Міністерство транспорту та зв'язку України

Державний департамент із питань зв'язку та інформатизації
Одеська національна академія зв'язку ім. О. С. Попова

Кафедра мережі зв'язку

АДМІНІСТРУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ

Модуль 1. Адміністрування комп'ютерних систем

для студентів

з напряму вищої освіти 0924 – Телекомунікації
Спеціальність: 6.092400 Інформаційні мережі зв'язку

Частина 2

Адміністрування комп'ютерних систем

Одеса 2008

УДК 681.3.06

Укладачі: Нікітченко В. В., Яворська О. М.

У даному керівництві розглядаються теоретичні основи та практичні аспекти реалізації базових серверів мережі. Досліджуються можливості реалізації статичної маршрутизації програмним шляхом, засобами системи Free BSD.

Вивчаються важливі для функціонування мережі та спрощення її настроювання та супроводу сервери DHCP та DNS, які дозволяють автоматизувати процес видачі IP-адрес, зіставляти їх значення символьним іменам машин. Розглянуто програмну реалізацію функції брандмауера на базі IPFW для Free BSD. Також розгядається реалізація деяких сервісних служб для мережі, а саме FTP на базі сервера Pro FTPD та електронної пошти на базі Sendmail.

СХВАЛЕНО
на засіданні кафедри мережі зв'язку
і рекомендовано до друку
Протокол № 5 від 7.12.2007 р.

ЗАТВЕРДЖЕНО
Методичною радою
ОНАЗ ім. О. С. Попова, деканат ІМ
Протокол № від «» 2008 р.

ЗМІСТ

І Передмова

Загальна характеристика дисципліни:
Кількість кредитів ECTS – 3
модулів – 1
значеннєвих модулів – 5
загальна кількість годин – 108
у т. ч.:
- аудиторних – 52 год.
- лекції – 26 год.
- практичні заняття – 14 год.
- лабораторні заняття – 12 год.
- самостійна робота студента й індивідуальна робота з викладачами – 56 год.

семестри: 4.3 (аудиторні заняття).

Види контролю:

- поточний контроль знань за лекційним матеріалом;
- іспит;
- контроль виконання комплексних завдань на практичних заняттях.

ІІ Мета даної дисципліни:

Надання студентам спеціалізованих знань з інсталяції, настроюванні й обслуговуванні комп'ютерних систем і мереж. Вивчення принципів функціонування й методики налаштування базових мережних служб. Конфігурування клієнтського й серверного програмного забезпечення. Розгляд методів взаємодії локальних мереж і Інтернет та методів захисту локальних сегментів від несанкціонованого зовнішнього проникнення. Здобуття навичок роботи з установлення інформаційної мережі й сервісного обслуговування апаратно-програмних засобів.

ІІІ Зміст дисципліни

Модуль 1. Функціонування базових служб комп'ютерних мереж і серверів інформаційних служб Інтернет (3 кредити)

Вхідні вимоги для вивчення модуля.

№	Зміст знань	Шифр
1	Знання протоколів мережного й транпортного рівнів моделі OSI. Знання базових понять про методи маршрутизації. Знання апаратної архітектури комп'ютерних систем. Знання принципів функціонування мережі Інтернет й основних її додатків. Знання протоколів http, ftp, pop3, imap, smtp. Базові знання про захист інформації	
	Зміст умінь	
2	Уміння користуватися стандартами й нормативними документами ISO і IEEE, використати апаратно-програмні засоби на рівні кваліфікованого користувача.	

Структура модуля

№	Значеннєвий модуль	Лекції (годин)	Заняття		Самост. робота (годин)	Індив. робота (годин)
			практ. (годин)	лабор. (годин)		
Модуль 1: Функціонування базових служб комп'ютерних мереж і серверів інформаційних служб Інтернет (3 кредити = 108 годин)						
1	Загальні процедури адміністрування комп'ютерних систем і мереж	2	2	2	10	1
2	Служби настроювання параметрів стека протоколів TCP/IP і передачі цих параметрів клієнтам	8	4	4	11	1
3	Розподіл мережних ресурсів	4	2	2	10	1
4	Служби забезпечення безпеки комп'ютерних мереж	6	4	2	10	1
5	Інформаційні служби Інтернет	6	2	2	10	1
	Усього	26	14	12	51	5

Перелік практичних занять

	Тема	Години
1	Резервне копіювання й відновлення даних	2
2	Принцип організації DNS у MS Windows Server 2003. Функції DNS-клієнта	2
3	Сервери дистанційного доступу й VPN. Адміністрування клієнтів вилученого доступу	2
4	Установка локальних і мережних принтерів і керування ними	2
5	Керування користувачами й групами користувачів. Планування організаційних одиниць і групової стратегії	2
6	Керування файловими ресурсами й Group Policy. Відновлення розподілених ресурсів і прав доступу до них	2
7	Планування способів мережної безпеки	2
	Усього:	14

Перелік лабораторних занять

1	Резервне копіювання й відновлення файлових систем і даних засобами утиліти Windows Server 2003 Backup	2
2	Налаштування й обслуговування серверу доменних імен у Windows Server 2003	2
3	Налаштування й обслуговування серверу DHCP у Windows Server 2003	2
4	Налаштування розподіленої файлової системи Dfs	2
5	Налаштування й обслуговування брандмауера й проксі-серверу ISA	2
6	Налаштування й обслуговування web-серверу Apache	2
	Усього:	12

Вихідні знання й уміння

№	Знання	Шифр
1	Знання процедур адміністрування комп'ютерних систем і резервного копіювання даних. Знання головних мережних служб, розуміння взаємодії цих служб. Знання структури мережних баз даних і організація мережних ресурсів. Знання інтернет-аутентифікації, VPN і трансляція мережних адрес NAT. Знання інформаційних служб Інтернет/Інтранет	
	Уміння	
2	Уміння набудовувати сервери DHCP, DNS, WINS. Уміння надавати мережний доступ до принтерів і файлових ресурсів. Уміння поновлювати конфігурацію системи після апаратних або програмних збоев. Уміння набудовувати мережні екрани й проксі-сервера, здійснювати конфігурування серверів web, ftp і smtp, надавати до них доступ клієнтам	

IV Методи навчання

Лекції з використанням технічних засобів, практичні й лабораторні роботи; використання методичного забезпечення з дисципліни й електронної бази даних кафедри.

V Методи оцінювання

- поточний контроль знань з лекційного матеріалу;
- іспит;
- контроль виконання індивідуальних завдань на практичних заняттях;
- захист протоколів виконання лабораторних робіт.

Оцінювання проводиться за шкалою ESTS, національною шкалою й шкалою ОНАЗ (100 балів).

VI Література

1. Рассел Ч., Кроуфорд Ш., Джеренд Дж. MS Windows 2003. Справочник администратора. – М.: СП ЭКОМ, 2004.
2. Дэвис Дж., Ли Т. MS Windows 2003. Протоколы и службы TCP/IP. – М.: СП ЭКОМ, 2005.
3. Немет Э., Сннайдер Г., Сибас С., Хейн Т. UNIX: Руководство системного администратора. – М.: ПИТЕР, 2003.
4. Вахалия Ю. UNIX изнутри. – М.: ПИТЕР, 2003.
5. Остерлох Х. TCP/IP. Семейство протоколов передачи данных в сетях компьютеров. – М.: Диасофт, 2002.
6. Мюллер С. Модернизация и ремонт ПК. – М.: Вильямс, 2002.

Лабораторна робота № 1

ПОБУДОВА ПРОГРАМНОГО МАРШРУТИЗАТОРА

1 Мета роботи

1.1 Вивчення практичних аспектів маршрутизації в комп'ютерних мережах.

1.2 Набуття навичок побудови програмного маршрутизатора на базі ОС FreeBSD.

2 Ключові положення

Щоб деяка машина могла знайти в мережі іншу, повинен бути механізм описування того, як добратися від однієї машини до іншої. Такий механізм називається маршрутизацією. Маршрутизація відбувається на мережному рівні. Коли приходить пакет, призначений для іншого вузла, його цільова IP-адреса відшукується в таблиці маршрутизації ядра. При збігу (хоча б частковому) з будь-яким маршрутом у таблиці пакет спрямовується за IP-адресою наступного шлюзу, зв'язаного з даним маршрутом.

Для ілюстрації різних аспектів маршрутизації розглянемо вивід команди netstat:

```
% netstat -r
Routing tables

Destination     Gateway         Flags    Refs    Use    Netif Expire

default     outside-gw    UGSc      37     418  ppp0
localhost   localhost     UH     0     181   lo0
test0       0:e0:b5:36:cf:4f  UHLW    5    63288  fxp0    77
10.20.30.255    link#1        UHLW    1     2421
example.com  link#1        UC     0    0
host1       0:e0:a8:37:8:1e   UHLW    3     4601   lo0
host2       0:e0:a8:37:8:1e   UHLW    0    5   lo0 =>
host2.example.com link#1       UC      0    0
224     link#1        UC     0    0
```

У перших двох рядках задаються маршрут за замовчуванням і маршрут на localhost. Інтерфейс (колонка Netif), що зазначена в таблиці маршрутів для використання з localhost і який названий lo0, має також другу назву, пристрій loopback. Це мається на увазі збереження всього трафіка для зазначеної адреси призначення всередині, без посилання його по мережі, тому що він однаково буде напрвлений туди, де був створений.

Наступними адресами, що виділяються, є адреси, що починаються з 00:e0:.... Це апаратні адреси Ethernet, або MAC-адреси. FreeBSD буде автоматично розпізнавати будь-який хост (у нашому прикладі це test0) у локальній мережі Ethernet і додасть маршрут для цього хоста, що вказує безпосередньо на інтерфейс Ethernet, fxp0. Із цим типом маршруту також зв'язаний параметр таймаута (колонка Expire), використовуваний у випадку невдалої спроби почути цей хост протягом деякого періоду ча-

су. Якщо таке відбувається, то маршрут до цього вузла буде автоматично вилучений. Такі хости підтримуються за допомогою механізму, відомого як RIP (Routing Information Protocol), що обчислює маршрути до хостів локальної мережі за допомогою визначення найкоротшої відстані.

FreeBSD додасть також усі маршрути до підмереж для локальних підмереж (10.20.30.255 є широкомовною адресою для підмережі 10.20.30, а ім'я example.com є ім'ям домену, що зв'язаний із цією підмережею). Призначення link#1 відповідає першому адаптеру Ethernet у машині. Зверніть увагу на відсутність додаткового інтерфейсу для цих рядків.

Рядок host1 відноситься до нашої машини, що відомий за адресою Ethernet. Тому що ми є що посилає хостом, необхідно використати loopback-інтерфейс (lo0) замість того, щоб здійснювати посилку в інтерфейс Ethernet.

Два рядки host2 є прикладом того, що відбувається при використанні аліасів у команді ifconfig. Символ => після інтерфейсу lo0 указує на те, що ми використовуємо не просто інтерфейс loopback (тому що це адреса, яка означає локальний хост), але до того ж це аліас. Такі маршрути з'являються тільки на комп'ютері, що підтримує аліаси; для всіх інших машин у локальній мережі для таких маршрутів будуть показані просто рядки link#1.

Останній рядок (підмережа призначення 224) має відношення до багатоадресної посилки.

Різні атрибути кожного маршруту перераховуються в колонці Flags. У табл. 1 наводяться деякі із цих прапорців та їхнє значення.

Таблиця 1 - Значення прапорців установлених маршрутів

Прапорці	**Значення**
U (Up)	Маршрут актуальний
H (Host)	Адресою призначення є окремий хост
G (Gateway)	Посилати все для цієї адреси призначення на зазначену вилучену систему, що буде сама визначати подальший шлях проходження інформації
S (Static)	Маршрут був настроєний вручну, а не автоматично сгенерований системою
C (Clone)	Новий маршрут сгенерований на основі зазначеного для машин, до яких ми підключені. Такий тип маршруту звичайно використається для локальних мереж
W (Was Cloned)	Указує на те, що маршрут був автоматично сконфігурований на основі маршруту в локальній мережі (Clone)
L (Link)	Маршрут включає посилання на апаратну адресу Ethernet

Маршрути за замовчуванням

Коли локальній системі потрібно установити з'єднання з вилученим хостом, вона звертається до таблиці маршрутів для того, щоб визначити, чи існує такий маршрут. Якщо вилучений хост попадає в підмережу, для якої відомий спосіб її досягнення (маршрути типу Cloned), то система визначає можливість підключитися до неї по цьому інтерфейсу.

Якщо всі відомі маршрути не підходять, у системи є остання можливість: маршрут "default". Це маршрут з особливим типом мережного шлюзу (звичайно єдиним, присутнім у системі), і в полі прапорців він завжди позначений як C. Для комп'ютерів

у локальній мережі цей мережний шлюз указує на машину, що має пряме підключення до зовнішнього середовища.

Якщо ви настроюєте маршрут за замовчуванням на машині, яка сама є мережним шлюзом у зовнішнє середовище, то маршрутом за замовчуванням буде мережний шлюз вашого провайдера Інтернет (ISP).

Ви можете легко задати використовуваний за замовчуванням шлюз за допомогою файла /etc/rc.conf. Для прикладу додамо такий рядок у файл /etc/rc.conf:

defaultrouter="10.20.30.1"

Це також можливо зробити й безпосередньо з командного рядка за допомогою команди route:

 # route add default 10.20.30.1

Комп'ютери з подвійним підключенням

Є ще один тип підключення, коли машина перебуває у двох різних мережах. Технічно будь-яка машина, що працює як мережний шлюз, вважається хостом з подвійним підключенням. В одному випадку в машині є два адаптери Ethernet, це кожен має адресу у розділених підмережах. Як альтернативу можна розглянути варіант із одним Ethernet-адаптером і використанням аліасів у команді ifconfig. У першому випадку використовуються дві фізично розділені мережі, в останньому є один фізичний сегмент мережі, але дві логічно розділені підмережі.

У кожному разі таблиці маршрутизації настроюються так, що для кожної підмережі ця машина визначена як шлюз (вхідний маршрут) в іншу підмережу. Така конфігурація часто використається, якщо потрібно реалізувати систему безпеки на основі фільтрації пакетів або функцій брандмауера в одному або обох напрямах.

Настроювання статичних маршрутів

Мережний маршрутизатор є звичайною системою, що пересилає пакети з одного інтерфейсу на інший. Для включення цієї можливості у FreeBSD необхідно змінити значення системної змінної gateway_enable у файлі rc.conf:

gateway_enable=YES # Set to YES if this host will be a gateway

Цей параметр змінить значення sysctl-змінної net.inet.ip.forwarding в 1. Якщо вам тимчасово потрібно виключити маршрутизацію, ви можете скинути її значення в 0.

Розглянемо приклад настроювання найпростішого маршрутизатора. Припустимо, що в нас є наступна мережа, зображена на рис. 1. У цьому сценарії Router – це наш комп'ютер, що виступає як маршрутизатор у мережу Інтернет. Його маршрут за замовчуванням настроєний на 10.0.0.1, що дозволяє йому з'єднуватися із зовнішнім середовищем. Будемо припускати, що Router уже правильно настроєний знає всі необхідні маршрути.

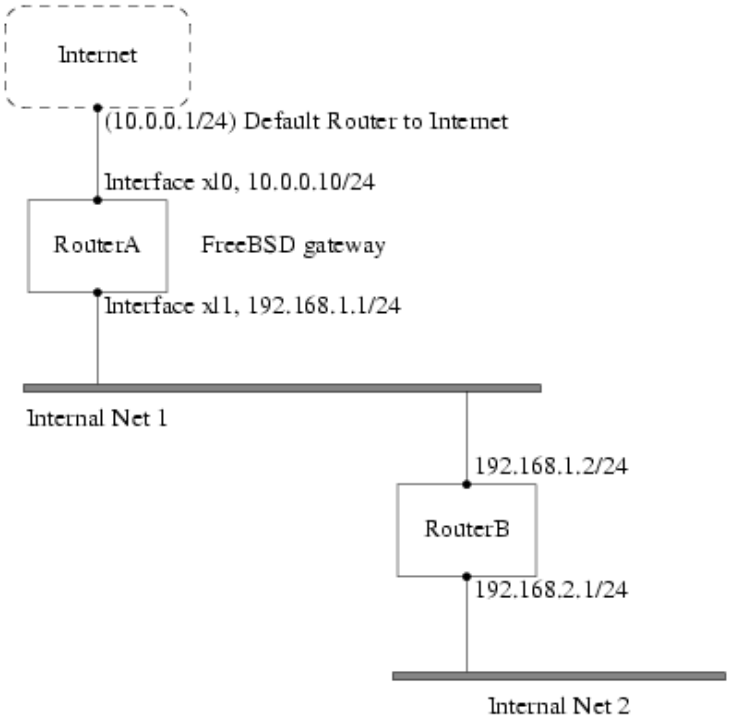

Рисунок 1 – Приклад тестової мережі

Якщо ми подивимося на таблицю маршрутизації Router, то побачимо приблизно наступне:

% netstat -nr
Routing tables

Internet:

Destination	*Gateway*	*Flags*	*Refs*	*Use*	*Netif*	*Expire*
default	*10.0.0.1*	*UGS*	*0*	*49378*	*xl0*	
127.0.0.1	*127.0.0.1*	*UH*	*0*	*6*	*lo0*	
10.0.0/24	*link#1*	*UC*	*0*	*0*	*xl0*	
192.168.1/24	*link#2*	*UC*	*0*	*0*	*xl1*	

З поточною таблицею маршрутизації Router не зможе досягти Internal Net 2. Один зі способів обходу цієї проблеми – додавання маршруту вручну. Наступна команда додає внутрішню_мережа_2 до таблиці маршрутизації Router з 192.168.1.2 як наступний вузол:

 # route add -net 192.168.2.0/24 192.168.1.2

Тепер Router зможе досягти будь-якого хоста в мережі 192.168.2.0/24.

