.1 ms
64 bytes from 192.0.34.166: icmp_seq=2 ttl=49 time=42.5 ms
64 bytes from 192.0.34.166: icmp_seq=3 ttl=49 time=39.5 ms
64 bytes from 192.0.34.166: icmp_seq=4 ttl=49 time=38.4 ms
64 bytes from 192.0.34.166: icmp_seq=5 ttl=49 time=39.0 ms

--- www.example.com ping statistics ---
5 packets transmitted, 5 received, 0% loss, time 4035ms
rtt min/avg/max/mdev = 38.472/39.971/42.584/1.432 ms
```

W powyższym przykładzie przeciętny czas potrzebny poleceniu `ping` na wysłanie pakietu i uzyskanie odpowiedzi zwrotnej wynosi około 39 tysięcznych sekundy.

Wyniki polecenia `ping` do przeciążonego serwera będą przedstawiały się podobnie do poniższych:

```
# ping www.example.com
PING www.example.com (192.0.34.166): from 10.0.0.11: 56(84)bytes of data
64 bytes from 192.0.34.166: icmp_seq=1 ttl=62 time=1252 ms
64 bytes from 192.0.34.166: icmp_seq=2 ttl=62 time=1218 ms
64 bytes from 192.0.34.166: icmp_seq=3 ttl=62 time=1290 ms
64 bytes from 192.0.34.166: icmp_seq=4 ttl=62 time=1288 ms
64 bytes from 192.0.34.166: icmp_seq=5 ttl=62 time=1241 ms

--- www.example.com ping statistics ---
5 packets transmitted, 5 received, 0% loss, time 5032ms
rtt min/avg/max/mdev = 1218.059/1258.384/1290.861/28.000 ms
```

W powyższym przykładzie czas wymagany przez pakiet `ping` wynosi około 1,3 sekundy. W porównaniu z wcześniejszym przykładem opóźnienie wzrosło o 31 razy! To wyraźny sygnał, że należy sprawdzić wykorzystanie transmisji danych.

W celu dokładniejszego sprawdzenia przepustowości danych można użyć narzędzia takiego jak *ttcp*. Sprawdzenie połączenia za pomocą narzędzia *ttcp* wymaga zainstalowania pakietu *ttcp* na komputerach zarówno wewnątrz, jak i na zewnątrz sieci. (Pakiet *ttcp* jest dostępny w dystrybucji Fedora Core oraz innych systemach Linux). Jeżeli użytkownik nie jest pewny, czy pakiet *ttcp* został zainstalowany, wystarczy z poziomu powłoki wydać polecenie `ttcp`. Polecenie `ttcp` powinno wyświetlić dane wyjściowe podobne do przedstawionych poniżej:

```
# ttcp
Usage: ttcp -t [-options] host [ < in ]
ttcp -r [-options > out]
Common options:
```

```
-l ##    length of bufs read from or written to network (default 8192)
-u       use UDP instead of TCP
-p ##    port number to send to or listen at (default 5001)
-s       -t: source a pattern to network
-r: sink (discard) all data from network
-A       align the start of buffers to this modulus (default 16384)
-O       start buffers at this offset from the modulus (default 0)
-v       verbose: print more statistics
-d       set SO_DEBUG socket option
-b ##    set socket buffer size (if supported)
-f X     format for rate: k,K = kilo{bit,byte}; m,M = mega; g,G = giga
Options specific to -t:
-n##     number of source bufs written to network (default 2048)
-D       don't buffer TCP writes (sets TCP_NODELAY socket option)
-w ## number of microseconds to wait between each write
Options specific to -r:
-B       for -s, only output full blocks as specified by -l (for TAR)
-T       "touch": access each byte as it's read
-I if Specify the network interface (e.g. eth0) to use
```

Pierwszym krokiem jest uruchomienie procesu odbioru danych na serwerze:

```
# ttcp -rs
ttcp-r: buflen=8192, nbuf=2048, align=16384/0, port=5001  tcp
ttcp-r: socket
```

Flaga `-r` wskazuje, że komputer serwera będzie odbiorcą danych. Z kolei flaga `-s` w połączeniu z flagą `-r` informuje `ttcp`, że wszystkie otrzymane dane mają zostać zignorowane.

Kolejnym krokiem jest posiadanie kogoś na zewnątrz wraz z łączem o podobnej prędkości i ustawienie procesu wysyłającego `ttcp`:

```
# ttcp -ts server.example.com
ttcp-t: buflen=8192, nbuf=2048, align=16384/0, port=5001  tcp
-> server.example.com
ttcp-t: socket
ttcp-t: connect
```

Następnie należy na kilka minut uruchomić proces, a potem nacisnąć klawisze *Ctrl+C* po stronie wysyłającego, co spowoduje przerwanie testu. Strona odbiorcy w ciągu kilku chwil dokona obliczeń oraz wyświetli wyniki:

```
# ttcp -rs
ttcp-r: buflen=8192, nbuf=2048, align=16384/0, port=5001  tcp
ttcp-r: socket
ttcp-r: accept from 64.223.17.21
ttcp-r: 2102496 bytes in 70.02 real seconds = 29.32 KB/sec +++
ttcp-r: 1226 I/O calls, msec/call = 58.49, calls/sec = 17.51
ttcp-r: 0.0user 0.0sys 1:10real 0% 0i+0d 0maxrss 0+2pf 0+0csw
```

W powyższym przykładzie przeciętna przepustowość łącza wynosi 29,32 kilobajta na sekundę. W przypadku łącza dotkniętego atakiem DDOS obliczona liczba może być jedynie małym ułamkiem rzeczywistej przepustowości łącza.

Jeżeli łącze danych wskazuje na przeciążenie, kolejnym krokiem jest próba określenia źródła połączeń. Bardzo efektywnym sposobem realizacji tego zadania jest użycie polecenia

netstat, które jest częścią podstawowej instalacji dystrybucji Fedora. Informacje dotyczące połączenia zostaną wyświetlone po wydaniu polecenia:

```
# netstat -tupn
```

W tabeli 6.3 zostały opisane wszystkie parametry użyte w poleceniu netstat.

Tabela 6.3. *Parametry polecenia netstat*

Parametr	Opis
-t, --tcp	Pokazuje gniazdo połączeń TCP.
-u, --udp	Pokazuje gniazdo połączeń UDP.
-p, --program	Pokazuje PID oraz nazwę programu, do którego należy każde gniazdo.
-n, --numeric	Pokazuje adres w postaci liczbowej zamiast próby określenia symbolicznego węzła, portu bądź nazw użytkowników.

Poniżej przedstawiono przykładowe dane wyjściowe polecenia netstat:

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address  Foreign Address   State    PID/Program name
tcp    0    0 65.213.7.96:22    13.29.132.19:12545   ESTABLISHED 32376/sshd
tcp    0  224 65.213.7.96:22    13.29.210.13:29250   ESTABLISHED 13858/sshd
tcp    0    0 65.213.7.96:6667  13.29.194.190:33452  ESTABLISHED 1870/ircd
tcp    0    0 65.213.7.96:6667  216.39.144.152:42709 ESTABLISHED 1870/ircd
tcp    0    0 65.213.7.96:42352 67.113.1.99:53       TIME_WAIT   -
tcp    0    0 65.213.7.96:42354 83.152.6.9:113       TIME_WAIT   -
tcp    0    0 65.213.7.96:42351 83.152.6.9:113       TIME_WAIT   -
tcp    0    0 127.0.0.1:42355   127.0.0.1:783        TIME_WAIT   -
tcp    0    0 127.0.0.1:783     127.0.0.1:42353      TIME_WAIT   -
tcp    0    0 65.213.7.96:42348 19.15.11.1:25        TIME_WAIT   -
```

Dane wyjściowe zostały zorganizowane w kolumny zdefiniowane następująco:

- Proto — protokół używany przez gniazdo.
- Recv-Q — liczba bajtów, które nie zostały jeszcze skopiowane przez program użytkownika dołączony do danego gniazda.
- Send-Q — liczba bajtów niezatwierdzonych przez węzeł.
- Local Address — adres oraz numer portu lokalnego zakończenia danego gniazda.
- Foreign Address — adres i numer portu zdalnej końcówki danego gniazda.
- State — bieżący stan gniazda. Lista stanów gniazda została przedstawiona w tabeli 6.4.
- PID/Program name — identyfikator procesu oraz nazwa programu procesu, który jest właścicielem danego gniazda.

