Лабораторна робота
Система нотарізації Syslog

Syslog це мережний стандарт відправки і реєстрації повідомлень про системні події.
Терміном «Syslog» називається як мережний протокол, так і інші компоненти системи нотарізації (реєстрації системних подій) - системний виклик, демон, бібліотека і конфігураційний файл. Останнім часом популярним програмним продуктом є Rsyslog, що включає всю функціональність Syslog, та має розширену функціональність. Існує ще одне популярне альтернативне рішення - Syslog-ng. Однак Syslog-ng не сумісний з оригінальним Syslog або Rsyslog.

Syslog - це функція (системний виклик), яка використовується багатьма програмами для того, щоб передати повідомлення системному реєстратору повідомлень (syslogd). Демон syslogd читає і виводить повідомлення в системну консоль, в реєстраційні файли (log файли), на інші машини і користувачам відповідно параметрам свого конфігураційного файла (/etc/syslog.conf).

При великій кількості серверів стає незручним обходити всі сервера щоб переглянути на них системний журнал і з'являється бажання системні журнали з усіх серверів зібрати в одному місці.

В цьому випадку можна скористатися можливістю більшості реалізацій syslog-демонів обмінюватися інформацією з іншими syslog-серверами використовуючи стандартний протокол

Система журналювання (syslog) є однією з найбільш чудових речей в UNIX. На відміну від деяких операційних систем, які змушують вас використовувати лише той обмежений діапазон журналів, які вони зволять вам надати, UNIX дозволяє вам реєструвати майже все що завгодно з практично будь-яким рівнем деталізації. Так як стандартні системні засоби журналирования передбачені для більшості засобів системи UNIX, адміністратор може вибрати конфігурацію журналирования, що задовольняє його вимогам. Зазвичай мережа має один вузол нотаризації, який підтримує журнал не тільки для FreeBSD-вузлів, але і для маршрутизаторів Cisco, комутаторів і будь-яких інших систем, що підтримують syslog.

Система нотаризації влаштована досить просто. Програми надсилають повідомлення, призначені для запису, до системного демона syslogd. Syslogd порівнює кожне отримане повідомлення з правилами, які знаходяться в файлі /etc/syslog.conf. Коли є відповідність, syslogd обробляє повідомлення за налаштуваннями в syslog.conf.

У програмі syslog може бути викликаний наступним чином:

```
#include <syslog.h>
...
openlog (“мітка”, 0, LOG_USER);
syslog(LOG_NOTICE, “Log message.”);
closelog();
…
```

де «мітка» - текст (TAG), який передуватиме повідомленням, LOG_USER - facility (підсистема), LOG_NOTICE - severity (рівень) повідомлення.

Файл /etc/syslog.conf складається з двох стовпців. У першому вказується правило відбору записів для журналу. У другому міститься опис дій для обробки відповідного повідомлення.

Більшість труднощів викликає повне розуміння того, як точно вказати правило відбору записів для нотаризації.

Джерело записів описується категорєю (facility) і рівнем значущості (severity або level). Категорія це або джерело записів, або програма, яка шле повідомлення демона syslogd. Існують наступні категорії:

- **auth** Все що пов'язано з авторизацією користувачів, на кшталт login і su.
- **authpriv** Теж саме що і auth, проте пише журнал в файл, який можуть читати лише деякі користувачі (очевидно, автор мав на увазі той факт, що повідомлення, які збираються в цій категорії, можуть містити відкриті паролі користувачів, які не повинні потрапляти на очі стороннім людям, і отже файли журналів повинні мати відповідні права доступу).
- **console** Повідомлення, що виводяться на системну консоль, можуть бути записані в журнал за допомогою цієї категорії.
- **cron** Повідомлення від системного планувальника.
- **daemon** Пастка для повідомлень від усіх інших системних демонів, які не мають явно описаних категорій.
- **ftp** За допомогою цієї категорії ви зможете налаштувати ваш FTP сервер, що б він записував свої дії. Дивіться /etc/inetd.conf.
- **kern** Повідомлення від ядра.
- **lpr** Повідомлення від системи друку.
- **mail** Повідомлення від поштової системи.
- **mark** Ця категорія використовується для того, щоб записувати в журнал повідомлення кожні 20 хвилин. Вона може бути корисна в комбінації з деякими іншими журналами (наприклад ви зможете дізнатися з 20-ти хвилинної точністю, коли ж завис ваш сервер).
- **news** Повідомлення від сервера новин.
- **ntp** Повідомлення від сервера точного часу.
- **security** Повідомлення від різних служб безпеки, таких як ipfw або ipf.
- **syslog** Система журналювання може журналіровать повідомлення від самої себе. Тільки не пишіть в журнал повідомлення про те, що ви пишете в журнал повідомлення від системи журналювання, інакше у вас закрутиться голова :-)
- **user** Призначена для збору різноманітних повідомлень. Якщо ви не вкажете категорію повідомлення для програм користувача, то вони будуть використовувати цю категорію (більш ніж спірне твердження, заув. перекладача).
- **uucp** Збирає повідомлення від UNIX-to-UNIX Copy Protocol. Це частина історії UNIX і найімовірніше вона вам ніколи не знадобиться (хоча до сих пір велика частина поштових повідомлень доставляється через UUCP).
- **local0 - local7** Зарезервовані категорії для використання адміністратором системи. Багато програм дають можливість вказати категорію повідомлень syslog. Якщо програма це дозволяє - вибирайте одну з них.

Більшість систем записують далеко не все, про що повідомляють їх програми. Частішн незначні повідомлення відкидаються, а записуються тільки важливі події. Однак те, що здається одній людині незначним, іншому може здатися істотним. Тут ми зустрічаємося з рівнями значущості повідомлень.

Syslog надає вісім **рівнів значущості (важливості)** повідомлень. З їх допомогою ви можете вказати syslog, що записувати в журнал, а що відкинути. Ось ці рівні, в порядку зменшення важливості:

- **emerg** Система в паніці. Повідомлення негайно виводяться на всі активні термінали. Система скоріш за все накривається мідним тазом :-), або залишається надзвичайно, надзвичайно нестабільною. Продовження роботи неможливо.
- **alert** Це погано, але не настільки погано, як рівень emerg. Система може продовжити роботу, але цю помилку слід усунути негайно.
- **crit** Це критичні помилки, такі як проблеми з апаратним забезпеченням або серйозні порушення роботи програмного забезпечення. Якщо ваш жорсткий диск містить погані блоки, вони проявляться у вигляді критичних помилок. Якщо ви дуже сміливий, спробуйте продовжити роботу.
- **err** Різноманітні помилки. Це погано, такі помилки повинні бути усунені, але вони не зруйнують вашу систему.
- **warning** Різноманітні попередження.
- **notice** Загальна інформація, що повинна бути записана, якщо вона вам потрібна, але, ймовірно, вона не вимагає вашої реакції.
- **info** Різна системна інформація.
- **debug** Цей рівень зазвичай використовується програмістами і іноді системними адміністраторами, які намагаються зрозуміти, чому ж ця програма так чинить. Налагоджувальні повідомлення можуть містити всю інформацію, яку визнав за необхідне вивести її розробник для налагодження коду; між іншим, вона може містити дані, що порушують приватність користувачів.
- **none** Це спеціальний рівень означає "нічого не записувати в даній категорії". Він зазвичай застосовується для виключення інформації з групових записів.

Опис правила відбору джерела інформації включає в себе категорію і рівень деталізації, розділені крапкою. Коли ви вказуєте рівень, за замовчуванням в журнал записуються повідомлення, рівень яких **вище або дорівнює вказаному**. Як приклад, розглянемо цей запис з файлу /etc/syslog.conf:

```
mail.info /var/log/ maillog
```

У журнал /var/log/maillog будуть записані повідомлення від поштової системи з рівнем info або більше.