Постійна конфігурація

Попередній приклад добре підходить для налаштування статичного маршруту в працюючій системі. Однак проблема полягає в тім, що маршрутна інформація не збе-

режеться після перезавантаження FreeBSD. Спосіб збереження доданого маршруту полягає в додаванні його у файл /etc/rc.conf:

 # Додавання статичного маршруту в Internal Net 2
static_routes="internalnet2"
route_internalnet2="-net 192.168.2.0/24 192.168.1.2"

У змінній static_routes перебувають рядки, розділені пробілами. Кожний рядок означає ім'я маршруту. У цьому випадку в static_routes є тільки один рядок, це internalnet2. Потім ми додали змінну route_internalnet2, куди поміщені всі параметри, які необхідно передати команді route.

Ми можемо додати в static_routes більш ніж один рядок. Це дозволить створити кілька статичних маршрутів. У наступному прикладі показане додавання маршрутів для мереж 192.168.0.0/24 і 192.168.1.0/24 (цей маршрутизатор не показаний на рис. 1):

static_routes="net1 net2"
route_net1="-net 192.168.0.0/24 192.168.0.1"
route_net2="-net 192.168.1.0/24 192.168.1.1"

Команда ifconfig

Команда ifconfig використовується для підключення й відключення мережного інтерфейсу, завдання його IP-адреси й маски підмережі, а також інших опцій і параметрів. Вона звичайно виконується під час початкового завантаження, але може застосовуватися й для внесення змін у працюючу систему.

У більшості випадків команда ifconfig має такий формат:

ifconfig інтерфейс адреса опції ... up

Параметр *інтерфейс* означає апаратний інтерфейс, до якого застосовується команда. Як правило це дво-трисимвольне ім'я пристрою, за яким слідує число. Приклади імен: fxp0, wb1, rl0.

Параметр *адреса* задає IP-адресу інтерфейсу. Як правило, він дається в традиційній для Інтернет точковій нотації, але можна вказувати й ім'я машини.

Ключове слово up активізує інтерфейс, а ключове слово down відключає його.

За необхідності призначити одному фізичному мережному адаптеру декількох IP-адрес можна скористатися так званими аліасами (alias). Якщо аліас перебуває в тій же самій мережі, що й уже настроєна на інтерфейсі адреса, то додайте в командному рядку для ifconfig приблизно наступне:

 # ifconfig fxp0 alias 192.0.2.2 netmask 0xffffffff

У противному випадку просто задайте мережну адресу й маску звичайним чином:

 # ifconfig ed0 alias 172.16.141.5 netmask 0xffffff00

Зверніть увагу на те, що при перезавантаженні аліаси будуть втрачені. Щоб цього не відбувалося, досить написати найпростіший скрипт, зберігши його як /etc/start_if.fxp0 (тут fxp0 - ім'я вашого мережного пристрою):

```
ifconfig fxp0 195.133.64.133 netmask 255.255.255.192
for i in 134 135 136 137 138
do
ifconfig fxp0 195.133.64.$i netmask 255.255.255.192  alias
done
```

Результат роботи буде наступним:

```
router# ifconfig fxp0
fxp0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 195.133.64.133 netmask 0xffffffc0 broadcast 195.133.64.191
        inet 195.133.64.134 netmask 0xffffffc0 broadcast 195.133.64.191
        inet 195.133.64.135 netmask 0xffffffc0 broadcast 195.133.64.191
        inet 195.133.64.136 netmask 0xffffffc0 broadcast 195.133.64.191
        inet 195.133.64.137 netmask 0xffffffc0 broadcast 195.133.64.191
        inet 195.133.64.138 netmask 0xffffffc0 broadcast 195.133.64.191
        ether 00:80:48:ed:af:df
```

3 Контрольні питання

3.1 Поясніть призначення й процес виконання маршрутизації.
3.2 Дайте визначення маршруту за замовчуванням.
3.3 Для чого потрібний маршрут на localhost?
3.4 Охарактеризуйте маршрут, що володіє прапорцями UHLW.
3.5 Як задати шлюз за замовчуванням?
3.6 Яку змінну необхідно установити для роботи системи як маршрутизатор?
3.7 Чим різні дії команди route add із ключами -net і -host?
3.8 Поясниіть призначення аліасов у команді ifconfig.

4 Домашнє завдання

4.1 Вивчіть ключові положення.
4.2 Підготуйте відповіді на ключові питання.

5 Лабораторне завдання

5.1 Визначити IP-адресу системи й зміст таблиці маршрутів.
5.2 Організуйте чотири логічні підмережі. Для цього створіть чотири віртуальних мережних інтерфейси, використовуючи аліасинг у команді ifconfig. Значення IP-адрес визначаються виразом:

lan1: 192.168.1.n/24
lan2: 192.168.2.n/24
lan3: 192.168.3.n/24
lan4: 192.168.4.n/24

де n - номер робочого місця студента.

5.3 Перегляньте таблицю маршрутів, поясніть зміни.
5.4 Переконайтеся, що задана конфігурація не буде збережена при перезавантаженні. Рестартуйте систему, перегляньте таблицю маршрутизації й мережні інтерфейси.
5.5 Настройте виконання аліасінга автоматично при старті системи, і переконайтеся в його працездатності.
5.6 У файл rc.conf додайте маршрути для підмереж:
192.168.5.0/24 шлюз 192.168.7.2
192.168.6.0/24 шлюз 192.168.8.2
Переконайтеся в працездатності всієї конфігурації.

Примітка. Для виконання команд ifconfig і route необхідно мати права суперкористувача. Оскільки студент таких прав не має, ці команди слід запускати через sudo (наприклад, sudo ifconfig).

6 Вимоги до змісту протоколу

6.1 Назва лабораторної роботи.
6.2 Мета роботи.
6.3 Результати виконання домашнього завдання.
6.4 Короткий опис виконаної роботи.
6.5 Висновки про виконану роботу.
6.6 Дата, підпис студента, віза викладача.

Лабораторна робота № 2

НАСТРОЮВАННЯ Й СУПРОВІД СЕРВЕРА DNS BIND

1 Мета роботи

1.1 Конфігурування служби доменних імен на базі серверу BIND v.9.x.
1.2 Одержання практичних навичок настроювання й супроводи DNS.

2 Ключові положення

За замовчуванням в FreeBSD використовується одна з версій програми BIND (Berkeley Internet Name Domain), що є найбільпоширеною реалізацією протоколу DNS для UNIX-систем. FreeBSD у цей час поставляється із сервером BIND9, що надає розширені настроювання безпеки, нову схему розташування файлів конфігурації й автоматичних настроювань для chroot. Документацію й програмне забезпечення BIND також можна завантажити з web-вузла www.isc.org. Новини й інструментальні засоби DNS доступні на вузлі DNS Resource Directory за адресою www.dns.net/dnsrd.

Програмне забезпечення серверу BIND складається з демона серверу імен, декількох зразків файлів конфігурації й бібліотек програми перетворення імен. Демон серверу імен BIND називається named. Щоб ваш комп'ютер працював як сервер імен, потрібно просто запустити цей демон у відповідній конфігурації. Демон буде очікувати надходження запитів на перетворення імен і повідомляти відповідну IP-адресу для запитуваного хост-імені. За допомогою утиліти rndc, що поставляється разом з BIND, можна запускати, зупиняти, перезапускати й перевіряти стан серверу під час його настроювання. Команда *stop* утиліти rndc зупиняє демон named, а команда *start* знову запускає його, при цьому зчитується файл конфігурації named.conf. Під час виклику утиліти rndc з опцією help виводиться список усіх доступних команд.

Сервери імен звичайно використовуються у двох видах: авторитетний сервер імен і кешуючий сервер імен. Авторитетний сервер імен потрібний, коли:

– потрібно надавати інформацію про DNS іншому середовищі, відповідаючи на запити авторизовано;
– зареєстрований домен, такий, як example.org і в цьому домені потрібно поставити імена машин у відповідність із їхніми адресами IP;
– блоку адрес IP потрібні зворотні записи DNS (перетворення IP-адрес в імена хостів);
– резервний (slave) сервер імен повинен відповідати на запити.

Кешируючий сервер імен потрібний, коли локальний сервер DNS може кешувати інформацію й відповідати на запити швидше, ніж це відбувається при прямому опитуванні зовнішнього серверу імен.

Наприклад, коли будь-хто запитує інформацію про www.freebsd.org, то звичайно резолвер звертається до серверу імен вашого провайдера, надсилає запит і очікує відповіді. З локальним кешуючим сервером DNS запит у зовнішнє середовище буде виконаний всього один раз. Кожний додатковий запит не буде посилатися за межі локальної мережі, тому що інформація вже є в кеші.

Настроювання BIND

Настроювання серверуа BIND вимагає присутності декількох файлів, посилання на які вказуються у файлі конфігурації named.conf. Повний набір файлів виглядає таким чином:

Файл настроювання (конфігураційний файл)

Визначає загальні аспекти роботи named і вказує джерела бази даних DNS, використовуваної цим сервером. Джерелами можуть слугувати локальні файли на вінчестері або вилучені сервери. Файл настроювання звичайно називається named.conf.

Структура команд настроювання файла named.conf схожа на структуру програм мови С. Оператор закінчується крапкою з комою (;), константи беруться у лапки (“”), а споріднені елементи групуються за допомогою фігурних дужок ({}). Коментар може обмежуватися парами символів /* і */, //, або #.

Основні команди настроювання named.conf наведені в табл. 1.

Таблиця 1 – Команди настроювання named.conf

Команда	**Призначення**
acl	Визначає список керування доступом, що складається з IP-адрес
include	Включає вміст зовнішнього файла у файл настроювання
key	Визначає ключі перевірки дійсності
logging	Визначає склад журналу повідомлень
options	Визначає глобальні параметри настроювання
server	Визначає властивості вилученого серверу
zone	Визначає зону

Вміст файла named.conf визначає роль серверу: основний сервер зони, підлеглий сервер зони або спеціальний кешируючий сервер.

Файл кореневих покажчиків

Вказує на сервери кореневих зон. Файл кореневих зон може називатися named.ca, db.cashe, named.root або root.ca. У розглянутій реалізації BIND використовується ім'я named.root. У процесі настроювання серверу імен цей файл НЕ РЕДАГУЄТЬСЯ.

Кільцевий файл

Використається для локального дозволу кільцевої адреси. Називається localhost.rev.

Файл зони прямого перетворення

Файл зони, що містить відображення імен вузлів в IP-адрес. Саме цей файл містить основний масив інформації про зону.

Файл зони зворотного перетворення

Файл зони, що містить відображення IP-адрес в імена вузлів.

Файли прямої й зворотної зон звичайно мають наочні імена, що дозволяють зрозуміти дані якої зони зберігаються у файлі.

На файлах зон зупинимося більш докладно. Ці файли мають схожу будову й складаються із записів тих самих типів, а саме стандартних записів ресурсів, відомих як RR-запису.

RR-запис має наступний формат:

[ім'я] [ttl] IN тип дані

ім'я

Ім'я доменного об'єкта, з яким зв'язаний запис. Це може бути окремий вузол або цілий домен. Рядок у поле імені інтерпретується відносно поточного домену, за винятком імені, що закінчується крапкою.

ttl

Час існування (time-to-live) визначає тривалість зберігання запису в кеші вилученої системи. Як правило, це поле залишають пустим, і в такому випадку використовується час існування за замовчуванням, установлене для всієї зони в цілому.

IN

Вказує, що RR-запис має клас Internet.

тип

Вказує тип RR-запису. Основні типи записів наведені в табл. 2.

дані

Таблиця 2 – Типи записів ресурсів

Тип запису	**Назва запису**	**Призначення**
SOA	Початок компетенції	Відзначає початок даних зони й визначає параметри, що впливають на зону в цілому
NS	Сервер імен	Указує сервер імен домена
A	Адреса	Забезпечує перетворення імені вузла на адресу
PTR	Покажчик	Забезпечує перетворення адреси вузла в ім'я
MX	Поштовий ретранслятор	Вказує, куди слід доставляти пошту, призначену певному доменному імені
CNAME	Канонічне ім'я	Визначає псевдонім для імені вузла
TXT	Текст	Зберігає довільні текстові рядки

Інформація, що властива даному типу RR-запису. Наприклад, у випадку адресної (A) запису поле даних містить IP-адресу.

Також у BIND існує чотири директиви, що спрощують створення файла зон або визначальних параметрів RR-записів.

Директива $TTL задає значення часу існування за замовчуванням для RR-записів, що не містять наявної вказівки параметра ttl.

Директива $ORIGIN установлює поточну зону, тобто доменне ім'я, яким доповнюються всі відносні доменні імена. Відносним вважається будь-яке доменне ім'я, що не закінчується крапкою. За замовчуванням $ORIGIN приймає значення доменного імені, зазначеного в операторі zone.

Директива $INCLUDE включає вміст зовнішнього файла як фрагмент файла зони.

Директива $GENERATE використається для створення серій RR-записів.

Утиліта nslookup

Nslookup – це інструмент налагодження, що входить до складу пакета BIND. Програма дозволяє користувачу прямо звертатися до серверу імен з запитами й одержувати будь-яку інформацію, збережену в розподіленій базі даних DNS. Команда

nslookup допомагає визначити факт працездатності серверу й коректність його настроювання, а також запросити інформацію, якою володіють вилучені сервери.

У наступному лістингі nslookup використовується для визначення IP-адреси певного вузла:

```
#nslookup  cmp1.orion.edu
Server:  router.orion.edu
Address:  192.168.0.135

Name:    cmp1.orion.edu
Address:   192.168.0.150
```

Приклад настроювання серверу імен

У даному прикладі ми займемося настроюванням уявлюваного домену example.org для того, щоб ви побачили, як усі компоненти серверу DNS працюють разом.

Файли конфігурації демону named розташовані в каталозі /etc/namedb і, лише у випадку, коли вам потрібний просто резолвер, вимагають модифікації. Для створення основної зони для локального хоста перейдіть у каталог /etc/namedb і виконайте команду

```
# sh make-localhost
```

У каталозі master повинні з'явитися файли localhost.rev для локальної адресної зони й localhost-v6.rev для для конфігурації IPv6. Посилання на ці файли вже втримуються у файлі конфігурації named.conf. Зміст файлу localhost.rev наведено нижче:

```
; From: @(#)localhost.rev   5.1 (Berkeley) 6/30/90
;  $FreeBSD: src/etc/namedb/PROTO.localhost.rev,v 1.6 2000/01/10 15:31:40 peter Exp $
;
; This file is automatically edited by the 'make-localhost' script in
; the /etc/namedb directory.
;

$TTL    3600

@       IN      SOA     test.example.org. root. test.example.org.  (
                        20070219      ; Serial
                        3600    ; Refresh
                        900     ; Retry
                        3600000       ; Expire
                        3600 ) ; Minimum
  IN      NS      test.example.org.
1IN       PTR     localhost. example.org.
```

Як видно з лістинга, сервер настроюється на машині з ім'ям test, для домену example.org.