Podczas ataku typu DOS zdalny adres jest z reguły taki sam dla każdego połączenia. W takim przypadku należy po prostu wyszukać właściciela adresu IP na stronie *http://www.*
➥*arin.net/whois/* i powiadomić swojego ISP.

Tabela 6.4. *Stany gniazda*

Stan	Opis
ESTABLISHED	Gniazdo ustanowiło połączenie.
SYN_SENT	Gniazdo aktywnie próbuje nawiązać połączenie.
SYN_RECV	Z sieci otrzymano żądanie połączenia.
FIN_WAIT1	Gniazdo zamknięte i wyłączone.
FIN_WAIT2	Gniazdo oczekuje na zdalne zamknięcie.
TIME_WAIT	Po zamknięciu gniazdo nadal oczekuje na obsługę pakietów wciąż znajdujących się w sieci.
CLOSED	Gniazdo nie jest używane.
CLOSE_WAIT	Zdalna końcówka gniazda została zamknięta, oczekiwanie na zamknięcie gniazda.
LAST_ACK	Zdalna końcówka gniazda została zamknięta, gniazdo zostało zamknięte, oczekiwanie na zatwierdzenie.
LISTEN	Obie końcówki połączenia są zamknięte, ale jeszcze nie wszystkie dane zostały wysłane.
CLOSING	Obie strony połączenia zostają zamknięte, ale nie wszystkie dane zdążyły zostać wysłane.
UNKNOWN	Stan gniazda jest nieznany.

Podczas ataku typu DDOS zdalny adres będzie prawdopodobnie inny dla każdego połączenia. W takim przypadku wyśledzenie wszystkich atakujących staje się niemożliwe, ponieważ są ich tysiące. Najlepszym sposobem obrony pozostaje wówczas skontaktowanie się z ISP i sprawdzenie, czy może on ograniczyć ruch na swoich routerach brzegowych.

Ochrona przed atakami intruzów

Crackerzy posiadają szeroką gamę narzędzi i technik, które są wykorzystywane podczas włamywania się do systemu. Ataki intruzów skupiają się na wykorzystaniu luk bezpieczeństwa, co umożliwia crackerom uzyskanie kontroli nad systemem (i potencjalnie dokonanie większych zniszczeń, niż z zewnątrz).

Na szczęście istnieje wiele narzędzi i technik pomagających w obronie przed atakami intruzów. W podrozdziale zostaną przedstawione najczęściej stosowane metody włamań oraz narzędzia służące do ochrony systemu. Mimo że przykłady przedstawiają funkcje stosowane w Fedorze oraz innych systemach Red Hat, to omówione narzędzia i techniki mają zastosowanie dla dowolnego systemu Linux bądź bazującego na Uniksie.

Szacowanie dostępu do usług sieciowych

Systemy Linux oraz Unix dostarczają wiele usług sieciowych, a wraz z nimi crackerzy uzyskują duże możliwości ataku. Użytkownik powinien znać te usługi oraz wiedzieć, w jaki sposób ograniczyć do nich dostęp.

Co oznacza pojęcie „usługa sieciowa”? Zasadniczo usługa sieciowa to każde zadanie wykonywane przez komputer, które wymaga wysyłania i odbierania informacji przez sieć za

pomocą zdefiniowanego zbioru reguł. Przekierowywanie poczty elektronicznej jest usługą sieciową, podobnie jak serwowanie stron internetowych. System Linux może potencjalnie oferować tysiące usług. Wiele z nich zostało wymienionych w pliku */etc/services*. Spójrzmy na przykładowy fragment tego pliku:

```
# /etc/services:
# service-name  port/protocol  [aliases ...]   [# comment]
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp-data        20/udp
# 21 is registered to ftp, but also used by fsp
ftp             21/tcp
ftp             21/udp          fsp fspd
ssh             22/tcp                  # SSH Remote Login Protocol
ssh             22/udp                  # SSH Remote Login Protocol
telnet          23/tcp
telnet          23/udp
# 24 - private mail system
smtp            25/tcp          mail
```

Po wierszach zawierających komentarze znajdują się trzy kolumny informacji. Lewa kolumna zawiera nazwę każdej usługi. W środkowej kolumnie został zdefiniowany numer portu oraz rodzaj protokołu używany przez daną usługę. Z kolei prawa kolumna zawiera opcjonalny alias bądź listę aliasów tej usługi.

Jako przykład przeanalizujemy ostatni wiersz w zaprezentowanym powyżej fragmencie pliku. Wiersz opisuje usługę SMTP (ang. *Simple Mail Transfer Protocol*), która jest usługą używaną w celu dostarczania poczty elektronicznej przez internet. Środkowa kolumna zawiera wpis `25/tcp`, który wskazuje, że protokół SMTP używa portu numer 25 oraz protokołu Transmission Control Protocol (TCP).

Czym dokładnie jest *numer portu*? To unikalna liczba, która została ustalona dla określonej usługi sieciowej. Pozwala na prawidłowe przekazywanie usług sieciowych do oprogramowania obsługującego tę usługę. Na przykład podczas dostarczania wiadomości e-mail z komputera nadawcy do komputera odbiorcy zdalny system musi w pierwszej kolejności nawiązać połączenie z komputerem odbiorcy. Komputer odbiorcy otrzymuje żądanie połączenia, analizuje je, stwierdza, że jest przeznaczone dla portu numer 25, a więc wie, że to połączenie powinno zostać obsłużone przez program do obsługi poczty elektronicznej (którym najprawdopodobniej jest sendmail).

Wspomniano wcześniej, że SMTP wymaga protokołu TCP. Niektóre usługi używają natomiast protokołu User Datagram Protocol (UDP). Na użytek dyskusji dotyczącej zagadnień bezpieczeństwa użytkownik powinien wiedzieć, że protokoły TCP i UDP oferują różne sposoby pakowania informacji i wysyłania ich przez połączenie sieciowe. Połączenie TCP zawiera mechanizm wykrywania błędów i ponownego przesyłania utraconych danych. Natomiast UDP nie sprawdza, czy dane zostały dostarczone w komplecie i nietknięte, co oznacza szybszy sposób wysyłania mniej ważnych informacji.

Wyłączanie usług sieciowych

Mimo że istnieją setki usług (wymienione w pliku */etc/services* wraz z oficjalnymi numerami portów), które potencjalnie mogą stać się celem ataku systemu Linux, to w rzeczywistości niewiele z nich jest zainstalowanych. W systemach Fedora oraz RHEL większość usług sieciowych jest uruchamiana za pomocą procesu `xinetd` albo skryptów startowych w katalogu */etc/init.d*. Inne systemy Linux używają procesu `inetd` zamiast `xinetd`.

Demony `xinetd` oraz `inetd` są demonami, które nasłuchują dużą liczbę portów sieciowych. Kiedy na określonym porcie następuje próba nawiązania połączenia, demon `xinetd` lub `inetd` automatycznie uruchamia odpowiedni program do obsługi danej usługi i pozwala na połączenie.

