Ćwiczenie Nr 3
Administracja systemem operacyjnym FreeBSD

Cel ćwiczenia:

Celem niniejszego ćwiczenia jest zapoznanie się studenta z mechanizmami instalacji oprogramowania w systemie FreeBSD przy wykorzystaniu pakietów binarnych oraz przy wykorzystaniu systemu portów, w którym instalowane oprogramowanie jest kompilowane bezpośrednio z postaci źródłowej.

W drugiej części ćwiczenia zainstalowane oprogramowanie student wykorzystuje do uruchomienia typowych usług sieciowych systemów serwerowych. W celu przetestowania działania systemu tworzy bazę testowych użytkowników zapoznając się równocześnie z bardzo podstawowymi mechanizmami zarządzania kontami użytkowników.

W trzeciej części ćwiczenia po połączeniu w pary studenci integrują ze sobą systemy operacyjne w środowiska testowe. Umożliwiają scentralizowane uwierzytelnienie się użytkowników oraz uwspólnienie zasobów sieciowych.

1. Scenariusz ogólny ćwiczenia

- Uruchomienie systemu z gotowego obrazu maszyny z systemem operacyjnym FreeBSD
- Zalogowanie do systemu i zapoznanie się konfiguracją sieciową
- Zapoznanie się podstawowymi komendami do zarządzani zainstalowanymi pakietami
- Instalacja pakietu binarnego
- Instalacja wybranego pakietu z systemu portów
- Instalacja usług sieciowych (ssh, NIS, NFS)
- Uruchomienie usługi ssh
- Utworzenie bazy użytkowników testowych i weryfikacja dostępu do systemu przez usługę ssh
- Zablokowanie dostępu do systemu dla wybranego użytkownika
- Integracja systemu uwierzytelnienia na jednej maszynie
- Uwspólnienie zasobów sieciowych za pomocą usługi NIS

2. Instalacja oprogramowania

Oprócz systemu operacyjnego na serwerze zawsze instalowane jest inne specjalizowane oprogramowanie dopasowane do przeznaczonej dla niego roli. Jeden serwer może obsługiwać wiele różnych usług, a więc istnieje potrzeba współistnienia w jednym systemie plików wielu programów pełniących swoje funkcje. Systemy typu UNIX wypracowały standard POSIX pomagający zarówno programistom, użytkownikom i administratorom w poruszaniu się na różnych wersjach UNIXa. FreeBSD jest klasycznym przykładem takiego systemu.

Oprogramowanie w systemie FreeBSD może być zainstalowane na kilka różnych sposobów:

- poprzez zwykłe skopiowanie plików wykonywanych,
- poprzez ręczną kompilacje programu i skopiowanie do katalogów systemowych,
- poprzez instalacje pakietów binarnych,
- albo przez instalacje z portów, czyli z pakietów źródłowych.

Pierwsze dwa sposoby są wystarczające w przypadku pojedynczej albo szczególnie unikalnej instalacji, natomiast na dłuższą metę są one uciążliwe przy administracji i aktualizacji rozbudowanego systemu. Dlatego w dalszej części skupimy się na dwóch ostatnich sposobach, czyli instalacji pakietów binarnych i źródłowych.

Instalacja oprogramowania za pomocą w FreeBSD została opisana: http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/ports.html

2.1 Dodawanie pakietów za pomocą pkg_add

Programu pkg_add można użyć zarówno do instalowania programów w postaci binarnej (gotowych, skompilowanych) z dysku, jak i z sieci.

Przykład. Pobieranie pakietu i instalowanie jej lokalnie.

```
# ftp -a ftp.pl.FreeBSD.org
Connected to ftp.pl.FreeBSD.org.
220 ftp2.FreeBSD.org FTP server (Version 6.00LS) ready.
331 Guest login ok, send your email address as password.
230-
230-    Welcome.
230-       FreeBSD.
230-      This motd is not real! :)
230-
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /pub/FreeBSD/ports/packages/All/
250 CWD command successful.
ftp> get lsof-4.56.4.tgz
local: lsof-4.56.4.tgz remote: lsof-4.56.4.tgz
200 PORT command successful.
150 Opening BINARY mode data connection for 'lsof-4.56.4.tgz'
(92375 bytes).
100%|***********************************************|92375
00:00
ETA
226 Transfer complete.
92375 bytes received in 5.60 seconds (16.11 KB/s)
ftp> exit
```