Після цього можемо приступитися до створення файлів зон. Розглянемо вміст файла зони прямого перетворення zone.example.org.

```
@       86400 IN      SOA   ns.example.org. root.example.org. (
        2003040501
        28800
        7200
        604800
        86400 )
        IN      NS    192.168.1.1
ns              IN    A     192.168.1.1
test    IN      A     192.168.1.1
localhost       IN    A     127.0.0.1
cmp1    IN      A     192.168.1.2
cmp2      IN    A     192.168.1.3
cmp3    IN      A     192.168.1.4
cmp4    IN      A     192.168.1.5
cmp5    IN      A     192.168.1.6
cmp6      IN    A     192.168.1.7
cmp7      IN    A     192.168.1.8
cmp8      IN    A     192.168.1.9
```

root@example.org – електронна адреса людини, відповідального за супровід сервера DNS.

2003040501 – порядковий номер; 28800 – періодичність відновлень (с); 7200 – повторення спроби дозволу (с); 604800 – старіння (с); 86400 – TTL кеша (с).

IN NS 192.168.1.1 - вказівка на IP-адресу серверу імен.

cmp1 IN A 192.168.1.2 – запис ресурсів, використовуваний для перетворення імені хоста в IP-адресу.

Розглянемо вміст файла зони зворотного перетворення 1.168.192.rev.

```
$TTL    86400
@       86400 IN      SOA   ns.example.org.      root.example.org. (
                                                  2003090501
                                                  28800
                                                  7200
                                                  604800
                                                  86400)
        IN      NS    ns.example.org.
1       IN      PTR   ns.example.org.
1       IN      PTR   test.example.org.
2       IN      PTR   cmp1.example.org.
3       IN      PTR   cmp2.example.org.
4       IN      PTR   cmp3.example.org.
5       IN      PTR   cmp4.example.org.
```

```
6       IN      PTR   cmp5.example.org.
7       IN      PTR   cmp6.example.org.
8       IN      PTR   cmp7.example.org.
9       IN      PTR   cmp8.example.org.
```

Структури файлів зон мають багато спільного між собою. Основною відмінністю є використання покажчика PTR для зворотного дозволу IP-адреси комп'ютера в його символічне ім'я. Зверніть увагу, що у файлі зони зворотного перетворення вказується лише останній октет IP-адреси, однозначно ідентифікуючий хост. Так, запис

```
3       IN      PTR   cmp2.example.org.
```

позначає, що імені cmp2.example.org зіставляється адреса 192.168.1.3.

Тепер пропишемо шляхи до файлів бази даних DNS у конфігураційному файлі named.conf:

```
// $FreeBSD: src/etc/namedb/named.conf,v 1.21.2.1 2005/09/10 08:27:27 dougb Exp $
//
// Refer to the named.conf(5) and named(8) man pages, and the documentation
// in /usr/share/doc/bind9 for more details.

options {
  directory       "/etc/namedb";
  pid-file        "/var/run/named/pid";
  dump-file       "/var/dump/named_dump.db";
  statistics-file "/var/stats/named.stats";

// If named is being used only as a local resolver, this is a safe default.
// For named to be accessible to the network, comment this option, specify
// the proper IP address, or delete this option.
  listen-on       { 192.168.1.1; };

// If you have IPv6 enabled on this system, uncomment this option for
// use as a local resolver.  To give access to the network, specify
// an IPv6 address, or the keyword "any".
//        listen-on-v6   { ::1; };

// In addition to the "forwarders" clause, you can force your name
// server to never initiate queries of its own, but always ask its
// forwarders only, by enabling the following line:
//
//     forward only;

// If you've got a DNS server around at your upstream provider, enter
// its IP address here, and enable the line below.  This will make you
// benefit from its cache, thus reduce overall DNS traffic in the Internet.
/*
  forwarders {
        195.5.27.1;
  };
```

```
*/
 /*
  * If there is a firewall between you and nameservers you want
  * to talk to, you might need to uncomment the query-source
  * directive below.  Previous versions of BIND always asked
  * questions using port 53, but BIND versions 8 and later
  * use a pseudo-random unprivileged UDP port by default.
  */
 // query-source address * port 53;
};

zone "." {
 type hint;
 file "named.root";
};

zone "0.0.127.IN-ADDR.ARPA" {
 type master;
 file "master/localhost.rev";
};

// RFC 3152
zone "1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA" {
 type master;
 file "master/localhost-v6.rev";
};

zone "example.org" {
 type master;
 file "master/zone.example.org";

};

zone "0.168.192.in-addr.arpa" {
 type master;
 file "master/1.168.192.rev";

};
```

Аналіз вмісту named.conf показує, що всі файли, які забезпечують роботу DNS, зберігаються в каталозі /etc/namedb. Файли дампа зберігаються в /var/dump/named_dump.db.

У якості форвардера використовується DNS-сервер 195.5.27.1. Також слід згадати файл кореневих покажчиків named.root, які ми в процесі настроювання DNS залишили без змін.

Тепер ми можемо запустити сервер імен. Для одноразового запуску демона в цій конфігурації використайте команду

 # /etc/rc.d/named forcestart

Щоб демон named запускався під час завантаження, помістіть у /etc/rc.conf наступний рядок:

named_enable="YES"

Працездатність серверу імен можна перевірити утилітою nslookup.

3 Контрольні питання

3.1 Поясніть призначення служби доменних імен.
3.2 З яких компонентів складається сервер BIND?
3.3 Яка утиліта використається для керування сервером BIND?
3.4 Поясніть відмінність основного серверу імен (master) від кешуючого.
3.5 Які файли необхідні для настроювання авторитетного серверу імен?
3.6Перелічте використовувані директиви для файлів зон. Пояснить їхнє призначення.
3.7 Перелічте основні типи RR-записів, поясніть кожну з них.
3.8 Яким чином стартувати демон named під час завантаження системи?

4 Домашнє завдання

4.1 Вивчіть ключові положення.
4.2 Підготуйте відповіді на ключові питання.

5 Лабораторне завдання

5.1 Настроювання сервера імен виробляється для домену *прізвище_студента.org*.
5.2 Створіть файли зони для локального хоста.
5.3 Керуючись прикладом з каталогу /home/studentXX/named напишіть свій файл для прямої зони, після чого з відповідним ім'ям скопіюйте його в каталог /etc/named/master/.
5.4 Керуючись прикладом з каталогу /home/studentXX/named напишіть свій файл для зворотної зони, після чого з відповідним ім'ям скопіюйте його в каталог /etc/named/master/.
5.5 Пропишіть посилання на відповідні файли в named.conf. Також відредагуйте інші необхідні параметри для функціонування серверу імен. У якості форвардера прийміть машину з IP-адресою 192.168.0.145.
5.6 Стартуйте сервер BIND за допомогою утиліти rndc.
5.7 За допомогою nslookup переконаєтеся в працездатності серверу.

6 Вимоги до змісту протоколу

6.1 Назва лабораторної роботи.
6.2 Мета роботи.
6.3 Результати виконання домашнього завдання.
6.4 Короткий опис пророблені роботи.
6.5 Висновки про виконану роботу.
6.6 Дата, підпис студента, віза викладача.

Лабораторна робота № 3

ПОБУДОВА ПАКЕТНОГО ФІЛЬТРА ЗАСОБАМИ IPFW

1 Мета роботи

1.1 Вивчення можливостей програмних брандмауерів.
1.2 Виконання настроювання захисного екрана на основі IPFW.

2 Ключові положення

Firewall (або брандмауер) – це система, що управляє проходженням пакетів даних через систему на основі заданих адміністратором правил та інформації, що втримується в заголовках пакетів. Звичайне використання брандмауера - заборона проходження небажаних пакетів, наприклад, відключення абонента, що не оплатив послуги зв'язку, або закриття частини сервісів локальної мережі від зовнішнього оточення. У системі FreeBSD firewall – це набагато більше, ніж просто пакетний фільтр – це потужний інструмент керування мережею, що дозволяє, наприклад, підраховувати трафік за будь-якими розумними правилами, що ґрунтується на даних заголовків пакетів протоколів стека TCP/IP, обробляти пакети зовнішніми програмами, приховувати за одним комп'ютером цілу мережу та ін.

Програмне забезпечення *IPFW*, що поставляє з FreeBSD, є системою фільтрації й обліку пакетів, що перебуває в ядрі й поставленою користувальницькою утилітою настроювання *ipfw*. Разом вони дозволяють визначати й переглядати правила, використовувані ядром при маршрутизації. IPFW складається із двох частин. Межмережний екран здійснює фільтрацію пакетів. Частина, що займається обліком IP пакетів, відслідковує використання маршрутизатора на основі правил подібних тим, що використовуються в міжмережнім екрані. Це дозволяє адміністратору визначати, наприклад, обсяг трафіка, отриманого маршрутизатором від конкретного комп'ютера, або обсяг пересилаємого www-трафіка.

Завдяки тому, як реалізовано IPFW, ви можете використати його й на клієнтських комп'ютерах для фільтрації вхідних і вихідних з'єднань. Це випадок більш загального використання IPFW, і в цій ситуації використовуються ті ж команди й техніка.

Опції ядра й файл rc.conf

Оскільки основна частина системи IPFW перебуває в ядрі, воно повинне бути зібране з підтримкою декількох опцій, специфічних для IPFW. Зупинимося на них більш докладно:

-IPFIREWALL – включення підтримки firewall в ядро. Якщо потрібні тільки найпростіші функції, начебто заборони або дозволу проходження пакетів, то цю опцію можна не включати – FreeBSD завантажить відповідний модуль автоматично, якщо в /etc/rc.conf дозволене використання firewall, але підтримка можливостей буде мінімальною;

-IPFIREWALL_VERBOSE – при включенні даної опції firewall може записувати всі "події" у плог-файл. Корисно для аналізу роботи системи й спостереження за спробами злому;

-IPFIREWALL_VERBOSE_LIMIT=<число> - обмеження числа повідомлень, що попадають у журнал. За відсутності ліміту хакер або просто збійний комп'ютер у

мережі здатні в лічені хвилини згенерувати стільки пакетів, що плог-файл firewall'a займе весь наявний дисковий простір;

-IPFIREWALL_DEFAULT_TO_ACCEPT - за замовчуванням при завантаженні системи firewall блокує весь мережний трафік. Дана опція змушує firewall завантажуватися в режимі "всім можна все". Слід використати у випадку гострої потреби. Прикладом може бути використання FreeBSD як операційної системи для бездискових робочих станцій – у цьому випадку за відсутності зазначеної опції в ядрі FreeBSD весь трафік буде заблокований до того, як FreeBSD зможе підмонтувати мережні диски й завантажити з них дійсний набір правил для firewall;

-IPDIVERT – підтримка можливості передати пакет на обробку зовнішній програмі. Програма обробки обов'язково повинна бути запущена на цій же машині;

-IPFIREWALL_FORWARD – можливість перенапрямку пакетів іншому адресату. Використається для маршрутизації з фільтрацією пакетів;

-DUMMYNET – система обмеження пропускної здатності певних з'єднань, заснована на затримці проходження пакетів через роутер.

Для того щоб firewall стартував при завантаженні системи, у файл /etc/rc.conf необхідно помістити рядки:

firewall_enable=”YES”
firewall_script=”шлях_до_файла_правил_ipfw”

Настроювання правил ipfw

Перевірка пакета виробляється за впорядкованим списом правил, які задаються адміністратором. Кожному правилу привласнюється номер (або вручну адміністратором, або автоматично), і правила перевіряються строго в порядку зростання номерів. Кілька правил можуть мати той самий номер - у цьому випадку вони перевіряються в тому порядку, в якому вони були занесені в список. Кожне правило містить умову й дію. Ось загальний вид правила firewall:

ipfw [-N] команда [номер] дія [log] протокол адреси [інтерфейс] [параметри]

При використанні цієї команди доступний тільки один прапорець - N, що пропонує дозволяти імена сервісів і адреси при відображенні.

Розглянемо окремо кожну складову правила ipfw. Згадуються лише основні, найбільш часто використовувані опції й параметри. Для одержання повного списку звернутися до документації з ipfw.

Команда
add – додавання правила до списку фільтрації/обліку;
delete – видалення правила зі списку фільтрації/обліку;
flush – очистити ланцюжок;
show або *list* – показати вміст ланцюжка правил;
zero або *resetlog* – обнулити лічильники правил.

Попередні версії IPFW використали окремі записи для фільтрації й обліку пакетів. Сучасні версії враховують пакети для кожного правила.

Дія

allow (також *pass*, *permit*, і *accept*) – дозволити проходження пакета, що відповідає правилу. Припинити пошук;

deny (або *drop*) – знищити пакет, не сповіщаючи відправника;

reject – відкинути пакет і відправити на адресу джерела ICMP-пакет, що повідомляє про недосяжність хоста або порту;

unreach код – знищити пакет і направити його відправнику ICMP-повідомлення із зазначеним *кодом*;

count – обновити лічильник пакета, але не застосовувати стосовно нього правила allow/deny. Пошук продовжиться з наступного правила в ланцюжку;

divert порт – передати пакет на зазначений порт хоста й не виконувати подальшу перевірку. Використовується, наприклад, для виконання трансляції адрес за допомогою natd;

fwd ipaddr [,port] – змінити *next-hop* (наступний відхід назад) для відповідних пакетів на IP-адресу *ipaddr*, що може бути IP-адресою в точково-десятковій нотації або ім'ям хоста;

check-state – перевіряє пакет за динамічними правилами. Якщо відповідність виявлена, пошук припиняється, у противному випадку виконується перехід до наступного правила. Якщо *check-state* правило не виявлене, динамічні правила перевіряються за першим правилом з опцією *keep-state*;

log [logamount number] – повідомлення про проходження пакета, що відповідає даному правилу з ключовим словом log, буде записано в журнал syslogd засобом LOG_SECURITY;

Протоколи

all - відповідає всім IP–пакетам;

icmp - відповідає ICMP–пакетам;

tcp - відповідає TCP–пакетам;

udp - відповідає UDP–пакетам.

Повний список доступних для вказівки протоколів ви можете подивитися у файлі /etc/protocols.

Адреси

Поле адреси є складовим:

джерело адреса/маска [порт] мета адреси/ маска [порт]

Може бути зазначена IP-адреса, доменне ім'я комп'ютера (типу www.example.org), або ціла підмережа у форматі IP:MASK або IP/LAN, наприклад 192.168.0. 0:255.255. 255.0 або 192.168.0. 0/24. Є так само два спеціальних слова – *any*, що означає будь-яку адресу (аналогічно 0.0. 0.0/0) і *me*, що означає будь-яку з адрес, що належить локальній системі. Номер порту вказується після адреси через пропуск. Кілька номерів портів можна вказати через кому. Перед адресою або номером порту може стояти слово "*not*", що інвертує значення адреси або порту, тобто from not 192.168.0.0/24 означає "всі пакети, що прийшли не з мережі 192.168.0.0/24".

Формат адреси джерела й одержувача збігаються.

Специфікація інтерфейсу

Допускаються наступні комбінації специфікації інтерфейсів:

in – тільки для вхідних пакетів;

out – тільки для вихідних пакетів;

via ifx – пакет повинен бути пропущений через інтерфейс ifx;

*via if** – пакет повинен бути пропущений через інтерфейс iX, де X - будь-який номер пристрою;

via ipno – пакет повинен бути пропущений через інтерфейс, IP-address якого ipno.

Вказівка ключового слова *via* завжди змушує перевіряти інтерфейс. Якщо зазначені ключові слова *recv* або *xmit*, то це теж змушує завжди перевірити інтерфейс. Інтерфейс recv може бути перевірений або на вхідні, або на вихідні пакети, у той час як інтерфейс xmit може бути перевірений тільки на вихідні пакети. Так out потрібний щоразу, коли використовується xmit. Поєднаняння via разом з xmit або з recv неприпустимо.

Параметри

frag – спрацьовує, якщо пакет не є першим пакетом дейтаграмми;

ipoptions spec – спрацьовує, якщо заголовок IP містить перерахований через кому список параметрів, зазначених у spec. Підтримувані параметри IP: ssrr (strict source route), lsrr (loose source route), rr (record packet route), і ts (time stamp). Дія окремих параметрів може бути змінена шляхом вказівки префікса !;

established – спрацьовує, якщо пакет є частиною вже встановленого TCP з'єднання (тобто якщо установлені біти RST або ACK). Ви можете підняти продуктивність межмережного екрана, помістивши правило з established близько до початку ланцюжка;

setup – відповідає, якщо пакет є спробою установки TCP з'єднання (установлений біт SYN, а біт ACK не установлений);

tcpflags прапори – спрацьовує, якщо заголовок TCP містить список перелічених через кому прапорців. Підтримувані прапорці: fin, syn, rst, psh, ack, і urg. Дію правил з окремих прапорців може бути змінено вказівкою префікса !;

icmptypes типи – спрацьовує, якщо тип пакета ICMP перебуває в списку *типи*. Список може бути зазначений у вигляді будь-якої комбінації діапазонів і/або окремих типів, розділених комами. Звичайно використовувані типи ICMP: 0 *echo reply (ping reply)*, 3 *destination unreachable*, 5 *redirect*, 8 *echo request (ping request)*, і 11 *time exceeded* (використовується для позначення закінчення TTL).