W przypadku demona `xinetd` plik konfiguracyjny (*/etc/xinetd.conf*) jest używany w celu dostarczenia ustawień domyślnych serwera `xinetd`. Katalog */etc/xinetd.d* zawiera pliki informujące `xinetd`, które porty powinien nadsłuchiwać oraz jakie programy uruchamiać (w przypadku demona `inetd` używany jest tylko plik */etc/inetd.conf*). Każdy plik w katalogu */etc/xinetd.d* zawiera informacje konfiguracyjne dla pojedynczego urządzenia i z reguły nosi nazwę odnoszącą się do konfigurowanej usługi. Na przykład w celu włączenia usługi *rsync* należy przeprowadzić edycję pliku *rsync* w katalogu */etc/xinetd.d* oraz odszukać następujący fragment pliku:

```
service rsync
{
disable = yes
socket_type       = stream
wait              = no
user              = root
server            = /usr/bin/rsync
server_args       = --daemon
log_on_failure    += USERID
}
```

Warto zwrócić uwagę na pierwszy wiersz powyższego fragmentu pliku, w którym następuje identyfikacja usługi jako *rsync*. Jest to nazwa dokładnie odpowiadająca nazwie usługi wymienionej w pliku */etc/services* i nasłuchującej protokołów TCP i UDP na porcie 873. Przykład pokazuje wyraźnie, że usługa jest domyślnie wyłączona (`disable=yes`). Włączenie usługi wymaga zmiany wymienionego wiersza na `disable=no`. Dlatego też po włączeniu usługi wiersz `disable` będzie przedstawiał się następująco:

```
disable = no
```

Jeżeli komputer ma działać jako serwer FTP, usługa *rsync* jest jedną z przeznaczonych do włączenia. Pozwala ona użytkownikom na używanie klienta *rsync* (zawierającego algorytm wyszukiwania sumy kontrolnej) do pobierania plików z serwera. Za pomocą tej funkcji użytkownicy mogą ponownie uruchomić przerwane pobieranie pliku bez potrzeby rozpoczynania tego procesu od początku.

Ponieważ większość usług jest domyślnie wyłączona, komputer może stać się mniej bezpieczny tylko w wyniku działań użytkownika. Należy dwukrotnie sprawdzić, czy niebezpieczne usługi, takie jak *rlogin* i *rsh* (które w systemach Fedora i RHEL znajdują się w pakiecie *rsh-server*), są wyłączone również w plikach */etc/xinetd.d/rlogin* i *rsh* (wiersz `disabled=yes`).

Usługa zdalnego logowania może być aktywna, ale warto zablokować używanie plików */etc/host.equiv* i *.rhosts* oraz wymóc podawania hasła podczas każdego użycia `rlogin`. Zamiast wyłączać usługę, znacznie lepszym rozwiązaniem jest odszukanie wiersza `server` w pliku *rsh* (`server = /usr/sbin/in.rshd`), a następnie na jego końcu dodanie spacji i opcji `-L`.

Po wprowadzeniu modyfikacji należy wysłać sygnał procesowi `xinetd`, aby plik konfiguracyjny został ponownie odczytany. Najszybszym sposobem realizacji takiego zadania w systemach Fedora i RHEL jest przeładowanie usługi `xinetd`. Przeładowanie usługi wymaga wydania z poziomu powłoki następującego polecenia jako użytkownik root:

```
# service xinetd reload
Reloading configuration: [ OK ]
```

Inną możliwością jest bezpośrednie nakazanie procesowi `xinetd` ponownego odczytania pliku konfiguracyjnego poprzez wysłanie mu sygnału `SIGHUP`. To rozwiązanie działa, jeśli używany jest demon `inetd` (w systemach takich jak Debian lub Slackware), i wymusza ponowne odczytanie pliku */etc/inetd.conf*. Na przykład w celu ponownego odczytania pliku konfiguracyjnego należy jako użytkownik root wydać następujące polecenie:

```
# killall -s SIGHUP inetd
```

To już wszystko, usługa *rsync* została włączona. Po dostarczeniu prawidłowo skonfigurowanego serwera FTP klienci powinni mieć możliwość pobierania plików za pomocą protokołu *rsync*.

Zabezpieczanie serwerów za pomocą SELinux

Firma Red Hat Inc., wprowadzając na rynek pierwszą implementację SELinux w systemach Red Hat, wykonała sprytne posunięcie. Zamiast tworzyć politykę kontroli każdego aspektu systemu Linux, po prostu utworzyła rodzaj polityki, która skupia się na zabezpieczeniu usług najbardziej podatnych na ataki. Następnie firma tak skonfigurowała te usługi, że nawet jeśli nastąpi włamanie, cracker i tak nie będzie mógł naruszyć pozostałej części systemu.

Po otworzeniu portu w zaporze sieciowej, umożliwiającego użytkownikom żądanie usługi, i po uruchomieniu danej usługi do obsługi żądań SELinux może być użyty do ustawienia zapór wokół tej usługi. W wyniku takiego rozwiązania jej proces demona, pliki konfiguracyjne oraz dane nie mają dostępu do zasobów, do których nie uzyskują specjalnego pozwolenia. W ten sposób pozostała część komputera (systemu) jest bezpieczniejsza.

Podczas gdy firma Red Hat kontynuowała wysiłki na rzecz wyeliminowania nieprawidłowości w SELinux, użytkownicy, którzy nie mieli zaufania do tej usługi, zazwyczaj ją po prostu wyłączali. Jeśli jednak SELinux faktycznie może uchronić użytkownika przed niebezpieczeństwem, znacznie lepszym rozwiązaniem jest dokładniejsze poznanie danej funkcji. Jeżeli użytkownik wykryje błąd w SELinux, warto go zgłosić i przyczynić się do usprawnienia tej usługi.

W przypadku udostępniania w systemie Fedora bądź RHEL usług FTP, WWW (HTTPD), DNS, NFS, NIS lub Samby warto rozważyć pozostawienie włączonej usługi SELinux oraz pracę z ustawieniami narzędzia *Konfiguracja poziomu bezpieczeństwa* do konfiguracji tych

usług. Więcej informacji o usłudze SELinux w kontekście dystrybucji Fedora znajduje się na stronie *http://fedora.redhat.com/docs/selinux-faq-fc5*.

Ochrona serwerów sieciowych za pomocą certyfikatów i szyfrowania

W poprzednim podrozdziale Czytelnik dowiedział się, w jaki sposób zabezpieczyć drzwi wejściowe do systemu Linux w celu ochrony przed crackerami. Jednak nawet najlepsze zabezpieczenia okażą się zupełnie nieprzydatne, jeśli użytkownik utraci klucze. Podobnie w świecie komputerów, nawet najlepsze zabezpieczenia są nic niewarte, gdy użytkownik wysyła niezabezpieczone hasła lub inne krytyczne dane przez internet.

Pomysłowy cracker może użyć narzędzia nazywanego *analizatorem protokołu* lub *snifferem sieciowym* i analizować przepływ danych w sieci, aby wydobyć hasła, dane kart kredytowych oraz inne cenne informacje. cracker robi to poprzez włamanie do słabo zabezpieczonego systemu w danej sieci, a następnie uruchamia wymienione oprogramowanie lub po uzyskaniu fizycznego dostępu do danej sieci podłącza własne wyposażenie.

Ochrona przed takimi zagrożeniami jest możliwa dzięki wykorzystaniu szyfrowania. Dwa główne rodzaje szyfrowania używane w chwili obecnej to kryptografia symetryczna oraz kryptografia z użyciem klucza publicznego.

Kryptografia symetryczna

Kryptografia symetryczna jest również nazywana *kryptografią klucza prywatnego* i używa pojedynczego klucza zarówno do szyfrowania, jak i rozszyfrowania wiadomości. Zasadniczo metoda ta jest nieodpowiednia do zabezpieczania danych, które będą używane przez innych, a to ze względu na skomplikowaną wymianę bezpiecznych kluczy. Kryptografia symetryczna jest za to użyteczna do szyfrowania danych tylko na własny użytek.