Następnie przy wykorzystaniu polecenia pkg_add [nazwa_pakietu.tgz] instalujemy pobrany z serwera ftp program.

```
FreeBSD# pkg_add lsof-4.56.4.tgz
```

Jeśli nie dysponujemy źródłami programów (np. na CDROMie lub DVD FreeBSD), to wygodnie będzie nam wykorzystać komendę pkg_add z opcją -r. Spowoduje to, że program pkg_add samodzielnie określi odpowiednią wersję oprogramowania dla aktualnej wersji naszego systemu operacyjnego. Następnie samodzielnie pobierze odpowiedni plik z sieci oraz go zainstaluje. Oczywiście wymaganym kryterium jest odpowiednia konfiguracja połączenia sieciowego:

```
FreeBSD# pkg_add -r lsof-4.56.4
```

W powyższym przykładzie pkg_add pobierze poprawny pakiet i zainstaluje go bez dalszej ingerencji użytkownika.

W systemie FreeBSD pakiety binarne rozpowszechniane są w formacie .tgz. Można je odnaleźć np. na głównym serwerze projektu ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/packages/, a w Polsce np. pod adresem ftp://ftp.pl.FreeBSD.org/pub/FreeBSD/ports/packages/. Struktura katalogu pakietów podobna jest do drzewa portów /usr/ports. Każda kategoria ma swój własny katalog, ponadto każdy pakiet jest podlinkowany w katalogu All (Wszystkie).

2.2 Usuwanie pakietu.

Aby usunąć uprzednio zainstalowane oprogramowanie należy wykorzystać polecenie pkg_delete.

```
FreeBSD# pkg_delete xchat-1.7.1
```

2.3 Ewidencjonowanie pakietów.

Do ewidencjonowania zainstalowanych w systemie pakietów służ narzędzie pkg_info, które pokazuje zainstalowane pakiety oraz prezentuje ich krótki opis.

Przykład:

pkg_version jest z kolei narzędziem, które podsumowuje wersje wszystkich zainstalowanych paczek. Porównuje je następnie z tymi które znajdują się w drzewie portów.

FreeBSD# pkg_version

cvsup =

docbook =

...

Symbole w drugiej kolumnie wyrażają relatywny wiek zainstalowanej wersji oprogramowania względem wersji odnalezionej w portach. Znaczenie symboli jest następujące:

= Wersja odnaleziona w portach jest identyczna.

< Wersja jest starsza, niż ta odnaleziona w portach.

> Zainstalowana wersja jest nowsza, niż ta, znaleziona w portach.

? Zainstalowana paczka nie może zostać odnaleziona w portach.

* Istnieje wiele wersji tego programu.

Wszystkie informacje o paczkach są zawarte w /var/db/pkg. Lista zainstalowanych plików, a także opis każdej paczki można odnaleźć właśnie w tym katalogu.

2.4 Przykładowe ćwiczenia do części dotyczącej instalacji oprogramowania

1. Instalacja programów wget,lynx,sl z plików binarnych ściągnietych za pomocą ftp (źródło np.: ftp://ftp.pl.freebsd.org/pub/FreeBSD/releases/i386/(...)/packages/).
2. Instalacja programu mutt wykorzystując funkcjonalność zdalnego pobierania pakietów (-r), z określonego serwera podanego w punkcie poprzednim (zmienna środowiskowa PACKAGESITE).
3. Inspekcja stanu, zawartości zainstalowanych pakietów: wyświetlenie streszczenia opisującego do czego dany pakiet służy, wyświetlenie zawartości plików z jakich składa się pakiet itp. (np. komenda pkg_info)
4. Instalacja dystrybucji ports. (analogicznie jak base i kernels trzeba skorzystać z skryptu install.sh: ftp://volt.iem.pw.edu.pl/pub/FreeBSD/releases/i386/(...)/ports)
5. Instalacja programu tree z portów (czyli ze źródeł). Należy odnaleźć pakiet tree w strukturze portów i odpowiednio zainstalować pakiet za pomocą make'a.
6. Instalacja dowolnie wybranego serwera pop3 z portów.
7. Instalacja oraz konfiguracja pakietu sudo z portów.
8. Instalacja wybranego edytora: vim lub pico.
9. Instalacja pakietów do monitorowania sieci: tcpdump oraz nmap.
10. Uaktualnienie struktury portów za pomocą narzędzia portugrade.
11. Wyszukanie w strukturze portów pakietów związanych z monitorowaniem sieci. (Co najmniej 8 pakietów.)