Продуктивність міжмережного екрана значною мірою залежить від черговості проходження правил пакетами. Залежно від призначення брандмауера порядок правил може змінюватися. Для брандмауерів загального призначення можна рекомендувати таку послідовність правил:

1. Спочатку йдуть інструкції deny і reject (якщо є), щоб заблокувати шкідливий трафік.
2. Потім count для зовнішнього трафіка, якщо цікава повна статистика.
3. Потім forward, якщо потрібно.
4. Далі allow на сервіси, доступні ззовні.
5. Далі divert.
6. Потім count для трафіка в розрізі внутрішніх хостів, якщо потрібно.
7. Потім allow на внутрішній сегмент мережі, або явна вказівка хостів.
8. Далі allow для дозволених з маршрутизатора/брандмауера з'єднань.
9. І на завершення deny на все, що залишилося.

Приклад настроювання

Розглянемо приклад настроювання брандмауера, конфигуруючого наш сервер для надання користувачам мережі 192.168.0.0/24 доступу в мережу Internet, що захищає сервер від небажаних з'єднань із зовнішнього середовища, і доступ, що надається адміністратору серверу для конфігурування системи з будь-якої зовнішньої мережі. Слід помітити, що приклад створений з ілюстративними цілями, щоб продемонструвати основні можливості ipfw. Деякі правила при промисловому використанні можуть бути організовані більш ефективно.

Створимо файл /etc/firewall/rc.firewall такого змісту:

```
#!/bin/sh

ipfw='/sbin/ipfw -q'
ournet='192.168.0. 1/24'
uprefix='192.168.0'
ifout='rl0'
ifuser='rl1'

${ipfw} flush

${ipfw} add 100 check-state

${ipfw} add 200 deny icmp from any to any in icmptype 5,9,13,14,15,16,17
${ipfw} add 210 reject ip from ${ournet} to any in via ${ifout}

${ipfw} add 300 allow ip from any to any via lo
${ipfw} add 310 allow tcp from me to any keep-state via ${ifout}
${ipfw} add 320 allow icmp from any to any
${ipfw} add 330 allow udp from me to any domain keep-state
${ipfw} add 340 allow udp from any to me domain
${ipfw} add 350 allow ip from me to any

${ipfw} add 400 allow tcp from any to me http,https,ssh
${ipfw} add 410 allow tcp from not ${ournet} to me smtp

${ipfw} add 500 fwd 127.0.0. 1,3128 tcp from ${ournet} to any http out via ${ifout}
${ipfw} add 510 divert natd ip from ${ournet} to any out via ${ifout}

${ipfw} add 1002 allow ip from ${uprefix}.2 to any via ${ifuser}
${ipfw} add 1002 allow ip from any to ${uprefix}.2 via ${ifuser}
${ipfw} add 1003 allow ip from ${uprefix}.3 to any via ${ifuser}
${ipfw} add 1003 allow ip from any to ${uprefix}.3 via ${ifuser}
${ipfw} add 1004 allow ip from ${uprefix}.4 to any via ${ifuser}
${ipfw} add 1004 allow ip from any to ${uprefix}.4 via ${ifuser}

#${ipfw} add 65535 deny ip from any to any
```

Розглянемо запропонований приклад , опустивши поки правила 100, 310 і 330. У перших рядках задаються значення змінних командного інтерпретатора, що описують відповідно клієнтську мережу (ournet), загальну частину всіх адрес наших клієнтів (uprefix), назву "зовнішнього" і "внутрішнього" інтерфейсів (ifout і ifuser) Такі змінні дозволять уникнути випадкових помилок, і дозволять легко змінити настроювання firewall у випадку, наприклад, виходу з ладу мережної карти, і заміни її на мережну карту іншого виробника (у цьому випадку назва зовнішнього інтерфейсу може змінитися, наприклад, на fxp0 для карт Intel EtherExpress).

Правила з номерами 200 і 210 слугують для підвищення «хакеростійкості» системи: правило 210 забороняє поява пакетів з адресою, що належить «внутрішній» мережі, на "зовнішньому" інтерфейсі, тому що улюблена зброя хакера – представити свій пакет як такий, що прийшов із локальної мережі, для якої ступінь довіри вище. Правило 210 заборонить проходження деяких ICMP-пакетів: icmptype 5 – це пакет ICMP-REDIRECT, що може бути використаний під час атаки типу "фальшивий маршрутизатор", інші icmptype – просто розкриють хакеру деяку "зайву" на наш погляд інформацію.

Правила 300-350 забезпечують працездатність самої системи. Правило 300 дозволяє проходження будь-яких пакетів усередині системи (lo – локальний інтерфейс системи, що має адресу 127.0.0.1). 320 правило дозволяє проходження будь-яких ICMP-пакетів, тому що їхнє виникнення погано передбачуване, але їхня втрата загрожує збоями в роботі - наприклад, icmp-пакет є єдиним способом повідомити "викликуваному" комп'ютеру, що "викликувана" адреса не доступна. Правило 340 дозволяє проходження пакетів DNS із зовнішнього середовища на сервер (me – всі локальні адреси серверу). Це має сенс, якщо сервер є DNS-сервером, що підтримує одну або кілька DNS-зон для мережі Internet - наприклад, зону вашої мережі. Правило 350 дозволяє серверу посилати будь-які пакети в усіх напрямах.

Правила 400 і 410 дозволяють користуватися деякими сервісами, надаваними системою. Правило 400 дозволяє всім (і користувачам локальної мережі, і Internet) підключатися до вашого web-серверу й до служби ssh (secure shell – захищена консоль, використовувана для усуненого керування системою). Правило 410 дозволяє прийом вхідної електронної пошти, якщо у вас установлений поштовий сервер.

Правила 500 і 510 забезпечують додаткову обробку користувацького трафіка. Правило 500 перенаправляє весь http-трафик на локальний проксі-сервер. Правило 510 відправляє весь вихідний трафік користувачів на обробку системою NAT для трансляції адрес.

Правила 10xx керують доступом в Internet окремих користувачів. Користувачу з адресою 192.168.0.2 відповідають два правила з номером 1002, що дозволяють проходження будь-якого трафіка до користувача й від користувача. Якщо Ви хочете відключити користувача – просто видаліть ці правила (однією командою – /sbin/ipfw delete 1002), і користувач зможе працювати тільки з вашим внутрішнім web-сервером.

Тепер про правила check-state і keep-state. Ми, мабуть, не хочемо, щоб користувачі зовнішньої мережі вільно підключалися до будь-яких портів серверу. З іншого боку, досить бажано, щоб наш сервер міг з'єднуватися з будь-якою машиною в Internet. Однак не існує способу щодо вмісту одного єдиного IP-пакета визначити чи ініційоване з'єднання нашою системою, чи потенційним зломщиком. Правило з позначкою keep-state дозволяє запам'ятати задовольняюче правилу з'єднання на якийсь час: якщо пакет відповідає правилу keep-state, те ipfw створює динамічне правило, що дозволяє проходження пакетів між адресами, зазначеними в першому пакеті. Наступні пакети

продовжують існування тимчасового правила ще на якийсь час. Якщо активність з'єднання припиняється - правило зі списку зникає, і з'єднання розривається.

Рядок 310 дозволяє проходження TCP-пакетів з локальної машини на будь-яку іншу машину, запам'ятовуючи з'єднання в тимчасовому правилі. Таким чином, дозволяється проходження й зворотні пакети від викликаної машини. Тимчасові правила перевіряються брандмауером при проходженні через правило check-state – тому ми й помістили його на перше місце.

Тепер розглянемо проходження декількох пакетів через firewall.

Нехай користувач 192.168.0.2 бажає подивитися сторінку www.google.com (вважаємо, що на сервері працює transparent-proxy і DNS). Першою справою користувач посилає DNS-запит – UDP-пакет на адресу серверу (192.168.0.1) на порт 53 (domain). Правила 100-330 пакет проходить, не затримуючись, тому що він їх не задовольняє. Правило 340 дозволяє проходження пакета з будь-якої адреси на локальний DNS, тому шлях запиту на цьому закінчується. Відповідь локального DNS-серверу – UDP-пакет з адреси 192.168.0.1, порт 53, на адресу 192.168.0.2 (порт – приміром, 1025, першого користувача, що потрапив під руку комп'ютеру,), проходить за правилом 350, що дозволяє проходження будь-яких пакетів серверу в усіх напрямах. Якщо в кеші локального DNS-серверу не знайдеться запису про сервер www.google.com, то він змушений буде надіслати запит своєму вищестоящому DNS-серверу – UDP-пакет з адресою джерела 195.5.27.16 на адресу 195.5.27.10, що пройде за правилом 330, спричинивши тимчасове правило, що дозволяє проходження зворотного пакета.

Наступним етапом відкриття сторінки буде TCP-з'єднання із сервером www.google.com на порт 80 (http). Користувач, що одержав від DNS-серверу IP-адресу www.google.com, надішле TCP-пакет з адресою відправника 192.168.0.2 (порт типу 1026) і адресою призначення xx.xx.xx.xx, порт 80. Цей пакет пройде по firewall до правила 1002, що дозволяє користувачеві будь-який трафік, після чого потрапить у систему маршрутизації, яка спробує відправити його через зовнішній інтерфейс –
тобто знову потрапить в firewall, але вже як вихідний. У цьому своєму другому проходженні він добереться до правила 500, де його адреса призначення буде відправлена на 127.0.0.1, порт 3128, після чого він потрапить у proxy-сервер. Proxy-сервер згенерує відповідний пакет з адресою призначення 192.168.0.1, що за правилом 350 буде відправлений користувачеві. Після того, як proxy-сервер зрозуміє, яку саме сторінку хоче одержати користувач (для цього користувачу й proxy-серверу потрібно обмінятися ще декількома пакетами за тією схемою), proxy почне власне tcp-з'єднання із сервером www.google.com - відправить tcp-пакет з адреси 195.5.27.16 на адресу xx.xx.xx.xx, порт 80. Цей пакет пройде за правилом 310, спричинивши динамічне правило. Відповідь від www.google.com буде пропущеною цим динамічним правилом при перевірці правила з номером 100 - check-state.

Наступний приклад – звернення користувача до зовнішнього ftp-серверу, наприклад – ftp.freebsd.org. DNS-запит на дозвіл доменного імені пройде аналогічно попередньому прикладу, повернувши користувачу IP-адресу 62.243.72.50. Одержавши адресу серверу, користувач відправить на нього tcp-пакет, що буде відповідати правилу 1002, і потрапить у маршрутизатор, який спробує відправити його через зовнішній інтерфейс.

При спробі "вибратися" із системи пакет дійде до правила 510, яке перенаправить його на обробку natd, і вийде із системи з адресою джерела, що дорівнює нашій зовнішній адресі – 195.5.27.16. При цьому natd запам'ятає в

тимчасовому правилі дійсне джерело пакета, і відповідний пакет, що прийшов, буде переданий користувачу, пройшовши через дозволене правило 1002.

3 Контрольні питання

3.1 Поясніть призначення мережного екрана, і перелічіть основні його функції.
3.2 Дайте коротку характеристику IPFW.
3.3 Присутність яких опцій у ядрі системи необхідно для функціонування пакетного фільтра?
3.4 Поясніть загальну структуру правила ipfw.
3.5 Чим відрізняються дії deny і reject?
3.6 За яких умов можливе ведення статистики пакетів, що проходять через брандмауер?
3.7 Як за допомогою параметра established підняти продуктивність мережного екрана?
3.8 Поясните дію правила: *ipfw add pass tcp from any to any 443 out*.

4 Домашнє завдання

4.1 Вивчыть ключові положення.
4.2 Підготуйте відповіді на ключові питання.

5 Лабораторне завдання

5.1 Скиньте всі старі правила для ipfw.
5.2 Дозвольте проходження всього трафіка для loopback, при цьому убезпечте себе від можливих підробок адреси 127.0.0.1 зовнішніми джерелами.
5.3 Дозвольте роботу по протоколу tftp.
5.4 Дозвольте роботу процедур rpc.bind.
5.5 Дозвольте проходження пакетів dhcp.
5.6 Дозвольте проходження пакетів nfs client status info і nfs lockd.
5.7 Дозвольте роботу icmp, відкидаючи фрагментовані пакети.
5.8 Забезпечте можливість роботи dns.
5.9 Забезпечте можливість роботи http.
5.10 Забезпечте можливість роботи ftp.
5.11 Забороніть усі пакети, що залишилися, і переконайтеся в працездатності створеного фільтра.

Примітка: Доступ до утиліти настроювання ipfw одержати за допомогою програми sudo.

6 Вимоги до змісту протоколу

6.1 Назва лабораторної роботи.
6.2 Мета роботи.
6.3 Результати виконання домашнього завдання.
6.4 Короткий опис проробленої роботи.
6.5 Висновки про виконануну роботу.
6.6 Дата, підпис студента, віза викладача.

Лабораторна робота № 4

НАСТРОЮВАННЯ Й СУПРОВОДЖЕННЯ СЕРВЕРА DHCPD

1 Мета роботи

Конфігурування серверу dhcpd для забезпечення автоматичного настроювання параметрів стека TCP/IP робочої станції в момент її завантаження.

2 Ключові положення

FreeBSD може бути настроєна на роботу як сервер DHCP на основі dhcpd, реалізації пакета DHCP від ISC (Internet Software Consortium). Синтаксис команди dhcpd:

dhcpd [-p порт] [-f] [-d] [-cf файл_настроювання] [-lf файл_оренди] [if0 [ifn]]

dhcpd звичайно запускається без будь-яких аргументів командного рядка. Аргументи використовуються в основному для тестування й налагодження. Два аргументи пов'язані з необхідністю спеціального настроювання:

-f

Наказує dhcpd працювати як додаток першого плану. За замовчуванням dhcpd виконується у фоновому режимі:

if0 [...ifn]

Вказує інтерфейси, на яких dhcpd приймає запити виділення адреси. Перелік фізичних інтерфейсів розділяється пропусками. Може бути використаний для обмеження області функціонування dhcpd.

Всі інші ключі командного рядка використовуються для налагодження й тестування. Наведемо найбільш часто використовувані ключі:

-p порт

Наказує dhcpd посилати пакети через нестандартний порт. Однобічна зміна порту як з боку серверу, так і клієнта, може призвести до непрацездатності всієї схеми видачі адрес:

-cf файл_настроювання

Наказує dhcpd використати зазначений файл конфігурації замість стандартного dhcpd.conf:

-lf файл_оренди

Вказує dhcpd записувати відомості про оренду адрес у довільний файл (замість стандартного dhcpd.leases). Зміна імені файла оренди може призвести до некоректного динамічного виділення адрес. Застосовуйте цей ключ з обережністю.

Конфігураційний файл dhcpd.conf

У момент запуску dhcpd зчитує свої настроювання з файла /usr/local/etc/dhcpd.conf. У dhcpd.conf утримується визначення мережі, що обслуговується даним сервером, й інформація настроювання, передавана сервером клієнтам.

Оператори файла настроювання визначають топологію мережі, що обслуговується сервером DHCP. У документації такі оператори називають

«оголошеннями», оскільки вони декларують певні факти, що відносяться до топології мережі.

З кожним із операторів можуть бути зв'язані параметри й опції. Параметри – це визначення, що ставляться до серверу й протоколу, такі як тривалість оренди адреси або розташування файла мережного завантаження. Опції дозволяють передавати клієнтам стандартні значення DHCP, визначені в різних RFC, наприклад необхідність включити пересилання ір-пакетів.

Параметри й опції, розташовані поза конкретними операторами топології, діють для всіх мереж, що обслуговуються даним сервером. Параметри й опції, зазначені в групуючому операторі, діють для сукупності об'єктів групи. Опції, що діють у глобальних масштабах, можуть перевизначатися тими ж настроюваннями на більш низьких рівнях. Така структура дозволяє адміністратору мережі задавати конфігурацію всієї мережі й окремих її частин. Зупинимося на всіх згадуваних объектах більш докладно.