Eksport technologii szyfrowania

Zanim zostaną przedstawione różne narzędzia szyfrujące, Czytelnik musi przeczytać kilka słów o nietypowej polityce rządu Stanów Zjednoczonych. Przez wiele lat rząd Stanów Zjednoczonych traktował technologię szyfrowania podobnie jak uzbrojenie. W wyniku takiego podejścia każdy, kto chciał eksportować tę technologię, musiał uzyskać zezwolenie w Departamencie Handlu. Ten wymóg dotyczył nie tylko oprogramowania opracowanego w USA, ale również uzyskanego z innych krajów, a następnie ponownie eksportowanego do innego kraju (nawet do tego samego, z którego pozyskano dane oprogramowanie).

Dlatego też, jeżeli użytkownik zainstalował w systemie Linux oprogramowanie zawierające technologię szyfrowania, a następnie zabrał komputer za granicę, łamał prawo federalne. Co więcej, również wysłanie koledze pocztą elektroniczną oprogramowania szyfrującego lub pozwolenie na jego pobranie z serwera naruszało prawo.

W styczniu 2000 roku prawo dotyczące eksportu oprogramowania szyfrującego zostało znacznie złagodzone. Jednak bardzo często Biuro Administracji Eksportu Departamentu Handlu USA wymaga umożliwienia mu zapoznania się z nowymi produktami szyfrującymi przed wydaniem zezwolenia ich eksportu. Firmy działające w USA wciąż nie mogą eksportować technologii szyfrujących do krajów oficjalnie oskarżanych przez USA o terroryzm.

Klasycznym przykładem użycia tego rodzaju kryptografii jest własny skarbiec haseł. Każdy, kto używa internetu przez pewien czas, zdążył już zgromadzić sporą liczbę nazw użytkownika oraz haseł do różnych witryn i zasobów. Osobisty skarbiec haseł pozwala na przechowywanie tych informacji w zaszyfrowanej postaci. Zaletą takiego rozwiązania jest konieczność pamiętania tylko jednego hasła, które odbezpiecza dostęp do wszystkich przechowywanych informacji.

Do niedawna rząd Stanów Zjednoczonych wykorzystywał algorytm szyfrowania symetrycznego o nazwie DES (ang. *Data Encryption Standard*) do zabezpieczania ważnych informacji. Ponieważ nie istnieje bezpośredni sposób złamania danych zaszyfrowanych algorytmem DES, rozszyfrowanie takich danych bez znajomości hasła będzie wymagało niewyobrażalnej ilości czasu oraz olbrzymiej mocy obliczeniowej w celu odgadnięcia użytego hasła. Takie rozwiązanie jest nazywane rozszyfrowaniem metodą *brutalnej siły*.

Ponieważ moc komputerów domowych wzrosła wykładniczo, algorytm DES został odesłany na zasłużoną emeryturę. Po wielu bardzo interesujących poszukiwaniach w jego miejsce rząd Stanów Zjednoczonych zaakceptował algorytm Rijndael, nazywany AES (ang. *Advanced Encryption Standard*). Mimo że algorytm AES również jest celem ataków metodą brutalnej siły, potrzeba zdecydowanie więcej mocy obliczeniowej do jego złamania niż w przypadku algorytmu DES.

Więcej informacji na temat algorytmu AES wraz implementacją algorytmu w postaci polecenia powłoki znajduje się na witrynie *http://aescrypt.sourceforge.net/*.

Kryptografia asymetryczna

Kryptografia z użyciem klucza publicznego nie ma problemów związanych z dystrybucją kluczy i dlatego też stała się zalecaną metodą szyfrowania podczas zabezpieczania komunikacji w internecie. Ta metoda polega na użyciu dwóch kluczy, z których pierwszy służy do szyfrowania wiadomości, natomiast drugi do jej rozszyfrowania. Klucz używany do szyfrowania wiadomości nosi nazwę klucza publicznego, ponieważ jest dostępny dla wszystkich. Z kolei klucz używany do rozszyfrowania wiadomości jest kluczem prywatnym i powinien być utrzymywany w całkowicie bezpiecznym miejscu.

Wyobraźmy sobie, że użytkownik chce wysłać drugiemu bezpieczną wiadomość, używając szyfrowania kluczem publicznym. Poniżej przedstawiono kolejne etapy takiego procesu:

1. Adresat wiadomości musi posiadać parę kluczy — jeden publiczny i jeden prywatny. W zależności od okoliczności takie klucze może wygenerować samodzielnie (za pomocą specjalnego oprogramowania) lub otrzymać z centrum wydającego takie klucze.
2. Nadawca wiadomości musi odnaleźć klucz publiczny adresata (a więc skorzystać z odpowiedniego oprogramowania).
3. Nadawca wiadomości szyfruje ją za pomocą klucza publicznego. Na tym etapie wiadomość może zostać rozszyfrowana tylko za pomocą klucza prywatnego (klucz publiczny nie może być użyty do rozszyfrowania wiadomości).
4. Adresat wiadomości rozszyfrowuje ją za pomocą klucza prywatnego.

Secure Socket Layer (SSL)

Klasyczną implementacją kryptografii z użyciem klucza publicznego jest komunikacja Secure Socket Layer (SSL). To technologia umożliwiająca użytkownikowi bezpieczne przesyłanie sprzedawcy danych zawierających informacje na przykład o karcie kredytowej. Elementami zaszyfrowanej sesji SSL są:

- przeglądarka internetowa, która obsługuje szyfrowanie SSL (Mozilla, Internet Explorer, Opera, Konqueror i inne),
- serwer WWW, który obsługuje szyfrowanie SSL (Apache),
- certyfikat SSL.

W celu zainicjowania sesji SSL przeglądarka internetowa nawiązuje połączenie z serwerem WWW na porcie 443., znanym również jako HTTPS (ang. *HyperText Transport Protocol Secure*). Po ustanowieniu połączenia między dwoma komputerami zachodzą następujące procesy:

1. Serwer wysyła przeglądarce internetowej certyfikat SSL.
2. Przeglądarka internetowa weryfikuje tożsamość serwera poprzez certyfikat SSL.
3. Przeglądarka internetowa generuje symetryczny klucz szyfrowania.
4. Przeglądarka internetowa używa certyfikatu SSL do zaszyfrowania symetrycznego klucza SSL.
5. Przeglądarka internetowa wysyła serwerowi zaszyfrowany klucz.
6. Serwer rozszyfrowuje symetryczny klucz za pomocą odpowiednika klucza prywatnego dla publicznego certyfikatu SSL.

Po wykonaniu powyższych czynności przeglądarka internetowa i serwer mogą zaszyfrować i rozszyfrować komunikację między sobą na podstawie klucza symetrycznego. Między przeglądarką i serwerem zachodzi bezpieczna wymiana danych.

Tworzenie certyfikatów SSL

W celu utworzenia własnego certyfikatu dla bezpiecznej wymiany danych za pomocą protokołu HTTP należy posiadać serwer WWW, który obsługuje SSL. Dostarczany wraz z dystrybucją Fedora oraz innymi systemami Linux serwer WWW Apache (pakiet *httpd*) posiada wbudowaną obsługę SSL. Przedstawiona poniżej procedura tworzenia certyfikatów SSL została przeprowadzona w systemie Fedora zawierającym serwer Apache z pakietu *httpd-2.2.3-5*. Zaprezentowana procedura może różnić się nieco w przypadku serwera Apache dostępnego w innych systemach Linux.