3. Administracja usługami sieciowymi

Celem drugiej części ćwiczenia jest zapoznanie się z kilkoma powszechnie używanymi usługami w systemach UNIX i MS Windows. Ćwiczenie obejmuje instalację, konfigurację, przetestowanie oraz zarządzanie kilkoma typowymi usługami sieciowymi, takimi jak: INETD, NIS, NFS, SMB, DNS.

W trakcie ćwiczenia powinny być nabyte następujące umiejętności:

- Jak zarządzać serwerem (demonem) inetd.
- Jak skonfigurować sieciowy system plików.
- Jak zainstalować serwer informacji sieciowych do współdzielenia kont użytkowników (NIS).
- Jak uruchomić serwer nazw (DNS).
- Jak skonfigurować system do udostępniania zasobów dla klientów Windows® za pomocą protokołu Samba.

3.1 Superserwer inetd

Demon inetd jest specjalnym programem, który nasłuchuje połączeń na określonych gniazdach sieciowych. Gdy na jednym z gniazd zaistnieje połączenie, decyduje on, jakiej usłudze to gniazdo odpowiada i wywołuje program, który obsłuży żądanie. Po zakończeniu programu, inetd kontynuuje nasłuchiwania gniazda (poza niektórymi wypadkami, opisanymi poniżej). Ogólnie, inetd umożliwia używanie jednego demona do wywoływania wielu innych, zmniejszając wymagane obciążenie systemu.

Opcje dostępne dla inetd:

-d Włącza debuggowanie.

Podczas uruchamiania, inetd odczytuje swoją konfigurację z pliku konfiguracyjnego, którym domyślnie jest /etc/inetd.conf. Musi tam być wpis dla każdego pola pliku konfiguracyjnego, z poszczególnymi wpisami dla danego pola; wpisy są oddzielane znakiem tabulacji lub spacji. Komentarze są zaznaczane przez ``#'' na początku linii. Musi istnieć wpis dla każdego pola. Pola pliku konfiguracyjnego są następujące:

- nazwa usługi (service name)
- rodzaj gniazda (socket type)
- protokół (protocol)
- określenie, czy usługa ma "zwlekać" (wait/nowait[.max])
- użytkownik[.grupa] (user[.group])
- program serwera (server program)
- argumenty programu serwera (server program arguments)

Aby podać usługę opartą o Sun-RPC , wpis powinien zawierać te pola.

- nazwa usługi/wersja (service name/version)
- rodzaj gniazda (socket type)
- rpc/protokół (rpc/protocol)

- zwłoka (wait/nowait[.max])
- użytkownik[.grupa] (user[.group])
- program serwera (server program)
- argumenty programu serwera (server program arguments)

Wpis nazwa-usługi jest nazwą prawidłowej usługi, zdefiniowanej w pliku /etc/services. Dla usług "wewnętrznych" (internal) (opisanych niżej), nazwa usługi musi być oficjalną nazwą usługi (to znaczy pierwszym wpisem w /etc/services). Podczas podawania usługi opartej o Sun-RPC , pole to jest prawidłową nazwą usługi RPC, zdefiniowaną w pliku /etc/rpc. Część na prawo od "/" jest numerem wersji RPC. Może to być zwyczajny argument numeryczny, lub zakres wersji. Zakres jest obramowany od niższej wersji do wyższej - "rusers/1-3".

Wpis rodzaj gniazda powinien być jednym z "stream", "dgram", "raw", "rdm", lub "seqpacket", zależnie od tego, czy gniazdo jest strumieniowe (stream), datagramowe (datagram), lub typu raw, reliably delivered message, czy też sequenced packet.

Pole protokół musi być prawidłowym protokołem, zdefiniowanym w /etc/protocols. Przykładami mogą być "tcp" lub "udp". Usługi oparte o RPC są określane przez "rpc/tcp" lub "rpc/udp."

Wpis użytkownik powinien zawierać nazwę użytkownika użytkownika, pod którym powinien uruchamiać się serwer. Umożliwia to serwerom posiadanie mniejszych praw niż prawa roota. Opcjonalnie, po dodaniu kropki do nazwy użytkownika, można podać w tym polu nazwę grupy. Umożliwia to serwerom pracę w innym (podstawowym) id grupy niż ten, podany w pliku z hasłami. Jeśli grupa jest podana, a użytkownik nie jest rootem, to uzupełniające grupy związane z użytkownikiem wciąż będą ustawione.