Оператори топології

share-network name { [parameters] [options] }

Використовується тільки у випадку, коли декілька логічних підмереж перебувають в одній фізичній мережі. Як name може використовуватися будь-яке описове ім'я. Параметри й опції оголошуються всередині фігурних скобок. dhcpd не може визначити, в якій із підмереж повинен перебувати клієнт. Як наслідок, динамічно видавані адреси походять із діапазонів адрес усіх підмереж і призначаються в міру необхідності.

subnet address *mask* netmask { [parameters] [options] }

Визначає IP-адресу й маску кожної підмережі, що обслуговується демоном. Адреси й маски використовуються для ідентифікації клієнтів, що належать підмережі. Параметри й опції усередині фігурних дужок діють для всіх клієнтів підмережі.

group { [parameters] [options] }

Оператор group групує оператори share-network, subnet і host, і дозволяє застосовувати набори параметрів і опцій до всіх елементів групи.

host hostname { [parameters] [options] }

Визначає параметри й опції для окремих клієнтів.

Параметри й опції конфігурації

Оператори параметрів управляють роботою серверу DHCP і протоколу DHCP. Стандартні значення настроювання DHCP, передані клієнтам, визначаються за допомогою операторів опцій. Розглянемо найбільш часто вживані параметри:

range [dynamic-bootp] low-address [high-address];

Параметр range визначає діапазон (верхню й нижню межу) IP-адрес, доступних для динамічного призначення. Параметр range повинен бути зв'язаний з оператором subnet.

hardware type address;

Визначає апаратну адресу клієнта. Параметр hardware повинен бути зв'язаний з оператором host.

fixed-address address[, address...];

Призначає вузлу один або кілька фіксованих IP-адрес. Даний параметр дійсний тільки у поєднанні з оператором host. Якщо зазначено кілька адрес, клієнту

призначається адреса, коректна для мережі, із якої виконує завантаження клієнт. Якщо такої адреси в списку немає, ніякі дані настроювання клієнту не передаються.

autoritative;

not autoritative;

Вказує, чи є сервер DHCP компетентним. За замовчуванням приймається значення autoritative. not autoritative може використовуватися, якщо до компетенції серверу DHCP не входить призначення адрес клієнтам. Сервер DHCP може обслуговувати кілька мереж, мати повноваження призначення адрес в одних сегментах і не мати в інших.

allow keyword;

deny keyword;

Визначає необхідність відповідати на запити різних типів. Ключове слово (keyword) указує тип дозволених (allow) або заборонених (deny) запитів. Існують такі ключові слова:

− unknown-clients – визначає можливість динамічного призначення адрес невідомим клієнтам. За замовчуванням дозволено;

− bootp – визначає необхідність відповідати на запити BOOTP. За замовчуванням дозволено;

− booting – використовується всередині оголошення host для вказівки необхідності відповідати тому або іншому клієнту. За замовчуванням дозволено.

Оператори опцій dhcpd відбивають усі стандартні параметри настроювання DHCP, визначені в існуючих RFC. Розглянемо найбільш корисні з них.

option subnet-mask mask;

Визначає маску підмережі у формі десяткового запису через крапку. Якщо subnet-mask відсутній у файлі dhcpd.conf, dhcpd використає маску підмережі з оператора subnet.

option routers address [, address...];

Перераховує доступні клієнтам маршрутизатори в порядку переваги.

option domain-name-servers address [, address...];

Перераховує доступні клієнтам сервери доменних імен у порядку переваги.

option lpr-servers address [, address...];

Перераховує доступні клієнтам сервери печатки LPR у порядку переваги.

option host-name host;

Вказує ім'я вузла для клієнта.

option domain-name domain;

Визначає ім'я домена.

option broadcast-address address;

Визначає широкомовну адресу для підмережі клієнта.

option static-routes destination gateway [, destination gateway...];

Перераховує доступні клієнту статичні маршрути. Маршрут за замовчуванням не може бути вказаний таким чином.

option nis-domain string;

Рядок символів, що визначає ім'я домена NIS (Network Information Services).

option nis-servers address [, address...];

Перераховує IP-адреси доступних клієнтам серверів NIS у порядку переваги.

Файл dhcpd.leases

Для збереження інформації про видані адреси, на випадок перезавантаження системи або рестарта dhcpd, список уже зайнятих адрес зберігається у файлі **/var/db/**dhcpd.leases. Перед тим як видати адресу хосту, dhcpd записує його в цей файл і примусово скидає буфери на диск, що б гарантувати схоронність інформації навіть на випадок несподіваного краху системи. При старті, після читання файла конфігурації, dhcpd зчитує dhcpd.leases щоб з'ясувати, які адреси вільні, а які зайняті клієнтами. Нові записи додаються в кінець файла. Для запобігання розростання час від часу dhcpd створює новий файл dhcpd.leases, що складається тільки з актуальних записів, а старий перейменовує в dhcpd.leases~.

Тут існує проблема з безпекою системи в плані DoS. Якщо процес dhcpd був примусово «завис» у момент, коли старий файл уже перейменований, а новий ще не переміщений на місце, то після перезавантаження системи виявиться, що відсутній файл **/var/db/**dhcpd.leases. У цьому випадку dhcpd просто не зможе стартувати й буде потрібно ручне втручання. Перше що приходить на розум – створювати пустий файл. Ніколи не робіть цього – у такий спосіб ви ризикуєте надати системі нестабільного стану, втративши інформацію про раніше розподілені адреси. Замість цього просто перейменуйте **/var/db/**dhcpd.leases~ в **/var/db/**dhcpd.leases, тим самим відновивши попередній стан.

Запуск dhcpd в FreeBSD

Для запуску служби DHCP у момент старту системи необхідно у файл /etc/rc.conf помістити такі директиви:

dhcpd_enable="YES"

dhcpd_ifaces="if0"

Замініте ***if0*** ім'ям інтерфейсу, на якому сервер DHCP повинен приймати запити від клієнтів.

Потім ви можете стартувати сервер DHCP за допомогою команди

 # /usr/local/etc/rc.d/isc-dhcpd.sh start

Якщо у майбутньому вам знадобиться зробити зміни в настроюванні вашого серверу, то важливо помітити, що посилка сигналу **SIGHUP** додатку dhcpd не приведе до перезавантаження настроювань, як це буває для більшості демонів. Вам потрібно послати сигнал **SIGTERM** для зупинки процесу, а потім запустити знову його за допомогою команди вище.

Приклад настроювання dhcpd.conf

Розглянемо типовий конфігураційний файл dhcpd.conf, що ілюструє основні можливості DHCPD від ISC.

```
#визначемо значення, що діє  для всіх систем

default-lease-time 86400;
max-lease-time 604800;
get-lease-hostname true;
option subnet-mask 255.255.255.0;
option domain-name "example.org";
option domain-name-servers 192.168.0.135, 192.168.0.130;
```

```
#визначення підмереж, параметрів для окремих пі мереж,,
#діапазону адрес, доступних для динамічного росподілення

subnet 192.168.0.0 netmask 255.255.255.0 {
    option routers 192.168.0.1;
    option broadcast-address 192.168.0.255;
    range 192.168.0.30 192.168.0.70;
}

subnet 192.168.1.0 netmask 255.255.255.0 {
    option routers 192.168.1.1;
    option broadcast-address 192.168.1.255;
    range 192.168.1.80 192.168.1.160;
}

#призначенняе фіксованих  IP-адрес

group {
    use-host-decl-names on;

host test1{
      hardware ethernet 00:17:76:82:3B:C2;
      fixed-address 192.168.0.8;
 }

 host test2{
      hardware ethernet 00:17:76:82:3B:2F;
      fixed-address 192.168.0.9;
 }
}
```

Це приклад настроювання серверу, що поєднує й обслуговує дві підмережі. Сервер динамічно призначає IP-адреси DHCP-клієнтам обох підмереж і здійснює видачу фіксованих адрес по MAC-адресах. Усі рядки, що починаються символом грат (#), є коментарями. Перші кілька активних рядків файла визначають ряд параметрів і режимів, що діють для всіх обслуговуваних сервером підмереж і клієнтів. Перші три рядки містять параметри для серверу. Кожний із трьох параметрів визначає один із аспектів роботи dhcpd при динамічному призначенні адрес.

default-lease-time

Вказує серверу тривалість оренди адреси за замовчуванням (у секундах). Клієнт може запросити оренду адреси на певний період часу, і в цьому випадку його запит виконується. Якщо ж клієнт не вказує значення часу явно, використовується значення за замовчуванням. У нашому випадку тривалість за замовчуванням становить один день (86400 с).

max-lease-time

Визначає максимально припустимий час оренди. Незалежно від тривалості оренди, що фігурує в запиті клієнта, це самий тривалий термін, на який може бути видана адреса. У цьому випадку - 1 тиждень.

get-lease-hostname

Пропонує dhcpd надавати кожному хосту поряд з динамічною адресою ім'я вузла. Для дозволу імені використовується DNS. Пошук імен вузлів для всіх динамічно виділюваних адрес значно збільшує час, необхідний на запуск демона. Значення *true* використвуйте тільки у невеликих мережах, у противному випадку використайте *false*.

Наступні чотири рядки являють собою ключи настроювання. Вони починаються словом *option*. За ним слідує ім'я ключа й призначене йому значення. Ключі визначають значення настроювання, використовувані клієнтом. Про призначення ключів легко догадатися, оскільки їхні імена носять описовий характер. Ми передаємо клієнтам маску підмережі, ім'я домена, й адреси серверів доменних імен.

Оператори *subnet* визначають підмережі, що обслуговують dhcpd. Ідентифікація мереж виробляється на основі адреси й адресної маски – обидва елементи є обов'язковими для оператора subnet. dhcpd передає інформацію про настроювання тільки клієнтам цих мереж. Для кожної підмережі, з якою фізично зв'язаний сервер, повинен бути присутнім окремий оператор subnet, навіть якщо в деяких із підмереж немає клієнтів. Інформація про підмережі потрібна dhcpd при запуску. Як видно, у нашому випадку присутні дві підмережі – 192.168.0.0/24 і 192.168.1.0/24

Дія ключів і параметрів усередині оператора subnet розповсюджується тільки на цю підмережу і клієнтів, що перебувають у ній. Призначення ключів у прикладі зрозуміло. Вони вказують клієнтам, який маршрутизатор і широкомовну адресу використати. Параметр *range* більш цікавий, оскільки прямо зв'язаний з однією із ключових особливостей DHCP.

Параметр range визначає діапазон адрес для динамічного виділення. Діапазон у параметрі range визначається парою зазначених адрес. Перша адреса визначає найменшу адресу, що може бути призначена автоматично, а друга – найбільшу. У нашому випадку адреси виділяються з області 192.168.0.30-192.168.0.70 і 192.168.1.80-192.168.1.160. Параметрів range для однієї підмережі може бути кілька. Так створюються роздільні послідовності динамічно виділюваних адрес. Коли це може пригодитися – наприклад, якщо кілька адрес були призначені раніше статично, і ви не хочете нічого змінювати. За наявності параметра range будь-який клієнт DHCP може одержати IP-адресу. У випадку відсутності range механізм динамічного виділення адрес не діє.

Оператор *group* групує будь-які інші оператори. Єдиним призначенням цього оператора є розповсюдження дії параметрів і ключів на всіх учасників групи. Саме це ми зробили в прикладі. Оператор group об'єднав усі оператори host. Параметр use-host-decl-names діє на всі вузли в групі. Цей параметр пропонує dhcpd привласнити кожному клієнту ім'я вузла, зазначене в операторі host клієнта. Також демонструється призначення фіксованих IP-адрес по фізичних адресах.

3 Контрольні питання

3.1 Поясніть призначення протоколу DHCP.
3.2 Для чого може бути використаний ключ *if*n у команді dhcpd?
3.3 Які оператори топології застосовуються у файлі dhcpd.conf? Дайте коротку характеристику кожному з них.
3.4 Поясніть призначення параметра range. В якому випадку параметр обов'язковий до використання?
3.5 За допомогою яких параметрів може бути виконана передача клієнту

фіксованої IP-адреси?
3.6 В якому випадку DHCP-сервер є компетентним?
3.7 Поясніть призначення файла dhcpd.leases.

4 Домашнє завдання

4.1 Вивчіть ключові положення.
4.2 Підготуйте відповіді на ключові питання.

5. Лабораторне завдання

5.1 Сконфігуруйте компетентний DHCP-сервер на своєму комп'ютері, при цьому забезпечте видачу клієнтам таких параметрів:

- максимально припустимий час оренди – 117600 с;
- ім'я домена – *прізвище_студента.org*;
- маска підмережі – 255.255.255.128;
- адреса сервера імен - 192.168.0.145;
- динамічно видавати адреси з області 192.168.0.180 - 192.168.0.190.

5.2 Для двох довільних хостів видавайте фіксовані IP-адреси (по Ethernet-адресах). Фізичні адреси одержати у викладача.
5.3 Перевірте наявність файла /var/db/dhcpd.leases, у випадку відсутності створити.
5.4 Стартувати dhcpd, переконайтеся в його працездатності.

6 Вимоги до змісту протоколу

6.1 Назва лабораторної роботи.
6.2 Назва роботи.
6.3 Результати виконання домашнього завдання.
6.4 Короткий опис проробленої роботи.
6.5 Висновки про виконану роботу.
6.6 Дата, підпис студента, віза викладача.

Лабораторна робота № 5

НАСТРОЮВАННЯ Й СУПРОВОДЖЕННЯ СЕРВЕРА ProFTPD

1 Мета роботи

1.1 Конфігурування серверу ProFTPD для забезпечення передачі файлів по мережі.
1.2 Одержання навичок настроювання анонімного доступу й віртуальних ftp-серверів.

2 Ключові положення

Програмне забезпечення ftp-серверу складається з ftp-демона і файлів конфігурації. Демон – це програма, що безупинно перевіряє наявність запитів на послуги FTP від вилучених користувачів. При одержанні запиту демон обслуговує процедуру реєстрації й установлює з'єднання за допомогою зазначеного користувачем облікового запису, а також виконує команди ftp відправлені вилученим користувачем. У випадку анонімного доступу ftp-демон дозволяє вилученому користувачу зареєструватися за допомогою облікового запису ftp через введення імені *anonymous* або *ftp***.** Після цього користувач одержує доступ до каталогів і файлів, визначеним для цього облікового запису. А з метою безпеки демон перетворить на час даного сеансу початковий каталог ftp у кореневий каталог, приховуючи в такий спосіб від вилученого користувача іншу частину файлової системи. Звичайно будь-який користувач системи може вільно переходити по каталогах, доступних для нього, тоді як користувач, зареєстрований за допомогою анонімного облікового запису, бачить тільки початковий ftp-каталог і його підкаталоги. Інша частина файлової системи для такого користувача залишається схованою. Це здійснюється із застосуванням операції *chroot*, що перетворить ftp-каталог, доступний для даного користувача, у кореневий каталог. Як правило, ftp-сервер вимагає також, щоб користувач працював з одним із припустимих командних інтерпретаторів. Список таких інтерпретаторів перебуває у файлі /etc/shells.

Сервер ProFTPD

Демон ProFTPD характеризується простою процедурою настроювання й підтримує такі розвинені засоби, як віртуальний хостінг. На відміну від інших ftp-серверів, при використанні ProFTPD не потрібно копіювати системні каталоги й файли у ftp-каталог. Зокрема, не потрібні файли з каталогів /bin або /etc. Сервер ProFTPD може працювати або як автономний сервер, або як сервер, що запускається за допомогою inetd. Для цього у файл proftpd.conf необхідно включити директиву ServerType з відповідним параметром. Крім того, ProFTPD можна настроїти таким чином, що він буде автоматично переходити з режиму роботи через сервер inetd в автономний безперервний режим роботи й назад, залежно від завантаженості системи. Основні ключі, використовувані при старті ProFTPD, наведені в табл. 1.

Таблиця 1 - Опції запуску ProFTPD

Опція	**Описування**
-h, --help	Виводить довідку, у тому числі за опціями запуску
-n, --nodaemon	Запускає процес proftpd в автономному режимі (необхідно установити у файлі конфігурації значення параметра ServerType, що дорівнює standalone)
-v, --version	Виводить номер версії демона ProFTPD
-d, --debug *рівень настроювання*	Установлює внутрішній рівень налагодження демона ProFTPD (1-5)
-c, --config *файл*	Задає альтернативний файл конфігурації
-l, --list	Виводить список усіх модулів, скомпільованих для використання демоном ProFTPD

Файли proftpd.conf і .ftpaccess

Сервер ProFTPD використовує єдиний файл конфігурації, *proftpd.conf***,** розташований у каталозі /usr/local/etc. Записи у цьому файлі мають форму директив. Даний формат файла конфігурації був обраний з метою забезпечення подібності директивам серверу Apache. За допомогою директив конфігурації можна визначати не тільки базові елементи функціонування серверу, зокрема його ім'я, але й виконувати більш складні настроювання, наприклад організовувати віртуальні ftp-хости. Така конструкція виявляється досить гнучкою, дозволяючи задавати конфігурацію для окремих каталогів, користувачів або груп.