Gdy użytkownik dysponuje gotowym serwerem, warto zapoznać się z ważnymi komponentami znajdującymi się po stronie serwera, a dotyczącymi certyfikatu SSL:

```
# ls -l /etc/httpd/conf
-rw-r--r--  1 root     root    36010 Jul 14 15:45 httpd.conf
lrwxrwxrwx  1 root     root    37 Aug 12 23:45 Makefile ->
                                 ../../../usr/share/ssl/certs/Makefile
```

```
drwx------  2 root     root    4096 Aug 12 23:45 ssl.crl
drwx------  2 root     root    4096 Aug 12 23:45 ssl.crt
drwx------  2 root     root    4096 Jul 14 15:45 ssl.csr
drwx------  2 root     root    4096 Aug 12 23:45 ssl.key
drwx------  2 root     root    4096 Jul 14 15:45 ssl.prm

# ls -l /etc/httpd/conf.d/ssl.conf
-rw-r--r--  1 root     root      11140 Jul 14 15:45 ssl.conf
```

Katalogi */etc/httpd/conf* oraz */etc/httpd/conf.d* zawierają wszystkie komponenty, które będą potrzebne do utworzenia własnego certyfikatu SSL. Każdy komponent został opisany poniżej:

- ***httpd.conf*** — plik konfiguracyjny serwera WWW;
- ***MakeFile*** — skrypt tworzący certyfikat;
- ***ssl.crl*** — katalog listy odwołań certyfikatu;
- ***ssl.crt*** — katalog certyfikatu SSL;
- ***ssl.csr*** — katalog plików żądań certyfikatu (CSR);
- ***ssl.key*** — katalog klucza prywatnego certyfikatu SSL;
- ***ssl.prm*** — parametry certyfikatu SSL;
- ***ssl.conf*** — podstawowy plik konfiguracyjny serwera WWW SSL.

Po zapoznaniu się z podstawowymi komponentami pora zabrać się za narzędzia używane podczas tworzenia certyfikatów SSL:

```
# cd /etc/httpd/conf
# make
This makefile allows you to create:
o public/private key pairs
o SSL certificate signing requests (CSRs)
o self-signed SSL test certificates

To create a key pair, run "make SOMETHING.key".
To create a CSR, run "make SOMETHING.csr".
To create a test certificate, run "make SOMETHING.crt".
To create a key and a test certificate in one file, run "make SOMETHING.pem".

To create a key for use with Apache, run "make genkey".
To create a CSR for use with Apache, run "make certreq".
To create a test certificate for use with Apache, run "make testcert".

Examples:
make server.key
make server.csr
make server.crt
make stunnel.pem
make genkey
make certreq
make testcert
```

Polecenie `make` wywołuje `makefile` do utworzenia certyfikatu SSL. Wydanie polecenia `make` bez żadnych argumentów po prostu wyświetla informacje przedstawione w powyższym przykładzie. Poniżej przedstawiono definicję każdego argumentu, który może zostać użyty wraz z poleceniem `make`:

- `make server.key` — tworzy podstawową parę klucz publiczny – prywatny;
- `make server.csr` — generuje podstawowy plik CSR certyfikatu SSL;
- `make server.crt` — generuje podstawowy, testowy certyfikat SSL;
- `make stunnel.pem` — generuje podstawowy, testowy certyfikat SSL, ale klucz prywatny zostaje umieszczony w tym samym pliku, w którym znajduje się certyfikat SSL;
- `make genkey` — tak samo jak `make server.key`, poza tym, że klucz zostaje umieszczony w katalogu *ssl.key*;
- `make certreq` — podobnie jak `make server.csr`, poza tym, że usługa żądania certyfikatu zostaje umieszczona w katalogu *ssl.csr*;
- `make testcert` — tak samo jak `make server.crt`, poza tym, że certyfikat testowy zostaje umieszczony w katalogu *ssl.crt*.

Używanie certyfikatów pochodzących od firm trzecich

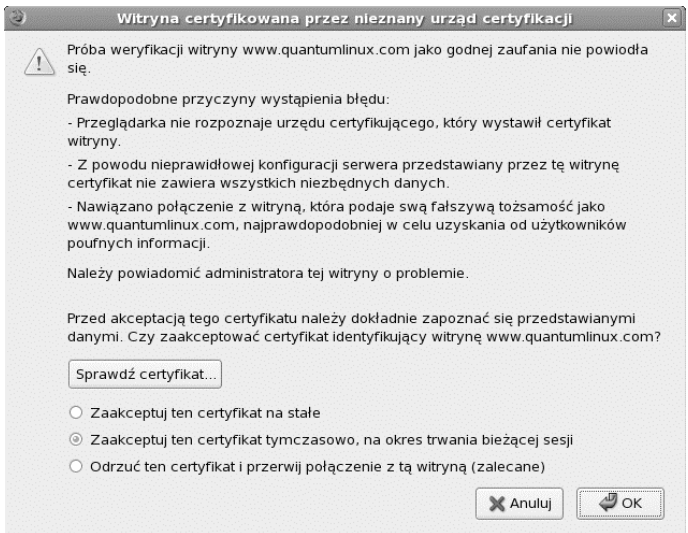
W świecie rzeczywistym jedna osoba rozpoznaje drugą po jej twarzy, głosie i manierach. W internecie nie ma takiej możliwości i trzeba polegać na zaufanych organizacjach, które potwierdzają tożsamość. Aby zagwarantować niezmienność certyfikatu, musi on zostać podpisany przez zaufaną firmę podczas wydawania certyfikatu oraz jego weryfikacji, gdy użytkownik końcowy chce wykorzystać zalety bezpiecznego połączenia z witryną. Poniżej przedstawiono listę zaufanych centrów, które zajmują się wydawaniem certyfikatów i ich potwierdzaniem:

- GlobalSign — *https://www.globalsign.net/*,
- GeoTrust — *https://www.geotrust.com/*,
- VeriSign — *https://www.verisign.com/*,
- RapidSSL — *http://www.freessl.com/*,
- Thawte — *http://www.thawte.com/*,
- EnTrust — *http://www.entrust.com/*,
- ipsCA — *http://www.ipsca.com/*,
- COMODO Group — *http://www.comodogroup.com/*.

Z powodu niestabilnej natury rynku certyfikatów część z wymienionych firm może już nie świadczyć swoich usług, ale również mogły powstać nowe. Bieżąca lista centrów autoryzacji jest dostępna z poziomu przeglądarki Mozilla Firefox po wybraniu opcji *Preferencje* z menu *Edycja*. W oknie dialogowym *Preferencje* należy przejść na zakładkę *Zaawansowane/Bezpieczeństwo/Zarządzanie certyfikatami*. Na ekranie zostanie wyświetlone nowe okno dialogowe *Menedżer certyfikatów*, które na zakładce *Ośrodki certyfikacji* zawiera listę centrów autoryzacji, z których użytkownik posiada certyfikaty.

Każde z wymienionych centrów autoryzacji zaszyło w niemal wszystkich przeglądarkach internetowych dostępnych na świecie elementy kodu kryptograficznego. Pozwalają one przeglądarce internetowej na określenie, czy dany certyfikat SSL jest autentyczny. Bez tego rodzaju weryfikacji crackerzy mogliby stosunkowo łatwo wygenerować własne certyfikaty i oszukiwać użytkowników, którzy byliby przekonani, że podają informacje wrażliwe wiarygodnemu podmiotowi.

Certyfikaty, które nie przechodzą weryfikacji, są nazywane *certyfikatami samodzielnie podpisanymi*. Jeżeli użytkownik porusza się po witrynie, której certyfikat nie został potwierdzony przez zaufaną organizację, przeglądarka internetowa wyświetli komunikat podobny do pokazanego na rysunku 6.2.