Wpis program serwera powinien zawierać ścieżkę programu, który ma być wywoływany przez inetd po otrzymaniu żądania na gnieździe. Jeśli inetd udostępnia tę usługę wewnętrznie, to wpis ten powinien być wpisem "internal".

Wpis argumenty programu serwera powinien wyglądać tak jak zwykłe argumenty, poczynając od argv[0], który jest nazwą programu. Jeśli usługa jest udostępniana wewnętrznie, to wpis powinien przyjąć nazwę "internal."

3.2 Dodatkowe informacje

Ćwiczenie oparte jest na Handbooku, Rozdział 25. http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/network-servers.html

3.3 Organizacja zajęć

Z uwagi na to, że usługi sieciowe wymagają co najmniej dwóch komputerów pracujących we wzajemnej interakcji w trakcie zajęć należy dobrać się w czteroosobowe grupy, złożone z dwóch dwuosobowych grup (w

sumie dwa komputery). Grupy dwuosobowe będą oceniane oddzielnie. Nie przewiduje się możliwości posiadania dwóch komputerów wirtualnych przez jedną grupę dwuosobową.

3.4 Scenariusz szczegółowy drugiej części

1) SSH

a) uruchomienie i konfiguracja serwera ssh, ograniczenie logowania do konkretnych użytkowników,

b) logowanie za pomocą kluczy z maszyną volt, (ssh-keygen , rsa) c) kopiowanie pliku przy pomocy scp i sftp na maszynę volt (kopiowanie pojedynczych plików, oraz rekursywnie drzew katalogów)

Literatura: http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/openssh.html

2) NFS

a) uruchomienie serwera nfs b) uruchomienie klienta nfs

c) montowanie cifs/smbfs z volta (analiza skryptu hmount)

Literatura: http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/network-nfs.html

3) FTP

a) uruchomienie serwera ftp - w trybie "stand alone" (czyli jako samodzielny serwer)

b) uruchomienie serwera ftp - w trybie poprzez usługę inet c) przesłanie plików

Literatura: http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/network-inetd.html
http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/network-ftp.html

4) DHCP

a) statyczna konfiguracja sieci w skryptach rc.conf

b) uruchomienie serwera DHCP na wskazanym przez prowadzacego interfejsie sieciowym c) pobranie konfiguracji DHCP z uruchomionego przez inna grupę serwera (rola klienta) na wskazany przez prowadzącego interfejs

Literatura: http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/network-dhcp.html

5) DNS

a) bind: konfiguracja pliku resolve.conf

b) uruchomienie serwera DNS (bind) dla wskazanej przez prowadzącego sieci c) skonfigurowanie klienta DNS przez inną grupę i przetestowanie uruchomionego serwera DNS

Literatura: http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/network-dns.html

5) NIS

a) konfiguracja serwera NIS (/var/yp)

b) konfiguracja klienta NIS oraz skanowanie zawartości plików informacyjnychn np passwd za pomocą komendy ypcat

Literatura: http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/network-nis.html

3.5 Przykładowe zadania

1. Uruchomienie dowolnego serwera pop3 przy wykorzystaniu serwisu inetd.
2. Ograniczenie oraz przetestowanie dostępu do serwisu inetd przy pomocy /etc/hosts.allow.
3. Uruchomienie oraz konfiguracja serwera NFS.
4. Udostępnienie do wybranego 1 hosta katalogu /home w trybie zapisu odczytu.
5. Udostępnienie w trybie tylko do odczytu katalogu /usr
6. Konfiguracja automatycznego montowania systemów plików udostępnionych w punkcie 3.
7. Uruchomienie oraz konfiguracja serwera DNS (bind) dla co najmniej czterech komputerów z sieci 192.168.0.
8. Uruchomienie oraz konfiguracja serwera NIS. Udostępnienie kont użytkowników do drugiej stacji. Odpowiednia konfiguracja drugiej stacji, tak aby korzystała z serwera głównego. (To zadanie musi być wykonane dwa razy. Dwa komputery z każdej czteroosobowej grupy muszą za każdym razem pełnić inna rolę: raz serwera, raz klienta.)
9. Instalacja oraz konfiguracja serwera Samba do pracy w grupie roboczej (WORKGROUP) na obydwu komputerach. Udostępnienie katalogów domowych użytkowników, oraz katalogu /opt. (Uwaga katalog /opt może nie istnieć. W takim przypadku należy go utworzyć.)