Настроювання певного каталогу можна здійснити, використовуючи файл ftpaccess**,** що містить опції конфігурації безпосередньо для даного каталогу. Опції файла ftpaccess мають перевагу перед опціями файла proftpd.conf. Файли .ftpaccess будуються й працюють приблизно так само, як файли .htaccess, використовувані сервером Apache для настроювання окремих каталогів web-вузла.

Директиви конфігурації мають різне призначення. Багато з них установлюють значення параметрів, як, наприклад, директива *MaxClient*, що задає максимально припустиме число клієнтів системи, або директива *ServerName*, що установлює ім'я ftp-серверу. За допомогою низки директив формуються блоки, що містять набір директив, застосовуваних до певних компонентів ftp-серверу. Директиви формування блоків, або *блокові директиви,* вводяться попарно – початкова директива, потім кінцева. Кінцева директива вказує на кінець блока й має те ж ім'я, що й початкова, але з косою рискою на початку. Блокові директиви мають параметр, що визначає конкретний об'єкт, до якого застосовується даний блок. Так, для блокової директиви *Directory* необхідно вказати ім'я каталогу. Наприклад, директива <Directory *mydir*> створює блок, директиви якого застосовуються до каталогу *mydir*. Створений попередньою директивою блок завершується директивою </Directory>. Директива <Anonymous *ftp-dir*> набудовує службу анонімного доступу до ftp-вузла. Для цієї директиви необхідно вказати каталог, виділений у системі під службу анонімного доступу ftp, наприклад /home/ftp. Блок завершується директивою </Anonymous>. Директива <VirtualHost *hostaddress*> дозволяє настроїти віртуальний ftp-сервер і повинна містити IP-адресу або доменне ім'я, використовуване для цього серверу. Відповідна завершальна директива – </VirtualHost>. Всі директиви, що перебувають у межах блока, заданого попередніми двома директивами, застосовуються до зазначеного ftp-вузла. Директива < Limit *permission*> визначає вид доступу, якого необхідно обмежити. Як параметр ця директива приймає кілька ключових слів, що

позначають вид доступу: **WRITE** – на запис, **READ** – на читання, **STOR** – на прийом файлів і **LOGIN** – на реєстрацію.

Анонімний доступ

За допомогою директиви Anonymous можна створити блок і помістити в ньому директиви, що настроюють анонімну службу ftp. У цю директиву включається ім'я каталогу системи, використовуваного даною службою. Демон ProFTPD виконає щодо зазначеного каталогу команду chroot, перетворюючи його в кореневий каталог для вилученого користувача, який одержав доступ до служби ftp. За замовчуванням підтримується анонімна реєстрація, причому як пароль користувачі повинні вводити адресу своєї електронної пошти. Ви можете змінити параметри анонімного доступу й зробити його більш контрольованим, увівши гостьову реєстрацію з паролем.

У випадку використання серверу ProFTPD у каталозі анонімного ftp-вузла не потрібно наявності будь-яких системних файлів. Перш ніж ProFTPD виконає команду chroot і сховає від користувача всю ту частину системи, що перебуває за межами каталогу ftp**,** він одержить доступ до усіх необхідних йому системних файлів.

У наступному прикладі показаний стандартний файл конфігурації анонімного ftp-вузла.

```
<Anonymous /home/ftp>
 User       ftp
 Group     ftp
UserAlias anonymous ftp
RequireValidShell  off
 <Directory *>
       <Limit WRITE>
          DenyAll
       </Limit>
 </Directory>
       //При передачі файлів від клієнта серверу
       //єдиним припустимим параметром є STOR
       //
              <Directory incoming>
     <Limit READ WRITE>
          DenyAll
     </Limit>
     <Limit STOR>
          AllowAll
     </Limit>
   </Directory>
</Anonymous>
```

Перша директива Anonymous визначає каталог /home/ftp як початковий каталог ftp-вузла. Директива User задає ім'я користувача, з яким буде працювати анонімний ftp-демон, а також ім'я групи цього користувача. В обох випадках застосовується ім'я ftp – стандартне ім'я, використовуване у більшості систем для анонімного ftp-вузла. Директива Directory із символом * як шаблон імен файлів визначає блок директив, що буде застосований до всіх підкаталогів каталогу /home/ftp. Шаблон * підходить для будь-яких імен файлів і каталогів. Усередині першого блока Directory перебуває

директива Limit, що накладає обмеження на право користувачів здійснювати різні дії в каталозі. Ця директива приймає кілька параметрів, включаючи **READ** для позначення режиму читання й **WRITE** – режиму запису. У даному прикладі директива Limit обмежує права користувачів на запис. Усередині директиви Limit директива DenyAll забороняє запис, тому користувачі не зможуть створювати й видаляти файли, а будуть мати право тільки читати їх. Друга директива Directory задає виключення із установленого вище правила для каталогу incoming**,** призначеного для прийому файлів від вилучених користувачів. Для цього каталогу перша директива Limit забороняє запис і читання всім користувачам (DenyAll), тим самим забороняючи видаляти й переглядати файли. Друга директива Limit з параметром STOR дозволяє зберігати в даному каталозі передані користувачами файли (директива AllowAll).

Важною директивою для настроювання анонімних ftp-вузлів є RequireValidShell, що забезпечує використання тільки припустимого командного інтерпретатора. За замовчуванням ftp-демон насамперед перевіряє, чи реєструється вилучений користувач для роботи із припустимим командним інтерпретатором, таким як bash або C-shell. Якщо використовуваний вилученим користувачем командний інтерпретатор не є припустимим, з'єднання розривається. Ви можете відключити таку перевірку за допомогою директиви RequireValidShell з опцією off. Тоді вилучений користувач зможе зареєструватися за допомогою будь-якого командного інтерпретатора.

Відзначимо, що протокол FTP спочатку призначався для того, щоб вилучені користувачі могли установлювати з'єднання з використанням власного облікового запису. Тепер же за допомогою служби ftp вилучені користувачі можуть реєструватися в системі за допомогою різних облікових записів. Анонімні користувачі мають право реєструватися тільки із застосуванням облікового запису anonymous. Однак ви можете назвати інших користувачів з їх власними початковими каталогами й так званими гостьовими обліковими записами, які прирівнюються до анонімних користувачів. При реєстрації за допомогою таких облікових записів вилучені користувачі повинні знати своє реєстраційне ім'я й, звичайно, пароль. Після установлення з'єднання вони одержують право тільки на читання файлів, доступ до яких дозволений у відповідному обліковому записі, а інша частина системи залишається від них схованою. По суті, ви створюєте ще один анонімний ftp-вузол у системі, але з більш обмеженим доступом.

Віртуальні ftp-сервери

Демон ProFTPD може обслуговувати відразу кілька ftp-вузлів. За допомогою директиви VirtualHost можна додати ще один набір директив настроювання окремого ftp-серверу. Директива VirtualHost звичайно застосовується для настроювання як FTP-вузли так званих *віртуальних серверів.* Для цього використається здатність ОС Unix підтримувати кілька Ip-адрес. З цією метою додаткові адреси можуть застосовуватися віртуальними серверами, що не є самостійними комп'ютерами, але умовно вважаються такими. Завдяки цьому на одному комп'ютері можна організувати кілька ftp-вузлів, привласнивши їм різні IP-адреси. Оскільки такий додатковий вузол не працює незалежно на окремій машині, він називається віртуальним ftp-сервером. Коли вилучений користувач звертається до системи за адресою віртуального ftp-вузла, демон ProFTPD розпізнає цей запит і діє як сервер даного вузла. Сервер ProFTPD може обслуговувати одночасно безліч віртуальних вузлів. Скориставшись

можливостями його настроювання, ви можете призначити кожному ftp-вузлу свою специфічну роль: одному − роль гостьового вузла, іншому − анонімного вузла для певної групи, третьому − анонімного вузла для конкретного користувача й т.д.

Віртуальний ftp-вузол настроюється шляхом уведення у файл proftpd.conf блока директиви VirtualHost для даного вузла. Блок відкривається початковою директивою VirtualHost з параметром у вигляді IP-адреси й завершується директивою </VirtualHost>. Для створення анонімного або гостьового вузла потрібно уставити в блок директив Anonymous і Guest. Можна навіть увести директиви для конкретних каталогів, що робиться за допомогою директиви Directory. Використовуючи при настроюванні автономних серверів директиву Port, можна створити віртуальний хост, що працює на тім же комп'ютері, але підключений до іншого порту.

```
<VirtualHost 192.168.1.1>
ServerName "Test virtual FTP server"
 </VirtualHost>
```

Віртуальні хости в автономній конфігурації й конфігурації, призначеної для роботи із сервером inetd, працюють по-різному. Демон inetd, виявивши запит до віртуального хоста, передає його ftp-серверу. Він аналізує адресу й порт, зазначена в запиті, й обробляє запит для відповідного віртуального серверу. В автономній конфігурації ftp постійно стежить за надходженням запитів на всі задані порти й генерує для їхнього оброблення дочірні процеси. В автономній конфігурації демон ProFTPD може одночасно підтримувати велике число віртуальних хостів.

У прикладі нижче показаний файл конфігурації віртуального ftp-серверу. У директиві VirtualHost як параметри використовуються доменні імена. Доменному імені повинна бути поставлена у відповідність IP-адреса на сервері доменних імен мережі. У свою чергу IP-адреса повинна указувати на комп'ютер, де працює ProFTPD. На віртуальному сервері ftp.example.org настроюється анонімний гостьовий обліковий запис user, що вимагає при реєстрації уведення пароля. Настроюється також анонімний обліковий запис ftp, що використає початковий каталог /home/ftp/virtual/data.

```
<VirtualHost  ftp.example.org>

        ServerName              "Test Virtual FTP  Server"
        MaxClients              10
        MaxLoginAttempts        1
        DeferWelcome                    on

 <Anonymous      ~user>
        User        user
        Group       user
        AnonRequirePassword   on

<Anonymous  /home/ftp/virtual/data>

        User            ftp
        Group           ftp
        UserAlias       anonymous ftp
```

</Anonymous>

</VirtualHost>

Директиви конфігурації серверу ProFTPD

Найбільш часто використовувані конфігураційні директиви серверу ProFTPD наведені в табл. 2.

Таблиця 2 - Параметри настроювання proftpd.conf

Директива	Описування
Allow ["from"] "all" | "none" | *host* | *network [,host* | *network[,...]]*	Використається усередині директиви <Limit> для явної вказівки того, які хости й/або мережі мають доступ до команд, для яких установлені обмеження доступу. Використовується з ключовими словами Order і Deny для завдання правил керування доступом. Значення за замовчуванням: Allow from all. Контекст: <Limit>
AllowAll	Дозволяє доступ до блоків <Directory>, <Anonymous> або <Limit>. Значення за замовчуванням: доступ дозволений усім. Контекст: <Directory>, <Anonymous>, <Limit>, .ftpaccess
AllowGroup *список_груп*	Задає список груп, яким дозволений доступ у блоці Limit. Значення за замовчуванням: відсутні. Контекст: <Limit>
AllowUser *список користувачів*	Задає список користувачів, яким дозволений доступ. Значення за замовчуванням: відсутнє. Контекст: <Limit>
AnonRequirePassword on | off	Установлює або знімає вимогу уведення при анонімній реєстрації пароля, що повинен збігатися з паролем користувача, який запустив FTP-демон. Використається для створення гостьового облікового запису, що функціонує як анонімний, але вимагає уведення пароля. Значення за замовчуванням: AnonRequirePassword off. Контекст: <Anonymous>
<**Anonymous** *кореневий катаюг*> **Bind** *адреса*	Створює анонімний обліковий запис FTP. Завершується відповідною кінцевою директивою </Anonymous>. Значення за замовчуванням: відсутнє. Контекст: конфігурація серверу, <VirtualHost> Дозволяє прив'язку додаткової IP-адреси до основного або віртуальному хоста. За допомогою декількох таких директив можна прив'язати кілька адрес до одному хоста. Значення за замовчуванням: відсутнє. Контекст: конфігурація серверу, <VirtualHost>
DefaultRoot *каталог [список_груп]*	Кореневий каталог, призначуваний за замовчуванням користувачу при реєстрації в системі. Значення за замовчуванням: DefaultRoot /. Контекст: конфігурація серверу, <VirtualHost>,
Deny ["from"] "all" | "none" |	Список хостів і мереж, яким явно заборонений доступ

host | network [,host network[,...]]	у даному блоці Limit. Параметр all означає, що всім хостам доступ заборонений, а параметр none – що доступ явно не заборонений нікому. Значення за замовчуванням: відсутнє. Контекст: <Limit>
DenyAll	Забороняє анонімним користувачам доступ до каталогу або до компонентів, зазначених у блоці Limit. Значення за замовчуванням: відсутнє. Контекст: <Directory>, <Anonymous>, <Limit>, .ftpaccess
DenyUser *список користувачів*	Забороняє доступ користувачам, зазначених у блоці Limit. Значення за замовчуванням: відсутнє. Контекст: <Limit>
<**Directory** *дорожне_ім'я*>	Задає специфічні правила доступу до зазначеного каталогу і його підкаталогів. Значення за замовчуванням: відсутнє. Контекст: конфігурація серверу, <VirtualHost>, <Anonymous>, <Global>
<**Global**>	Глобальний блок параметрів конфігурації Global використовується з метою створення набору директив, застосовуваних для настроювання як основного, так і всіх віртуальних хостів. Значення за замовчуванням: відсутнє. Контекст: конфігурація серверу, <VirtualHost>
<**Limit** *команда група_команд [команда2 ...]*>	Задає обмеження на виконання FTP -команд у межах даного контексту. Значення за замовчуванням: відсутнє. Контекст: конфігурація серверу, <VirtualHost>, <Directory>, <Anonymous>, <Global>, .ftpaccess
MaxClients *число | none повідомлення*	Задає максимально припустиме число підключених клієнтів. Зазначене повідомлення виводиться, якщо користувачу відмовлено в доступі. Значення за замовчуванням: MaxClients none. Контекст: конфігурація серверу, <Anonymous>, <Global>
Order *allow, deny | deny, allow*	Задає черговість виконання директив Allow і Deny усередині блока Limit. Значення за замовчуванням: Order allow, deny. Контекст: <Limit>
ReguireValidShell **on | off**	Дозволяє або забороняє реєстрацію з командних інтерпретаторів, не перерахованих у файлі /etc/shells. Значення за замовчуванням: RequireValidShell on. Контекст: конфігурація серверу, <VirtualHost>, <Anonymous>, <Global>
ServerType *ідентифікатор_режиму_серверу*	Задає режим роботи серверу: за допомогою серверу inetd або автономний (standalone). Значення за замовчуванням: ServerType standalone. Контекст: конфігурація серверу
TimeoutLogin *час*	Максимальний час, протягом якого користувач повинен зареєструватися. Значення за замовчуванням: відсутнє. Контекст: конфігурація серверу
Umask *маска_у_восьмирічному_коді*	Задає права доступу, призначувані знову створеному файлу або каталогу в межах даного контексту.