Rysunek 6.2. *Okno dialogowe, które ostrzega użytkownika o nieuwierzytelnionym certyfikacie*

Przedstawiony powyżej komunikat nie oznacza, że dana witryna zajmuje się nielegalną, niemoralną lub świńską działalnością. Wiele witryn używa *samodzielnie podpisanych* certyfikatów nie dlatego, że chcą naciągnąć użytkownika, ale z braku powodów do weryfikacji właściciela certyfikatu oraz chęci uniknięcia kosztów związanych z uzyskaniem uwierzytelnionego certyfikatu. Niektóre z powodów używania *samodzielnie podpisanych* certyfikatów to:

- **Witryna internetowa nie pobiera danych wejściowych**. W takich przypadkach użytkownik nie musi się o nic martwić. Nikt nie próbuje ukraść jego informacji, ponieważ na witrynie nie są podawane żadne dane. W większości przypadków takie rozwiązanie ma na celu zapewnienie bezpiecznej komunikacji między witryną i użytkownikiem. Chociaż dane nie muszą być wrażliwe, to witryna i tak zapewnia bezpieczną komunikację, aby uniemożliwić innym przechwytywanie danych.
- **Witryna przeznaczona jest dla małego kręgu odbiorców**. Jeżeli dana witryna internetowa posiada bardzo ograniczony krąg użytkowników, można po prostu poinformować ich o braku uwierzytelnionego certyfikatu. W takim przypadku użytkownicy mogą przeglądać informacje dotyczące certyfikatu i sprawdzać je na przykład poprzez telefon lub bezpośrednio.

- **Testowanie**. Nie ma sensu ponoszenia kosztów certyfikatu SSL podczas testowania nowej witryny lub aplikacji sieciowej. Użycie *samodzielnie podpisanego* certyfikatu jest wówczas wystarczające.

Tworzenie pliku CSR

W celu utworzenia w systemie Fedora Linux certyfikatu SSL uwierzytelnionego przez firmę trzecią należy rozpocząć od pliku CSR (ang. *Certificate Service Request*). Aby utworzyć plik CSR, w serwerze WWW trzeba wykonać następujące czynności:

```
# cd /etc/httpd/conf
# make certreq
umask 77 ; \
/usr/bin/openssl genrsa -des3 1024 > /etc/httpd/conf/ssl.key/server.key
.
.
.
```

Na ekranie zostanie wyświetlone pytanie o hasło zabezpieczające klucz prywatny. Wymienione hasło powinno posiadać co najmniej osiem znaków, nie powinno być słowem znajdującym się w słowniku i nie może zawierać znaków przestankowych. Wpisywane znaki nie będą wyświetlane na ekranie, co uniemożliwia podejrzenie go przez innego użytkownika.

```
Enter pass phrase:
```

Hasło należy podać dwukrotnie w celu jego weryfikacji:

```
Verifying - Enter pass phrase:
```

Po zweryfikowaniu hasła rozpocznie się proces generowania certyfikatu.

Na tym etapie można zacząć podawać do certyfikatu pewne informacje identyfikacyjne, które później będą sprawdzane przez firmę trzecią. Przed tym należy jednak odblokować wygenerowany wcześniej klucz prywatny. Odblokowanie polega na podaniu hasła. Następnie trzeba podawać odpowiedzi na pytania wyświetlane na ekranie. Przykładowa sesja dodawania informacji do certyfikatu została przedstawiona poniżej:

```
Enter pass phrase for /etc/httpd/conf/ssl.key/server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called
a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]: US
State or Province Name (full name) [Berkshire]: Connecticut
Locality Name (eg, city) [Newbury]: Mystic
Organization Name (eg, company) [My Company Ltd]: Acme Marina, Inc.
Organizational Unit Name (eg, section) []: InfoTech
Common Name (eg, your name or your server's hostname) []: www.acmemarina.com
Email Address []: webmaster@acmemarina.com
```

W celu zakończenia procesu użytkownik zostanie zapytany, czy do certyfikatu chce dodać dodatkowe atrybuty. O ile nie istnieje specjalny powód dostarczenia większej ilości informacji, należy po prostu nacisnąć klawisz *Enter*, pozostawiając w ten sposób poniższe pola niewypełnione.

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Podpisywanie pliku CSR

Po utworzeniu pliku CSR należy go wysłać do centrum autoryzacji w celu weryfikacji. Pierwszym krokiem tego procesu jest wybór centrum autoryzacji. Każde z nich posiada własną ofertę, ceny oraz produkty. Warto sprawdzić centra wymienione we wcześniejszej części rozdziału i wybrać najlepsze dla własnych potrzeb. Poniżej przedstawiono obszary, na których występują różnice pomiędzy centrami autoryzacji:

- wiarygodność i stabilność,
- ceny,
- rozpoznanie przez przeglądarki internetowe,
- gwarancje,
- pomoc techniczna,
- jakość certyfikatu.

Po wybraniu centrum autoryzacji użytkownik będzie musiał przejść przez pewne etapy weryfikacji. Każde z centrów posiada opracowane własne metody weryfikacji tożsamości oraz certyfikowania informacji. Niektóre będą wymagały wysłania faksem umowy spółki, podczas gdy inne będą wymagały rozmowy z przedstawicielem firmy. Na pewnym etapie tego procesu użytkownik zostanie poproszony o skopiowanie i wklejenie zawartości utworzonego pliku CSR do formularza sieciowego centrum.

```
# cd /etc/httpd/conf/ssl.csr
# cat server.csr
-----BEGIN CERTIFICATE REQUEST-----
MIIB6jCCAVMCAQAwgakxCzAJBgNVBAYTAlVTMRQwEgYDVQQIEwtDb25uZWN0aWN1
dDEPMA0GA1UEBxMGTXlzdGljMRowGAYDVQQKExFBY211IE1hcmluYSwgSW5jLjER
MA8GA1UECxMISW5mb1RlY2gxGzAZBgNVBAMTEnd3dy5hY211bWFyaW5hLmNvbTEn
MCUGCSqGSIb3DQEJARYYd2VibWFzdGVyQGFjbWVtYXJpbmEuY29tMIGfMA0GCSqG
SIb3DQEBAQUAA4GNADCBiQKBgQDcYH4pjMxKMldyXRmcoz8uBVOvwlNZHyRWw8ZG
u2eCbvgi6w4wXuHwaDuxbuDBmw//Y9DMI2MXg4wDq4xmPi35EsOlOfw4ytZJn1yW
aU6cJVQro46OnXyaqXZOPiRCxUSnGRU+0nsqKGjf7LPpXv29S3QvMIBTYWzCkNnc
gWBwwwIDAQABoAAwDQYJKoZIhvcNAQEEBQADgYEANv6eJOaJZGzopNR5h2YkR9Wg
l8oBl3mgoPH60Sccw3pWsoW4qbOWq7on8dS/++QOCZWZI1gefgaSQMInKZ1II7Fs
YIwYBgpoPTMC4bp0ZZtURCyQWrKIDXQBXw7BlU/3A25nvkRY7vgNL9Nq+7681EJ8
W9AJ3PX4vb2+ynttcBI=
-----END CERTIFICATE REQUEST-----
```

W celu skopiowania i wklejenia pliku CSR do formularza sieciowego centrum można wykorzystać mysz.