Якщо виникне потреба, то ви можете скористатися символом "*" в описі джерела повідомлення. Наприклад, для запису всіх повідомлень від поштової системи ви можете скористатися наступним синтаксисом:

```
mail. * /var/log/maillog
```

Для запису в журнал абсолютно всіх подій, що відбуваються в системі, розкоментуйте рядок all.log (в файлі /etc/syslog.conf):

```
*. * /var/log/all.log
```

Це спрацює, однак такий файл буде містити занадто багато дуже докладних відомостей, щоб його можна було реально використовувати. Для знаходження корисних відомостей вам доведеться кожного разу споруджувати нетривіальні послідовності команд grep.

Завдяки тому, що категорія налагоджувальних повідомлень теж підпадає під це правило, всі приватні відомості користувачів потраплять в цей журнал. Ймовірно, ви не захочете записувати подібну інформацію. Ви можете виключити автентифікаційну інформацію,

використовуючи категорію authpriv з рівнем none. Крапка з комою дасть вам можливість об'єднати правила в одному рядку:

```
*. *; authpriv.none /var/log/all.log
```

У /etc/syslog.conf ви можете використовувати оператори порівняння. Можливі наступні оператори: "<" (менше ніж), "=" (дорівнює), ">" (більше ніж). Застосувавши ці оператори, ви зможете, наприклад, розділити журнал записів поштового трафіку і журнал налагоджувальної інформації, наданої поштовою системою:

```
mail.info /var/log/maillog
mail. = debug /var/log/maillog.debug
```

Таким чином, вам не потрібно буде відсортовувати повідомлення для того, щоб дізнатися, що думає ваш поштовий сервер про те, що він робить.

Подібним чином у вас може виявитися програма, яка захоче використовувати для ведення журналу, наприклад, категорію local3. Ви можете записати інформацію від неї в такий спосіб:

```
local3.* /var/log/whatever
```

Як джерело записів, ви можете вказати ім'я програми. Якщо програма дозволяє використовувати категорії, застосовуйте їх. Однак, якщо вам не вистачає категорій (local0-7 цілком можуть закінчитися), або програма просто не підтримує syslogd, то ви можете використовувати її ім'я.

Такий запис складається, як мінімум, з двох рядків. У першому рядку знаходиться назва програми, на початку якого знаходиться знак оклику. Друга містить параметри журналювання. Наприклад, як виглядає запис для збору інформації про дії ppp:

```
! ppp
*. * /var/log/ppp.log
```

Вона починається з вказання імені програми і потім вказує syslog, що необхідно записувати абсолютно всю інформацію від неї в файл. Навряд чи ви можете бути впевнені, що випадкова програма сторонніх виробників має відповідні категорії нотаризації, так що кращим виходом буде запис в журнал всіх повідомлень цієї програми.

Нарешті ми підійшли до опису другої частини файлу /etc/syslog.conf.
У більшості випадків він містить повне ім'я файлу журналу, але існують і інші способи обробки повідомлень.

Ви можете надсилати інформацію нотаризації на іншу машину, поставивши перед її ім'ям символ "@". Наступний приклад демонструє як можна перенаправляти всі отримані вашим syslog повідомлення на виділений syslog-сервер в мережі:

```
*.* @loghost.blackhelicopters.org
```

/etc/syslog.conf на loghost використовується для остаточної обробки надісланих записів.

Як спосіб обробки записів ви можете вказати імена користувачів, розділені комами. Якщо ці користувачі будуть знаходиться в системі в момент приходу повідомлення, що задовольняє

вказаній умові, то воно буде перенаправлено на їх термінали. Якщо ви хочете показувати деякі повідомлення на всіх терміналах користувачів, то слід скористатися символом "*".

Нарешті, якщо ви хочете скористатися якою-небудь іншою програмою для обробки поідомлень, ви можете використовувати символ "|" для перенаправлення потоку введення-виведення на цю програму:

```
mail. * | /usr/local/bin/mailstat.pl
```

Тепер, коли підсистему нотаризації налаштовано відповідно до ваших запитів, вас повинно турбувати, що врешті-решт файли журналів заповнять весь ваш диск!