	Значення за замовчуванням: відсутнє. Контекст: конфігурація серверу, <Anonymous>, <VirtualHost>, <Directory>, <Global>, .ftpaccess
User *ім'я _користувача*	Містить ім'я користувача, привласнене демонові ProFTPD. Значення за замовчуванням: відсутнє. Контекст: конфігурація серверу, <VirtualHost>, <Anonymous>, <Global>
UserAlias *псевдонім ім'я_користувача*	Містить реєстраційне ім'я, що вводиться клієнтом (псевдонім), ставиться у відповідність імені користувача, заданій в обліковому записі на сервері. Часто використається у блоці Anonymous для дозволу анонімної реєстрації під псевдонімом. Значення за замовчуванням: відсутнє. Контекст: конфігурація серверу, <VirtualHost>, <Anonymous>, <Global>
<VirtualHost *адреса*>	Директива конфігурації, застосовувана до певного імені хоста або IP-адресі. Часто використається для настроювання додаткових віртуальних серверів, що працюють на тій же фізичній машині. Блок завершується директивою </VirtualHost>. Значення за замовчуванням: відсутнє. Контекст: конфігурація серверу

Команди для керування ftp-сервером

За допомогою команди *ftpshut* можна зупинити ftp-сервер у зазначений час, а не припиняти його роботу зненацька для всіх, знищуючи серверний процес. Це дозволяє заздалегідь попередити користувачів про закінчення роботи, для того, щоб вони не починали в цей час завантаження файлів великого обсягу. Команда ftpshut має ряд опцій, за допомогою яких можна вказати час останова й увести текст попереджувального повідомлення. Як параметр команди ftpshut повинен бути зазначений час, після закінчення якого буде виконаний останов, а потім текст попереджувального повідомлення, переданого користувачам. Час може задаватися словом (наприклад, *now* – негайно), числом хвилин до вимикання зі знаком + спереду або значенням часу доби у форматі ЧЧММ, де ЧЧ – година у 24-годинному форматі, а ММ – хвилини. У наведеному нижче прикладі показано, як виконати останов ftp-серверу через 10 хвилин і направити користувачам попереджувального повідомлення:

ftpshut +10 "Shutdown in ten minutes"

Після видачі команди останова буде заборонено встановлення нових з'єднань із FTP-сервером за 10 хвилин до запланованого часу вимикання. За необхідності цей час можна змінити, додавши опцію -l і вказавши після неї необхідне число хвилин. За п'ять хвилин до останова всі поточні з'єднання будуть розірвані. Це час можна змінити за допомогою опції -d. Текст попереджувального повідомлення повинен містити не більше 75 символів.

За допомогою команди *ftpwho* можна визначити, хто в цей момент підключений до ftp-серверу. Вона надає також поточну інформацію про системні процеси, запущені кожним користувачем. На екран виводиться п'ять полів з інформацією про ідентифікатор процесу, назву з'єднання і його стан, кількість процесорного часу, викори-

стовуваного процесом, та іншими відомостями про з'єднання. Стан з'єднання позначається символами R – активне, S – що перебуває в стані очікування й Z – що завершився аварійно. Серед інших параметрів присутні IP-адреси комп'ютера, з яким установлене з'єднання, ім'я користувача, назва виконуваного завдання (наприклад, завантаження файла). Це поле починається з імені ftp-демона, звичайно ftpd**,** за яким йдуть інформаційні сегменти, розділені двокрапками.

Команда *ftpcount* виводить інформацію про кількість користувачів, підключених тепер до FTP-серверу, з розбивкою за класами, зазначеними у файлі **.ftpaccess**. Ця команда показує поточне число користувачів, а також максимально припустиме число користувачів.

Приклад файла proftpd.conf

```
# Це приклад файла конфігурації серверу ProFTPD
ServerName          "ProFTPD Default Installation"
ServerType          standalone
DefaultServer        on

# Порт 21 є стандартним для FTP
Port                21
    Umask               022.
    MaxInstances        30

# Завдання імені користувача й групи, з якими звичайно
# працює сервер
User                nobody
Group               nobody

# Звичайно дозволяється перезапис файлів
<Directory /*>

AllowOverwrite     on
</Directory>

# Проста конфігурація анонімного FTP-вузла,
# утримуючого лише один каталог для прийому файлів
<Anonymous ~ftp>
User                ftp
Group               ftp
RequireValidShell   off
MaxClients          10
# Клієнти повинні мати можливість реєструватися
# як під ім'ям anonymous, так і під ім'ям ftp
UserAlias     anonymous    ftp


# При реєстрації у вікні повідомлення повинен відображатися вміст
```

```
# файла ‘welcome.msg’, а при переході в інший каталог - файла '.message'
DisplayLogin                welcome.msg
DisplayFirstChdir           .message
# Заборонити запис в усіх каталогах ftp, крім призначеного
#для прийому файлів
<Directory *>
    <Limit WRITE>
        DenyAll
    </Limit>
</Directory>

<Directory incoming>
    <Limit WRITE>
        AllowAll
    </Limit>
    <Limit READ>
        DenyAll
    </Limit>
</Directory>

</Anonymous>
```

3 Контрольні питання

3.1 Назвіть складові серверу ProFTPD.
3.2 Які види доступу можливі до ftp-серверу?
3.3 Якими можливостями володіє сервер ProFTPD?
3.4 При установці правил доступу до каталогу одночасно в proftpd.conf і .ftpaccess якому файлу буде віддана перевага?
3.5 Назвіть атрибути доступу, що є присутніми у директиві Limit. Поясніть кожен з атрибутів.
3.6 Чим визначається кореневий каталог користувача при анонімному доступі до ftp-серверу?
3.7 Проаналізуйте приклад файла proftpd.conf. Поясніть правила доступу до ресурсів, установлювані даним файлом.

4 Домашнє завдання

4.1 Вивчіть ключові положення.
4.2 Підготуйте відповіді на ключові питання.

5 Лабораторне завдання

5.1 Сконфігуруйте сервер ProFTPD для роботи з наступними параметрами:
- при підключенні виведіть коментар «Student0x FTP Server» (тут *x* - номер робочого місця);
- сервер ftp запустіть автономно;
- як початковий каталог ftp установіть домашній каталог користувача;

− максимальна кількість одночасних з'єднань - 10;

− максимально припустиме число клієнтів установити, як таким, що дорівнює 3, при відмові у з'єднанні виведіть повідомлення “Sorry, max %m users - try again later”;

− максимальний час, протягом якого користувач повинен зареєструватися установіть як такий, що дорівнює 200 с;

− дозвольте реєстрацію на сервері користувачу student0x, іншим забороніть.

5.2 Забезпечте можливість анонімного доступу з наступними параметрами:

− початковий каталог для анонімного доступу – /var/ftp/pub;
користувач – ftp, anonymous, група – ftp;

− дозвольте безпарольний доступ користувачу ftp;

− скасуйте всі письмові операції в анонімній кореневій директорії й піддиректоріях, за винятком директорії /var/ftp/pub/incoming, де клієнт може зберігати інформацію, але нічого не зчитуйте.

5.3 Настройте додатковий мережний інтерфейс з IP-адресою 192.168.1.*x*, (*x*- номер робочого місця), шляхом створення аліаса в команді ifconfig.

5.4 Настройте віртуальний ftp-сервер з IP- адресою з попереднього пункту.

5.5 Стартуйте proftpd через sudo.

5.6 Увійдіть на сервер з обліковим записом student0x, переконайтеся у працездатності й коректності настроювань.

5.7 Отримайте анонімний доступ до серверу, зробіть висновки про те, яка ділянка файлової системи доступна при такому виді доступу.

5.8 Підключіться до віртуального серверу, переконайтеся в його працездатності.

6 Вимоги до змісту протоколу

6.1 Назва лабораторної роботи.

6.2 Мета роботи.

6.3 Результати виконання домашнього завдання.

6.4 Короткий опис проробленої роботи.

6.5 Висновки про виконану роботу.

6.6 Дата, підпис студента, віза викладача.

Лабораторна робота № 6

НАСТРОЮВАННЯ Й СУПРОВОДЖЕННЯ ПОШТОВОГО СЕРВЕРА Postfix

1 Мета роботи

1.1 Виконання базового настроювання поштового сервера.
1.2 Дослідження взаємодії між компонентами поштового сервера.
1.3 Тестування настроювань поштових скриньок користувачів.

2 Ключові положення

У даній роботі ми розглянемо створення поштового серверу на базі Postfix. Цей пакет програмного забезпечення став досить популярний серед адміністраторів усіх видів UNIX-систем. Причина такого розвитку подій тривіальна - Postfix характеризується як максимально проста в освоєнні, зручна в настроюванні й надзвичайно стійка система. Давайте подивимося, як виглядає зсередини поштова система, побудована на основі Postfix.

Основа системи побудована на використанні декількох напіврезидентних програм. Кожна з них відповідно до філософії UNIX виконує тільки одне завдання, але робить це добре. На відміну від стандартного підходу, за якого головна програма викликає допоміжні в міру необхідності й знищує їх, коли вони вже не потрібні, резидентность службових програм дозволяє заощадити на створенні нових процесів. Втім, якщо грамотно розподілити етапи обробки пошти для кожної програми й не сильно дробити весь процес, то й класичний підхід буде цілком припустимий. У той же час немає необхідності у значній кількості одночасно працюючих програм, що самостійно виконують ті ж самі дії для листів, що перебувають в обробленні. Їх усі можна замінити одним або декількома резидент ними примірниками потрібної підсистеми, що обробляють по черзі всі запити, що надходять. Відповідно будь-яка службова програма за необхідності може звернутися із запитами до будь-якого іншого компонента поштової системи. Такий твердий розподіл на підсистеми дозволяє легко відключати ті модулі, які вам не потрібні.

Перейдемо до детального розгляду архітектури Postfix. На чолі всього стоїть демон master, що звичайно запускається командою Postfix при старті системи й працює постійно доти, поки діє поштова система. Цей процес відповідає за запуск на вимогу всіх інших демонів і перезапуск тих з них, які передчасно завершили свою роботу через будь-які проблеми. Його обов'язком також є стежити, щоб кількість дочірніх процесів не перевищувало обмеження установлених у файлі master.cf, і щоб кожний з них жив не довше, ніж визначено настроюваннями поштової системи.

Яким же способом листа попадають у поштову систему? Найпоширеніший випадок – коли нові листи приходять через мережне з'єднання. Уперше їх одержує smtp-демон (рис. 1). Якщо адміністратор включив відповідну можливість, то буде зроблена перевірка, чи не попадає лист під обмеження UCE-фільтрації. UCE (unsolicited commercial email) – незатребувана комерційна пошта, або, простіше, спам. Після цього з отриманим листом починає працювати процес cleanup, що займається його зачищенням. Він проводить первісні перевірки на правильність оформлення й формат вхідного повідомлення, дозволяючи захистити іншу систему від багатьох атак, розрахованих на переповнення буфера. За необхідності додає в лист відсутні службові

заголовки й за допомогою процесу trivial-rewrite приводить адреси до виду: *користувач@поддомен. домен*. На даний момент у postfix немає повноцінної мови, що дозволяє керувати процесом переписування адрес, тому замість нього використовуються службові таблиці canonical і virtual для зберігання правил, що керують перезаписом.

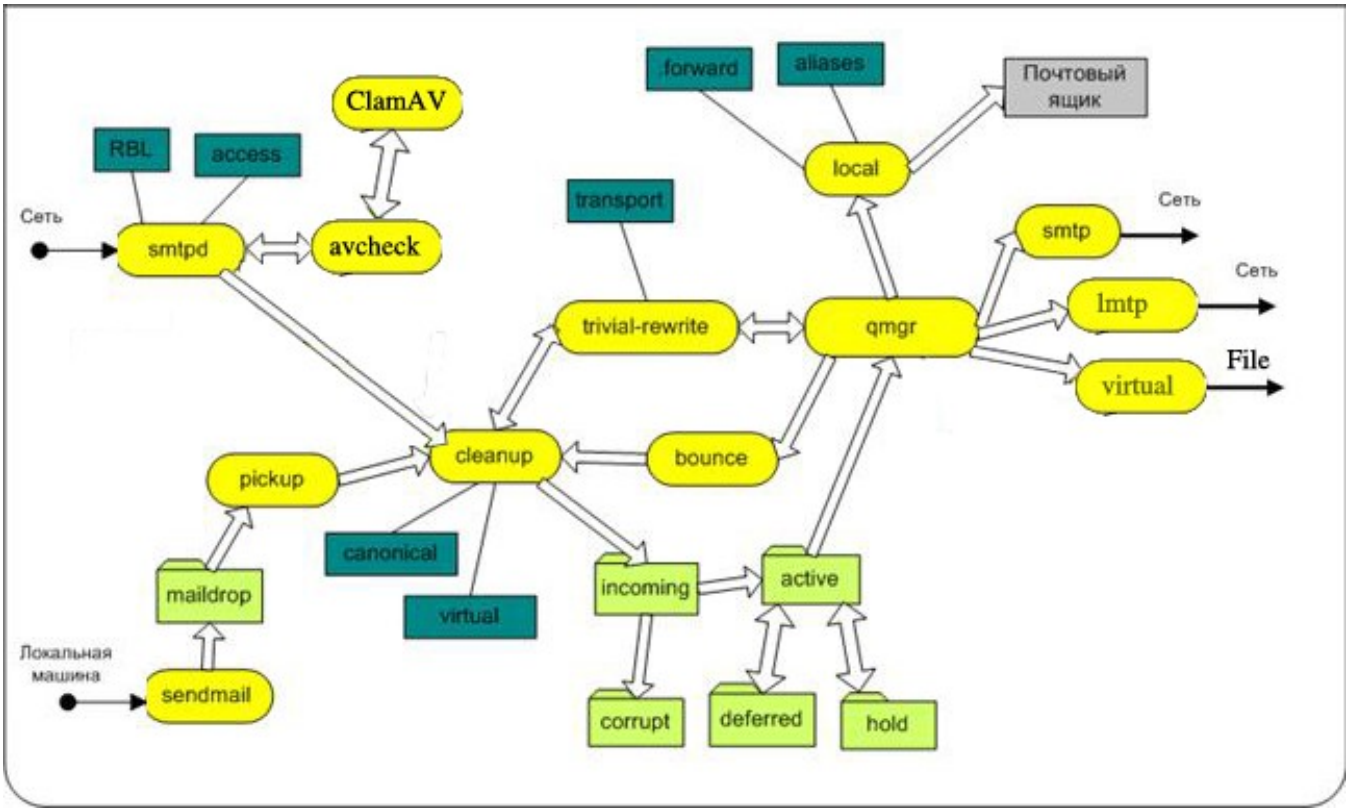

Рисунок 1 - Структурна схема серверу Postfix

Після такого оброблення лист зберігається на диск у форматі файла поштової черги й кладеться в директорію, де зберігається черга incoming, звичайно це /var/spool/postfix/incoming/. Ця черга використовується тільки для оброблення вхідних повідомлень. Потім процес cleanup повідомляє процес queue manager, що керує всіма чергами, про прибуття нової пошти.

Наступний випадок − лист, відправлений з нашої локальної системи. У багатьох UNIX-системах як поштова програма традиційно використовується sendmail. Більшість скриптов, написаних раніше й використовуваних зараз апріорі вважає, що в системі установлений саме цей різновид поштового серверу. Щоб не зруйнувати сумісність із величезною кількістю програмного забезпечення при переході до використання Postfix, у систему разом з іншими файлами установлюється програма, що замінює sendmail. Таким чином, виходить, що команда mail передає лист програмі sendmail, установленої Postfix, а та у свою чергу викликає привілейовану програму postdrop. Ну, а остання вже кладе файли з листами в службову директорію maildrop, що звичайно перебуває в /var/spool/postfix/maildrop/. Далі лист підхоплює процес pickup і аналогічно smtp-демону перевіряє відповідність послання установленим форматам, у результаті чого лист попадає до вільного на даний момент процесу cleanup. У випадку, якщо лист не піддається корекції, він знищується. Варто відзначити, що відправник не одержить ніяких повідомлень про даний факт. Якщо ж нічого

аномального не було знайдено то повідомлення безперешкодно потрапить у чергу incoming, а queue manager одержить сигнал про появу нових листів.

Нова пошта також може з'явитися у випадку, якщо лист неможливо доставити адресату й настроювання системи диктують відправляти сповіщення відправнику. Тоді процес bounce або defer генерує лист із повідомленням про проблему. Адресовано воно, звичайно ж, відправникові первісного повідомлення. Іншим джерелом виникнення листів може бути настроювання Postfix, що змушує його відправляти адміністратору повідомлення у випадку проблем з SMTP або порушення тих або інших політик настроювань поштової системи. Відповідно в обох випадках, щоб доставити лист адресату, доводиться, як звичайно, віддати його процесу cleanup і потім відправити в чергу incoming.