4. Zabezpieczenie systemu – kopia bezpieczeństwa

Celem trzeciej części zajęć jest zapoznanie się z problemem wiążącymi się z zabezpieczeniem danych działającego systemu. Na zajęciach powinny zostać uruchomione wybrane usługi oraz system kopii bezpieczeństwa.

4.1 Kopie bezpieczeństwa

Podstawowa filozofia backupu FreeBSD została zwięźle i rzeczowo opisana w dokumentacji FreeBSD: http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/disks.html . Kilka podsumowujących informacji można znaleźć w prezentacji Pana Pawła Rutkowskiego: http://konferencja2005.meetbsd.org/files/meetbsd2005-pawel_rutkowski.pdf.

W trakcie trzeciej części zajęć studenci powinni zapoznać się z dwoma wybranymi technikami wykonywania kopii bezpieczeństwa:

- kopiowanie wybranych fragmentów na inną partycję,
- kopiowanie wybranych fragmentów na inną maszyną (często technikę tę nazywa się 'backupem skrośnym').

4.2 Przykładowe problemy

- Proszę zastanowić się przed jakimi problemami każda z technik jest w stanie zabezpieczyć system.
- Obydwa zadania backupu chcielibyśmy aby spełniały następujące wymagania:
    - Chcielibyśmy aby procedura backupu była zautomatyzowana w skrypcie napisanym w dowolnym języku (najłatwiej w języku powłoki).
    - Skrypt ma zostać zainstalowany w cronie, tak aby uruchamiał się co godzinę. (/etc/crontab)
    - Za każdym razem gdy skrypt jest uruchomiony ma utworzyć jeden plik z backupem zawierający kopię wybranych katalogów. Nazwa musi jednoznacznie identyfikować maszynę, której backup zawiera plik oraz czas w którym skrypt został uruchomiony. Niech będzie to nazwa zgodna z szablonem: hostname_yyyy-mm-dd_hh:MM , gdzie mm - miesiąc, dd - dzień, hh - godzina, MM - minuta. (należy wykorzystać komendy: hostname , date +%Y-%m-%d_%H:%M)
    - Skrypt, przy każdym uruchomieniu, powinien automatycznie kasować pliki starsze niż 3 okresy backupu (np. 3 godziny). (wykorzystać komendę find, która wyświetli pliki starsze o pewien okres od aktualnego find . -name 'hostname*' -amin +180 , proszę zapoznać się z opcjami programu find.)
- Dane w kopii zapasowej muszą być skompresowane (np. tar -cvzf ... lub tar + gzip)
- W przypadku backupu na maszynie sieciowej (komputer kolegów) należy wykorzystać OpenSSH (scp) z wykorzystaniem automatycznej autoryzacji za pomocą kluczy RSA. (Logowanie na maszynę kolegów bez podawania hasła. shh-keygen -t rsa)

Przykład skryptu, który może posłużyć jako szablon na zajęcia:

```
# Jun 2010, robert@iem.pw.edu.pl
# Pelne sciezki katalogow lub plikow ktore maja byc zachowane.
# To sa tylko przyklady. Prosze zwrocic uwage na cudzyslowia.

dirs="
/home
/bin
/usr/local/share/doc"

# Utworz nazwe pliku z backupem:
backup="`hostname`_`date (tutaj odpoowiedni format)`.tgz"
```

```
# Upewnij się czy katalog dla backupów istnieje:
mkdir /pelna_sciezka/do_miejsca/przechowywania_backupow

# Upewnij sie ze dostep do backupow ma tylko root
chown -R root:wheel /ta/sama/sciezka/co/poprzednio
chmod -R 755 /ta/sama/sciezka/co/poprzednio

# Usun stare backupy (lub ewentualnie plik o takiej samej nazwie)
for f in `tutaj odpowiedni find`; do
  % rm $f
  % proponuje do testow na poczatek wyswietlac pliki, ktore chcemy usuwac:
  echo "Usuwam: rm $f"
done
rm /ta/sama/sciezka/co/poprzednio/$backup

# 'tarujemy'
tar -c --file /ta/sama/sciezka/co/poprzednio/$backup -p -P --gzip -v $dirs

# Jeżeli wykonujemy backup na maszynie kolegów, to tutaj najlepiej jest skopiować
# go w odpowiednie miejsce na ich komputerze.
```

3. Literatura

[1] http://manpages.ubuntu.com/manpages/natty/pl/man8/inetd.8.html

[2] http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/ports.html

[3] http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/network-servers.html

[4] http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/network-dns.html