Po zakończeniu procesu weryfikacji informacji, opłaceniu centrum i udzieleniu odpowiedzi na wszystkie pytania proces zostanie ukończony. W przeciągu 48 do 72 godzin użytkownik powinien otrzymać wiadomość e-mail wraz z nowym certyfikatem SSL. Wspomniany certyfikat będzie wyglądał podobnie do przedstawionego poniżej:

```
-----BEGIN CERTIFICATE-----
MIIEFjCCA3+gAwIBAgIQMI262Zd6njZgN97tJAVFODANBgkqhkiG9w0BAQQFADCB
ujEfMB0GA1UEChMWVmVyaVNpZ24gVHJ1c3QgTmV0d29yazEXMBUGA1UECxMOVmVy
aVNpZ24sIEluXy4xMzAxBgNVBAsTKlZlcmlTaWduIEludGVybmF0aW9uYWwgU2Vy
dmVyIENBIC0gZ2xhc3MgMzFJMEcG10rY2g0Dd3d3LnZlcmlzaWduLmNvbS9DUFMg
SW5jb3JwLmJ51FJlZi4gTElBQklMSVRZIExURC4oYyk5NyBWZXJpU2lnbjAeFw0w
MzAxMTUwMDAwMDBaFw0wNDAxMTUyMzU5NTlaMIGuMQswCQYDVQQGEwJVUzETMBEG
A1UECBMKV2FzaG1uZ3RvbiThErE371UEBxQLRmVkZXJhbCBXYXkxGzAZBgNVBAoU
Ek1ETSBTZXJ2aWMlcywgSW5jLjEMMAoGA1UECxQDd3d3MTMwMQYDVQQLFCpUZXJt
cyBvZiB1c2UgYXQgd3d3LnZlcmlzaWduLmNvbS9ycGEgKGMpMDAxFDASBgNVBAMU
C2lkbXNlcnYuY29tMIGfMA0GCSqGS1b3DQEBAQUAA4GNADCBiQKBgQDaHSk+uzOf
7jjDFEnqT8UBa1L3yFILXFjhj3XpMXLGWzLmkDmdJjXsa4x7AhEpr1ubuVNhJVI0
FnLDopsx4pyr4n+P8FyS4M5grbcQzy2YnkM2jyqVF/7yOW2pDl30t4eacYYaz4Qg
q9pTxhUzjEG4twvKCAFWfuhEoGu1CMV2qQ1DAQABo4IBJTCCASEwCQYDVR0TBAIw
ADBEBgNVHSAEPTA7MDkGC2CGSAGG+EUBBxcDMCOwKAYIKwYBBQUHAgEWHGh0dHBz
Oi8vd3d3LnZlcmlzaWduLmNvbS9ycGEwCwYDVRRPBAQDAgWgMCgGA1UdJQQhMB8G
CWCGSAGG+EIEM00c0wIYBQUHAwEGCCsGAQUFBwmCMDQGCCsGAQUFBwEBBCgwJjAk
BggrBgEFBQcwAYYYaHR0cDovL29jc2AudmVyaXNpZ24uY29tMEYGA1UdHwQ/MD0w
O6A5oDeGNWh0dHA6Ly9jcmwudmVyaxNpZ24uY29tL0NsYXNzM0ludGVybmF0aW9u
YWxTZXJ2ZXIuY3JsMBkGCmCGSAgG+E+f4Nfc3zYJODA5NzMwMTEyMA0GCSqGSIb3
DQEBBAUAA4GBAJ/PsVttmlDkQai5nLeudLceb1F4isXP17B68wXLkIeRu4Novu13
8lLZXnaR+acHeStR01b3rQPjgv2y1mwjkPmC1WjoeYfdxH7+Mbg/6fomnK9auWAT
WF0iFW/+a8OWRYQJLMA2VQOVhX4znjpGcVNY9AQSHm1UiESJy7vtd1iX
-----END CERTIFICATE-----
```

Otrzymany certyfikat należy skopiować i wkleić do pustego pliku o nazwie *server.crt*, który musi znajdować się w katalogu */etc/httpd//conf/ssl.crt*, a następnie ponownie uruchomić serwer WWW:

```
# service httpd restart
```

Zakładając, że wcześniej witryna internetowa funkcjonowała bez zarzutu, dodanie certyfikatu umożliwi jej przeglądanie za pomocą bezpiecznego połączenia (litera „s" w ciągu tekstowym http adresu witryny). Dlatego też, jeżeli wcześniej witryna była wyświetlana po podaniu adresu *http://www.acmemarine.com*, to po dodaniu certyfikatu można ją wyświetlić w bezpieczny sposób po podaniu adresu *https://www.acmemarina.com*.

Tworzenie samodzielnie podpisanych certyfikatów

Generowanie i wdrożenie samodzielnie podpisanego certyfikatu jest znacznie łatwiejsze niż w przypadku uwierzytelnionych certyfikatów. W celu wygenerowania samodzielnie podpisanego certyfikatu SSL w systemie Fedora należy wykonać następujące czynności:

1. Usunąć aktualnie istniejący certyfikat oraz klucz:

   ```
   # cd /etc/httpd/conf
   # rm ssl.key/server.key ssl.crt/server.crt
   ```

2. Utworzyć własny klucz:

   ```
   # make genkey
   ```

3. Utworzyć samodzielnie podpisany certyfikat za pomocą polecenia:

```
# make testcert
umask 77 ; \
/usr/bin/openssl req -new -key
/etc/httpd/conf/ssl.key/server.key
     -x509 -days 365 -out
/etc/httpd/conf/ssl.key/server.crt
.
.
.
```

Na tym etapie można zacząć podawać do certyfikatu pewne informacje identyfikacyjne, które później będą sprawdzane przez firmę trzecią. Przed tym należy jednak odblokować wygenerowany wcześniej klucz prywatny. Odblokowanie polega na podaniu hasła. Następnie trzeba podawać odpowiedzi na pytania wyświetlane na ekranie, jak przedstawiono na poniższym przykładzie:

```
You are about to be asked to enter information that will be
incorporated into your certificate request.
What you are about to enter is what is called
a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:US
State or Province Name (full name) [Berkshire]: Ohio
Locality Name (eg, city) [Newbury]: Cincinnati
Organization Name (eg, company) [My Company Ltd]: Industrial Press, Inc.
Organizational Unit Name (eg, section) []: IT
Common Name (eg, your name or your server's hostname)
[]: www.industrialpressinc.com
Email Address []: webmaster@industrialpressinc.com
```

Proces generujący umieści wszystkie pliki w odpowiednich miejscach. Użytkownik musi tylko ponownie uruchomić serwer WWW oraz podać w adresie URL ciąg tekstowy *https* zamiast zwykłego *http*. Nie należy zapominać, że serwer WWW wyświetli w przeglądarce internetowej komunikat dotyczący weryfikacji certyfikatu, który można jednak bezpiecznie zignorować.

Ponowne uruchamianie serwera WWW

Czytelnik prawdopodobnie zwrócił uwagę, że serwer WWW w trakcie każdego uruchomienia wymaga podania hasła certyfikatu. Ten krok ma na celu ochronę przed sytuacją, gdy ktoś włamie się do serwera i ukradnie klucz prywatny. Ten etap powinien być zachowany, gdyż daje pewność, że klucz prywatny jest bezpieczny, a cracker nie będzie mógł zrobić z niego użytku. Bez takiego zabezpieczenia cracker mógłby łatwo ukraść klucz prywatny i podawać się za użytkownika.

Jeżeli użytkownik nie chce podawać hasła w trakcie każdego uruchamiania serwera WWW i akceptuje opisane powyżej ryzyko, istnieje możliwość usunięcia szyfrowania klucza prywatnego. W tym celu należy wydać następujące polecenia:

```
# cd /etc/httpd/conf/ssl.key
# /usr/bin/openssl rsa -in server.key -out server.key
```

Rozwiązywanie problemów związanych z certyfikatami

Przedstawione poniżej wskazówki powinny być pomocne, gdy użytkownik napotka problemy związane z certyfikatem SSL:

- Dozwolony jest tylko jeden certyfikat na jeden adres IP. Jeżeli użytkownik chce dodać do serwera więcej niż tylko jedną witrynę zawierającą certyfikat SSL, do interfejsu sieciowego musi zostać dołączony kolejny adres IP.
- Należy się upewnić, że maska uprawnień dla katalogów */etc/httpd/conf/ssl.** oraz ich zawartości została ustawiona na wartość `700` (`rwx------`).
- Należy się upewnić, że port 443. na serwerze WWW nie jest blokowany. Wszystkie żądania *https* są kierowane do portu 443. Jeżeli ten port zostanie zablokowany, obsługa bezpiecznych stron będzie niemożliwa.
- Certyfikat jest ważny przez okres jednego roku. Po upłynięciu roku certyfikat należy odnowić we właściwym centrum autoryzacji. Każde centrum posiada własną procedurę odnawiania certyfikatu, warto sprawdzić odpowiednie informacje na witrynie danego centrum.
- Należy się upewnić, że został zainstalowany pakiet *mod_ssl*. Jeżeli tak nie jest, serwer WWW nie będzie mógł obsłużyć żadnego ruchu związanego z SSL.