Ми розібралися з тим, якими шляхами листи з'являються всередині поштової системи. Тепер усе, що перебуває в черзі incoming, потрібно доставити одержувачам. Давайте подивимося, як це відбувається. Одержавши від cleanup повідомлення про надходження нової пошти демон queue manager, що керує чергами, перекладає лист у чергу active. Слід зазначити, що ця черга дуже маленького розміру. В ній може перебувати всього лише кілька листів, які в цей момент перебувають у процесі доставлення. Зроблено з по двох причин. По-перше, для того щоб при великому навантаженні менеджер черг не споживав занадто багато пам'яті. Друга причина полягає в тому, що у випадку, коли сторона, що приймає не спроможна одержувати листи з тією швидкістю, з якою Postfix намагається їх передати, доводиться пригальмувати швидкість відправлення. З малою чергою це робити набагато простіше. Поклавши лист у чергу active, процес queue manager створює запит до процесу trivial-rewrite, за допомогою якого визначається точка призначення повідомлення. У відповідь можна буде довідатися, чи локальний одержувач, чи вилучений. Додаткову інформацію про маршрутизацію пошти можна одержати з файла transport. Залежно від отриманих даних queue manager входить у контакт із одним з наступних агентів доставки:

− local − використається для доставки пошти усередині локальної системи. Уміє працювати зі стандартними для UNIX поштовими скриньками. У випадку, якщо для даного користувача не заведено системних псевдонімів у файлі псевдонімів, звичайно це /etc/aliases і немає файлів локального перенапрямку .forward, які будь-який користувач може створити у своїй домашній директорії, то лист відразу ж попадає в цільову поштову скриньку. У противному випадку лист буде передано туди, куди вони вказують;

− virtual − агент віртуальної доставки, являє собою дуже урізану версію агента локальної доставки. Тому він може працювати тільки з поштовими скриньками у форматі mailbox. У ньому також відсутня підтримка системної бази псевдонімів і файлов.forward. За рахунок таких обмежень даний агент доставки вважається самим безпечним з усіх;

− smtp-клієнт, є ще одним агентом, вступає в дію в той момент, коли потрібно доставити лист користувачу вилученої системи. Звичайно queue manager передає йому наступні дані: ім'я файла черги, адресу одержувача, хост або домен, куди потрібно доставити пошту, адресу відправника. Першою справою за допомогою DNS-запиту потрібно одержати список поштових серверів для цільового домена й відсортувати його за пріорітет. Далі ми починаємо пробувати кожну адресу доти, поки не знайдемо ту, яка перебуває в робочому стані. Звичайно доставка йде одночасно відразу для декількох доменів, тому в системах, через які проходить великий потік пошти, можна побачити трохи smtp-клієнтів, що працюють паралельно.

Якщо доставка завершилася вдало, то smtp-клієнт модифікує файл поштової черги так, щоб було зрозуміло, що він оброблений. У противному випадку клієнт повідомляє queue manager або про фатальну помилку, або про тимчасові утруднення. Проблеми першого типу можуть бути викликані відсутністю потрібного користувача вилученої системи, а другі, приміром, неполадками в мережі або непрацездатністю приймаючого серверу. У першому випадку процес bounce посилає відправнику сповіщення про невдачу початої дії й робить запис до протоколу роботи поштової системи за допомогою syslogd. А в другому лист міститься в спеціальну чергу deferred, де він повинен чекати наступної спроби доставки;

−lmtp-клієнт працює точно так само, як і smtp-клієнт, хіба що протокол використовується інший;

−Pipe − mailer-інтерфейс, призначений для роботи із зовнішніми транспортними агентами. Прикладом такої взаємодії може бути робота з UUCP.

Розглянемо кілька типів черг, що є присутніми у Postfix. У випадку, якщо доставка не вдалася, менеджер черг переправляє файл, в якому зберігається лист, у директорію, де перебуває черга deferred. На файл із листом ставиться часовий штамп, що знаходиться в майбутньому, відповідно наступне оброблення цього файла відбудеться саме в той момент, на який вказує штамп. Періодично менеджер черг перевіряє, чи не прийшов час розпочати наступну спробу доставлення відкладених повідомлень. У випадку, якщо є необхідність виконати чергову спробу раніше, ніж мине тайм-аут, можна скористатися командою postfix flush.

Наступним типом є черга corrupt. В неї попадають усі ушкоджені, що не читаються або неправильно відформатовані файли поштових черг. Така обережність дозволяє ізолювати підозрілі дані доти, поки адміністратор системи не вирішить, що з ними робити.

Остання з існуючих у системі черг називається hold. Тут зберігаються листи, доставка яких припинена за тих або інших причин. Вони будуть знаходитися в цій черзі, поки не надійде спеціальна команда, що виводите їх зі стану паузи.

Postfix має досить складний внутрішній пристрій. На момент останнього релізу вихідний код нараховував 30 000 рядків після видалення коментарів. У такому великому комплексі доводиться як можна ретельніше піклуватися про безпеку системи. Для цього застосовуються наступнізаходи:

− використання найменших привілеїв. Postfix може бути легко обмежений за допомогою chroot і працювати від імені самого безправного користувача;

− жодна зі службових програм не використовує SUID-біт;

− між програмами, відповідальними за доставку пошти, і користувальницькими процесами, що працюють у системі, відсутні відносини батько-нащадок. Це дозволяє звести нанівець спроби використання експлоїтів, заснованих на тому, що зловмисний батьківський процес може передавати дочірньому спеціально змінені змінні середовища, сигнали, відкриті файли;

− пам'ять для текстових фрагментів і службових буферів виділяється динамічно, що дозволяє значно знизити ймовірність успішного використання помилок переповнення буфера. Занадто більші текстові масиви обробляються після розрізування на фіксовані фрагменти. Після завершення всіх потрібних дій вони знову склеюються воєдино. Такий підхід дозволяє суттєво зменшити вимоги програми до наявності оперативної пам'яті;

− кількість об'єктів, щознаходяться у пам'яті, суворо обмежена. Відповідно навіть під великим навантаженням Postfix не буде споживати занадто багато системних ресурсів.

Команди Postfix

Команда *Postfix* дозволяє управляти роботою поштової системи Postfix: запускати й завершувати роботу демона *master*, здійснювати перевірку цілісності й виконувати низку інших схожих за тематикою операцій. Команда установлює стандартизоване середовище для роботи й запускає shell-скрипт *postfix-script*, що і виконує безпосередньо роботу системи. Команда postfix передбачає наступний синтаксис:

postfix [-c config_dir] [-D] [-v] command

− **c** *config_dir* − повний шлях до каталогу, в якому розташовані конфігураційні файли Postfix. Використайте цей параметр для вказівки того, який саме із примірників Postfix використати (передбачається, що їх на машині установлено декілька);

− **D** (використається тільки з командою *postfix start*) − запускають усі демони серверу Postfix під керуванням налагоджувача, зазначеного через конфігураційний параметр *debugger_command*;

− **v** − активізує видачу повідомлень про роботу команди з метою налагодження.

На даний момент доступні такі команди:

check − перевіряє конфігурацію системи Postfix. Видає повідомлення про проблеми, пов'язаних з правами на файли й каталоги, і створює необхідні каталоги, якщо вони відсутні;

start − запускає поштовий сервер. Також виконує перевірку конфігурації, описану вище;

stop − коректно завершає роботу поштової системи. При цьому всі запущені системою процеси перериваються при першому ж зручному випадку. Для того, щоб обновити параметри системи після внесення змін у конфігурацію, не використовуйте послідовно команди *start* і *stop*. Замість цього скористайтеся командою *reload*;

abort – робить аварійне завершення роботи серверу. Всі запущені системою процеси перериваються негайно;

flush − форсує доставлення листів: здійснюється спроба доставити кожне з повідомлень, що знаходяться у черзі відкладених повідомлень. Звичайно цей процес здійснюється періодично, через заданий проміжок часу, що подвоюється з кожною невдалою спробою;

reload − заново зчитує інформацію з конфігураційних файлів. При цьому всі запущені системою процеси перериваються при першому ж зручному випадку.

Сервер Postfix використовує значну кількість файлів для різних настроювань, таких як база аліасів користувачів, перенапрямок листів на інші поштові адреси та ін. Базовими для забезпечення працездатності є файли master.cf - конфігураційний файл серверного процесу master, /var/spool/postfix/pid/master.pid - файл блокувань процесу master, і main.cf - конфігураційний файл серверу. На початковому етапі роботи в master.cf зміни як правило не вносяться, тому зосередимо свою увагу на конфігураційному файлі.

В /etc/postfix/main.cf вам потрібно установити значення низки конфігураційних параметрів. Ці параметри нагадують змінні оболонки shell, але є дві важливих відмін-

ності. Перше: Postfix не знає про лапки (quotes) у розумінні оболонки UNIX. Ви привласнюєте значення конфігураційному параметру в такий спосіб:

/etc/postfix/main.cf:
parameter = value

і використовуєте значення параметра, додаючи символ "$" перед його ім'ям:

/etc/postfix/main.cf:
other_parameter = $parameter

Ви можете використати $parameter перед тим, як привласните йому значення (це друга основна відмінність від змінних оболонки UNIX). Мова Postfix використовує ледаче оброблення (lazy evaluation) і не дивиться на значення параметрів, поки вони не знадобляться під час роботи.

Розглянемо основні параметри, використовувані при настроюванні серверу Postfix.

Параметр *myorigin* указує ім'я домена, що використовується в пошті, що відправляється із цієї машини. За замовчуванням використовується ім'я локальної машини. Якщо тільки Postfix працює не на дуже маленькому сайті, ви ймовірно захочете призначити параметру *myorigin* значення *$mydomain*, яке за замовчуванням містить доменну частину повного імені машини. Приклади (використайте лише один варіант із наступних):

myorigin = $myhostname (за замовчуванням: надсилати листа від "user@$myhostname")

myorigin = $mydomain (можливо, віддати перевагу: "user@$mydomain")

Параметр *mydestination* указує, для яких доменів пошта буде доставлятися локально замість пересилання на інший хост. За замовчуванням Postfix приймає пошту тільки для локальної машини. Може бути кілька варіантів настроювання цього параметра:

− Значення за замовчуванням:
mydestination = $myhostname localhost.$mydomain localhost
− Поштовий сервер для всього домена:
mydestination = $myhostname localhost.$mydomain localhost $mydomain
− Хост із декількома адресними записами DNS:
mydestination = $myhostname localhost.$mydomain localhost www.$mydomain ftp.$mydomain

Щоб уникнути зациклення доставлення пошти (mail delivery loops), Вам належить указати всі імена машини, включаючи $myhostname і localhost.$mydomain.

За замовчуванням Postfix пересилає пошту від клієнтів, що перебувають в авторизованій частині мережі, на будь-яку адресу. Авторизовані мережі визначає конфігураційний параметр *mynetworks*. Поводження за замовчуванням - авторизувати всіх клієнтів IP-підмережі, до яких приєднана машина. Якщо ваша машина має зовнішню адресу, такі настроювання можуть бути небезпечними. Можна використати такі варіанти:

mynetworks_style = subnet (за замовчуванням: авторизувати підмережі);
mynetworks_style = host (безпечно: авторизувати тільки локальну машину);
mynetworks = 127.0.0. 0/8 (безпечно: авторизувати тільки локальну машину);
mynetworks = 127.0.0. 0/8 168.100. 189.2/32 (авторизувати тільки локальну машину).

За стандартними настроюваннями Postfix пересилає пошту від сторонніх (тобто клієнтів, що перебувають за межами довірених мереж) тільки авторизованим доменам. Домены, на які дозволене пересилання кореспонденції від сторонніх клієнтів, визначає параметр *relay_domains*. За замовчуванням Postfix уважає авторизованими всі домени (і піддомени), зазначені в параметрі mydestination. Приклади використання:

relay_domains = $mydestination (за замовчуванням);
relay_domains= (безпечний варіант: ніколи не пересилати пошту від сторонніх);
relay_domains = $mydomain (пересилати пошту , адресовану своєму домену й поддоменам).

Ви повинні вказати псевдонім (alias) для postmaster-а в таблиці аліасів, який перенаправляє пошту реальній людині. Адреса postmaster повинна існувати для того, щоб користувачі могли повідомити про проблеми з доставленням пошти. Під час редагування таблиці аліасів не забудьте також перенаправити пошту суперкористувача (root) реальній людині. Вміст /etc/aliases:

```
postmaster: you
      root: you
```

Після зміни аліасів потрібно виконати команду newaliases.
Деякі поштові сервери підключені до Internet через NAT або проксі-сервер. Це означає, що зовнішні системи відкривають з'єднання із проксі або NAT, а не прямо з'єднуються з поштовим сервером. Проксі або NAT, у свою чергу, з'єднується із цільовим поштовим сервером, але Postfix про це не знає. У цьому випадку вам належить зазначити всі зовнішні адреси проксі або NAT-а, з яких Postfix одержує пошту, в параметрі *proxy_interfaces*. Ви можете зазначити символьні імена хостів замість мережних адрес. Також ви повинні зазначити Ваші зовнішні проксі/NAT адреси, коли ваша система функціонує як резервний MX хост для інших доменів, інакше при падінні основного поштового серверу відбудеться зациклення доставлення пошти.
Розглянемо приклад файла main.cf.

```
queue_directory = /var/spool/postfix
command_directory = /usr/sbin
daemon_directory = /usr/libexec/postfix
mail_owner = postfix
# ім'я вашого поштового серверу
myhostname = test.example.org
inet_interfaces = all
# домени, для яких ви одержуєте пошту
```

```
mydestination = example.org, sample.com, example.local
unknown_local_recipient_reject_code = 450
# ip-адреси вашої локальної мережі
mynetworks = 192.168.0.0/24, 127.0.0.0/8
# наступні два рядки дуже важливі!
# за замовчуванням postfix - open relay.
# Якщо ви їх не зазначити, то надто швидко
# вас пропишуть на spamcop.net в blacklist
# і ви не зможете відправляти пошту!
relay_domains = $mydestination
relay_domains_reject_code = 554
# домени, які будуть підставлятися в листи
# при відправленні по Інтернету
masquerade_domains = example.org
# у цьому файлі настроюється переадресації пошти
# формат рядків у цьому файлі наступний:
# адреса_одержувача нова_адреса_одержувача
# наприклад:
# user1@example.org  mailuser@rambler.ru
# Пошта, що  приходить на адресу  user1@example.org
# буде переспрямована на адресу mailuser@rambler.ru
virtual_alias_maps = hash:/etc/postfix/virtual
alias_maps = $alias_database
mail_spool_directory = /var/spool/mail
# вітання, видаване сервером при підключенні
# клієнтів. Може бути будь-яке.
smtpd_banner = $myhostname ESMTP $mail_name ($mail_version). Example SMTP
Server.
debug_peer_level = 2
debugger_command =
   PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin
   xxgdb $daemon_directory/$process_name $process_id & sleep 5
sendmail_path = /usr/sbin/sendmail.postfix
newaliases_path = /usr/bin/newaliases.postfix
mailq_path = /usr/bin/mailq.postfix
setgid_group = postdrop
manpage_directory = /usr/share/man
sample_directory = /usr/share/doc/postfix-2.0.8/samples
readme_directory = /usr/share/doc/postfix-2.0.8/README_FILES
# файл із псевдонімами
alias_database = hash:/etc/postfix/aliases
```

3 Контрольні питання

3.1 Поясніть внутрішню архітектуру серверу Postfix і її основні переваги.
3.2 Який системний процес відповідає за запуск, підтримку працездатності й завершення компонентів Postfix?
3.3 Назвіть основні способи попадання листів у поштову систему.

3.4 Які функції виконує процес cleanup?
3.5 Назвіть основні способи відправлення поштових повідомлень.
3.6 В якому випадку відправляють листи, що можуть виявитися в одній із черг deffered, corrupt або hold ?
3.7 Які заходи приймаються для підвищення захищеності серверу Postfix?
3.8 Назвіть основні файли, які використовує у своїй роботі сервер Postfix.

4 Домашнє завдання

4.1 Вивчіть ключові положення.
4.2 Підготуйте відповіді на ключові питання.

5 Лабораторне завдання

5.1 Перевірте наявність у системі облікового запису postfix, а також груп postfix і postdrop, необхідних для роботи серверу Postfix.
5.2 Виконайте настроювання поштового серверу, забезпечивши такі його параметри:
– ім'я поштового домена – *прізвище_студента.org*
– поштовий сервер слухає запити на інтерфейсах lo0 і rl0
– одержувати пошту для доменов *студент1_бригади.org ... студент_бригади.org*
– при відправленні листів підставляти ім'я домена *прізвище2_студенти.org*
5.3 Перевірте коректність настроювань файла main.cf.
5.4 Стартуйте сервер Postfix.
5.5 За допомогою протоколу smtp установити консольне з'єднання з вашим поштовим сервером, і відправити тестовий лист від свого імені на поштову скриньку, зазначену викладачем.
5.6 За допомогою протоколу POP3 одержати доступ до поштової скриньки адресата, і переконатися в коректній доставці поштового повідомлення.

6 Вимоги до змісту протоколу

6.1 Назва лабораторної роботи.
6.2 Мета роботи.
6.3 Результати виконання домашнього завдання.
6.4 Короткий опис виконаної роботи.
6.5 Висновки про виконану роботу.
6.6 Дата, підпис студента, віза викладача.