За замовчуванням, /etc/syslog.conf буває налаштований таким чином, що дуже багато повідомлень виводиться прямо на консоль.

Приклад файла налаштування syslog.conf:

*.info;mail.none;authpriv.none;cron.none	/var/log/messages
authpriv.*	/var/log/secure
mail.*	/var/log/maillog
cron.*	/var/log/cron
*.emerg	*
uucp,news.crit	/var/log/spooler
local7.*	/var/log/boot.log

Послідовність налаштування Syslog:
Важливо: вам треба знати про syslog.conf, що він вимагає символи табуляції, а не пробіли! Тому, якщо після редагування цього файлу ви раптом починаєте отримувати помилки на кшталт цих, то швидше за все замість символів табуляції ви вставили пробіли. Майте на увазі, що редактор вставляє пробіли, навіть якщо ви натискаєте клавішу табуляції, в той час як vi в цьому гріху не помічений.

На логсервері:
1) На початку /etc/syslog.conf прописати:

Якщо машина-клієнт логсервера с ім'ям hostname, з якої будуть надходити логи, знаходиться в тому ж домені, що і logserver і в /etc/resolv.conf прописано search [домен] і domain [домен], то в налаштуваннях має бути:

```
+ hostname
*. * /var/log/hostname.log
```

інакше як ім'я машини потрібно буде вказати повністю FQDN, тому що саме це ім'я буде резолвитись і фігурувати в логах syslogd, тоді потрібно прописати в /etc/syslog.conf:

```
+ hostname.domain
*. * /var/log/hostname.log
```

При розбіжності імені машини, що отримfне від DNS (як його уявляє собі логсервер) з тим ім'ям хоста, яке прописано в syslog, логи від цієї машини писатися в файл не будуть.

2) Створити лог-файл для клієнта:
touch /var/log/hostname.log

chmod 640 /var/log/hostname.log

3) Щоб увімкнути віддалену реєстрацію, в файлі /etc/sysconfig/syslog сервера змінити:
SYSLOGD_OPTIONS = "- r -m 0"

4) Запустити знову syslogd:
service syslog restart

На машині, яка буде відправляти логи на віддалений логсервер з ім'ям хоста logserver:

1) Прописати редирект всіх повідомлень на логсервер:
*. * @logserver
Потрібно переконатися, що клієнт зможе отримати ім'я logserver (logserver повинен бути прописаний в /etc/hosts або знаходитись через DNS-сервер, вказаний в /etc/resolv.conf).

2) Запустити знову syslogd:
service syslog restart

З цього моменту віддалена нотаризація має працювати.

Завдання лабораторної роботи:

1. Налаштувати реєстрацію повідомлень заданого рівня/засобу в окремий журнал (за індивідуальним завданням).

2. Налаштувати протоколювання подій деякої програми (організуйте реєстрацію повідомлень, що видаються при виконанні команд login, passwd, su, sudo, iptables, ...в окремому журналі).

3. Скомпілювати програму, яка генерує повідомлення syslog за допомогою однойменного системного виклику. Створіть журнал для реєстрації повідомлень даної програми.

4. Налаштувати віддалений сервер на прийом і реєстрацію повідомлень Syslog з вашого хоста.

5. Налаштувати реєстрацію повідомлень на віддалений сервер.

6. Згенерувати за допомогою утиліти logger події, що викликають запис налаштованих раніше подій і перевірити, чи з'явилися в журналах реєстрації записи.

7. Використовуючи утиліти cut, sed і awk, організуйте нормалізацію, фльтрацію і експорт повідомлень з деякого log-файлу в формат, придатний для імпорту в СКБД (форматований текст з роздільниками).

8. За допомогою logrotate налаштуйте ротацію log-файлів (щодня заміщати log-файл заданої програми, не стискати, та зберігати архівні файли 8 днів) і перевірте зроблені налаштування експериментально.