Używanie narzędzi bezpieczeństwa systemu Linux uruchamianego z nośnika

Jeżeli istnieje podejrzenie, że komputer lub sieć zostały zaatakowane, użytkownik dysponuje całą gamą narzędzi bezpieczeństwa, których może użyć do przeprowadzenia skanowania antywirusem, analizy lub monitorowania aktywności intruza. Najlepszym sposobem nauczenia się obsługi wielu z tych narzędzi jest wykorzystanie dystrybucji systemu Linux dedykowanych i zbudowanych specjalnie pod kątem bezpieczeństwa i uruchamianych z nośnika.

Zalety odnośnie bezpieczeństwa dystrybucji działających z nośnika

Główną zaletą używania dystrybucji działającej bezpośrednio z nośnika CD lub DVD do sprawdzania bezpieczeństwa systemu jest fakt, że oddziela ona używane narzędzia od sprawdzanego systemu. Innymi słowy, ponieważ narzędzia do odnajdywania problemów na zainstalowanym systemie same mogły zostać naruszone, dystrybucja typu live CD, zawierająca zaufane oprogramowanie, jest gwarancją, że potencjalnie zainfekowany system jest sprawdzany za pomocą czystych narzędzi.

Jeżeli pomimo największych wysiłków (dobre hasła, zapory sieciowe, sprawdzanie plików dzienników zdarzeń) użytkownik jest przekonany, że intruz mógł uzyskać kontrolę nad

systemem, użycie dystrybucji działającej z nośnika jest dobrym rozwiązaniem. Systemy bezpieczeństwa w postaci live DC, takie jak System Rescue CD, INSERT lub BackTrack (wszystkie zostały umieszczone na płytach CD i DVD dołączonych do książki), są doskonałymi narzędziami sprawdzania i naprawy systemu.

Korzystanie z narzędzia INSERT do wykrywania kodu typu rootkit

Jeżeli intruz uzyska dostęp do systemu Linux i spróbuje przejąć kontrolę nad tym systemem (i użyć do czegoś więcej niż tylko włamania i ucieczki), może zainstalować kod nazwany *rootkit*. Wspomniany rootkit jest zestawem oprogramowania, które intruz chce wykorzystać do:

- realizacji swoich planów (na przykład do hostingu fałszywej zawartości WWW za pomocą przejętego serwera),
- ukrycia własnej aktywności.

Kod typu rootkit może stosować różne sposoby ukrywania własnego przeznaczenia. Bardzo często zdarza się, że rootkit zastępuje własnymi wersjami polecenia systemowe. Dlatego też na przykład polecenia `ls` oraz `ps` mogą zostać zmodyfikowane w taki sposób, aby nie wyświetlały odpowiednio pewnej zawartości dodanej do systemu lub ustalonych procesów działających w systemie.

Polecenie `chkrootkit` jest dobrym narzędziem wyszukiwania kodu rootkit. Służy również do sprawdzenia, czy pliki systemowe nie zostały zainfekowane. Narzędzie sprawdzi również potencjalne infekcje w poleceniach sprawdzania dysku (takich jak `du`, `find` i `ls`), poleceniach operacji na procesach (`ps` i `pstree`), poleceniach związanych z logowaniem (`login`, `rlogin`, `slogin`) oraz wielu innych. Poniżej przedstawiono sposób uruchomienia narzędzia `chkrootkit` z dystrybucji INSERT:

1. Do napędu CD włóż płytę CD dołączoną do książki.
2. Gdy na ekranie pojawi się znak zachęty, wpisz `insert` i naciśnij klawisz *Enter*. Podane polecenie spowoduje uruchomienie dystrybucji INSERT.
3. Aby móc sprawdzić system Linux zainstalowany na dysku twardym, należy zamontować partycje reprezentujące zainstalowany system Linux. Używając apletu *mount.app* (wyświetlany w prawym dolnym rogu ekranu), kliknij kursorem na aplecie, przechodząc przez kolejne dostępne urządzenia. Jeżeli Linux był zainstalowany na pierwszej partycji pierwszego dysku twardego, należy wybrać *hda1*. Po wybraniu urządzenia kliknij przycisk *Mount*, który powoduje zamontowanie partycji.
4. Otwórz terminal, klikając pulpit prawym przyciskiem myszy i wybierając *Terminal Session/Aterm — super user*. Na ekranie zostanie wyświetlone okno terminalu.
5. Wydaj polecenie `chkrootkit`, a jego dane wyjściowe zapisz w pliku. Na przykład wydanie poniższego polecenia spowoduje sprawdzenie systemu plików zamontowanego w punkcie */mnt/hda1* i zapisanie danych wyjściowych w pliku *chkroot-output.txt*:

```
# chkrootkit -r /mnt/hda1 > /tmp/chkroot-output.txt
```

6. Kiedy polecenie zakończy swoje działanie, przejrzyj dane wyjściowe. Przykładowo:

```
# less /tmp/chkroot-output.txt
ROOTDIR is '/mnt/hda1/'
Checking 'amd' ... not found
Checking 'basename' ... not infected
.
.
.
```

7. Naciskaj spację i przejdź przez wyświetlone dane wyjściowe, które powinny dostarczyć następujących informacji

   - Jeżeli w systemie został umieszczony kod rootkit, niektóre polecenia zostaną wskazane jako zainfekowane.
   - Jeżeli w dowolnym pliku lub katalogu zostanie odkryte dobrze znane oprogramowanie rootkit, będzie to odnotowane. Polecenie rozpoznaje ponad 60 rodzajów znanego oprogramowania rootkit.
   - Jeżeli zostanie odkryty podejrzanie wyglądający plik, będzie to odnotowane, co umożliwi użytkownikowi sprawdzenie takiego pliku (mimo że nie musi on oznaczać obecności oprogramowania typu rootkit).

Jeżeli w systemie zostanie odkryte oprogramowanie typu rootkit, oznacza to, że ktoś mógł przejąć kontrolę nad komputerem. Bardzo często najlepszym rozwiązaniem w takim przypadku jest po prostu ponowna instalacja systemu. Użytkownik może zastąpić zainfekowane polecenia czystymi, ale w pierwszej kolejności należy upewnić się, że w systemie nie zostały umieszczone tylne drzwi umożliwiające dostanie się do systemu.

Podsumowanie

Zapewnienie bezpieczeństwa systemowi Linux jest zadaniem, które należy przeprowadzać od samego początku i kontynuować w trakcie używania Linuksa. Stosowanie dobrych praktyk w zakresie bezpieczeństwa (takich jak wymienione na liście przedstawionej na początku rozdziału) zwiększa szanse utrzymywana intruzów na odległość.

Idąc jeszcze dalej, użytkownik może pomóc w zabezpieczaniu systemu Linux poprzez używanie szyfrowanych aplikacji sieciowych (na przykład ssh), monitorowanie plików dzienników zdarzeń i stosowanie dobrych technik wyboru haseł. Jeżeli system Linux jest używany w charakterze serwera, należy zachować szczególną ostrożność, zawęzić dostęp do serwera i chronić dane. Do tego celu służą narzędzia takie jak osłony TCP (ograniczające użytkowników, którzy mogą używać serwera) oraz certyfikaty (gwarantujące, że obie strony komunikacji z serwerem WWW są uwierzytelnione).